Case Study on the Development of a Recommender for Apple Disease Diagnosis with a Knowledge-based Bayesian Network

Gabriele Sottocornola[1], Sanja Baric[1], Fabio Stella[2] and Markus Zanker[1]

[1]Free University of Bozen-Bolzano, Piazza Università, 1, 39100 Bolzano, Italy

[2]University of Milano-Bicocca, Piazza dell'Ateneo Nuovo, 1, 20126 Milano, Italy

Abstract

This paper presents a case-study of a knowledge-based recommender system capable to diagnose post-harvest diseases of apples. It describes the process of knowledge elicitation and construction of a Bayesian Network reasoning system as well as its evaluation with three different types of studies involving diseased apples. The ground truth of diseased instances has been established by genome sequencing in a lab. The paper demonstrates the performance differences of knowledge-based reasoning mechanisms due to different users interacting with the system under different conditions and proposes methods for boosting the performance by likelihood evidence learned from the estimated consensus of users' and expert's interactions.

Keywords

Case Study in Agriculture, Knowledge-based Recommendation, Bayesian Network, Likelihood Evidence

1. Introduction

Apple trees are the most common temperate fruit tree species, since their fruits can be stored for prolonged periods of time under controlled atmosphere conditions. However, physiological disorders and pathogenic microorganisms can deteriorate the quality and quantity of the production during storage, and lead to considerable economic losses [1]. For instance, in Northern Europe, storage losses due to pathogenic microorganisms were estimated to reach up to 10% in integrated production and up to 30% in organic production [2]. Therefore, an effective knowledge-based recommender system, able to timely suggest a correct diagnosis of diseases manifested on stored apples, is of crucial importance. For instance, it depends on the exact pathogen species to decide on the right strategy for immediate damage containment and/or to recommend a plant protection scheme for the following year. In order to reliably determine the nature of the disease, several macroscopic symptoms, such as appearance, color, texture and consistency of the rot need to be considered by the system. Hence, we should provide a practical interface to elicit user feedback on manifested symptoms on a diseased apple in order to guide the reasoning to recommend a diagnosis. Thus, we propose *BN-DSSApple* a decision support system based on the framework of Bayesian Networks (BN), a graphical probabilistic method to reason about uncertainty relationships among symptoms, signs, and diseases. The user observation (i.e., the evidence) is elicited incrementally through an adaptive question-answering interface, illustrated by visual explanation of the requested information in order to facilitate user understanding. Furthermore, we illustrate the process adopted to build the diagnostic knowledge base with the help of a domain expert in the field of post-harvest apple diseases. We analyse and address the problem of transferability of such an expert model to a larger cohort of users with different expertise levels. We thoroughly tested *BN-DSSApple* under different experimental conditions, simulated in 3 user studies, to prove the effectiveness of the system and its transferability across different environments.

The methodological contribution of this case study is organized according to this pipeline: a) in Section 3.1, we describe the application domain and the implemented *BN-DSSApple* system; b) in Section 3.2, we illustrate the process of knowledge elicitation from a domain expert for crafting the knowledge base of the BN; c) in Section 3.3, we formalize the recommendation mechanism responsible for the suggestion of a suitable diagnosis given the user feedback; d) in Section 3.4, we define the trasportability problem of a knowledge-based model and we propose a possible solution, exploiting the so-called likelihood evidence.

3rd Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) & 5th Edition of Recommendation in Complex Environments (ComplexRec) Joint Workshop @ RecSys 2021, September 27–1 October 2021, Amsterdam, Netherlands

gsottocornola@unibz.it (G. Sottocornola); sanja.baric@unibz.it (S. Baric); fabio.stella@unimib.it (F. Stella); markus.zanker@unibz.it (M. Zanker)

0000-0001-9983-2330 (G. Sottocornola)

2. Background

A *Bayesian Network (BN)* [3, 4] is defined by its two main components: the qualitative part represented by its graphical structure and the quantitative part consisting of the conditional probabilities. More formally, a BN is graphically represented as a *directed acyclic graph (DAG)* $\mathcal{G} = (N, E)$, where $N = \{n_1, n_2, \ldots, n_l\}$ denotes the set of l nodes and $E \subseteq N \times N$ the set of directed edges between pairs of nodes. Each node $n_i \in N$ in the DAG $\mathcal{G}$ is mapped one-to-one with a random variable $X_i \in \mathcal{X}$, where $\mathcal{X}$ denotes the set of random variables involved in the model. A random variable $X_i \in \mathcal{X}$ is represented by a set of exclusive values (or states) in which the variable might be observed $Val(X_i) = \{x_i^1, x_i^2, \ldots, x_i^m\}$, where $x_i^j \in Val(X_i)$ denotes the j-th value of variable X_i. We use the notation $X_i = x_i^j$ for an observed event, to express that variable $X_i \in \mathcal{X}$ is observed (or instantiated) in the state $x_i^j \in Val(X_i)$. A *conditional probability table (CPT)* is associated to each random variable $X_i \in \mathcal{X}$. The CPT specifies the conditional probability distribution $P(X_i|pa(X_i)) \in \mathcal{P}$ over the states of X_i. Where, $\mathcal{P}$ represents the set of conditional probabilities in the model, and $pa(X_i) \subset \mathcal{X}$ denotes the set of the so-called *parents* of the variable X_i associated to the node n_i in the DAG $\mathcal{G}$. Specifically, the parent set of X_i is composed by every variable $X_j \in \mathcal{X}$ associated to the node n_j in the DAG $\mathcal{G}$, connected with a directed edge to n_i (the so-called *child* node). More formally, $pa(X_i) = \{X_j \in \mathcal{X} : (n_j, n_i) \in E\}$. We can further define an *ancestor* variable $an(X_i)$ of the variable X_i, and a *descendant* variable $de(X_i)$ of variable X_i, if exists a directed path (i.e., a set of directed edges) connecting node n_a (associated with variable $an(X_i)$) to n_i (associated with variable X_i), and n_i to n_d (associated with variable $de(X_i)$); namely $\{(n_a, n_j), (n_j, n_i), (n_i, n_h), \ldots, (n_g, n_d)\} \subset E$. It is important to mention that the DAG $\mathcal{G}$ of the BN specifies a set of probabilistic relationships among variables in the model. Namely, if an edge $(n_j, n_i) \in E$ exists in the graph, this generally implies that a causal relation holds between the variables X_j and X_i, associated to nodes n_j and n_i. Specifically, we typically assume that the parent X_j represents the cause and child X_i represents the effect in the domain. Thus, a fundamental assumption of conditional (in)dependence between variables could be derived. This assumption is the *Local Markov Assumption* (or *Local Independence Assumption*), and it states that: given its parents $pa(X_i) \subset \mathcal{X}$, defined in the DAG $\mathcal{G}$, a variable X_i is conditionally independent of all its non-descendant variables. More formally, for each variable X_i: $(X_i \perp X_j | pa(X_i))$, where $X_j \notin de(X_i)$, set of descendants of X_i. This property allows to specify the joint distribution over the space of the variables $\mathcal{X}$ in the BN model through the probability factorization $P(\mathcal{X}) = \prod_{i=1}^{l} P(X_i|pa(X_i))$, usually referred to as the *chain rule for Bayesian networks.*

3. Methodology

3.1. System Description

The presented knowledge-based decision support system, named *BN-DSSApple*, is conceptualized as an interactive easy-to-use web application that allows users with different levels of domain expertise in the area of apple production (e.g., farmers, quality controllers, and storage workers), to perform in-field diagnosis of post-harvest diseases of apple fruit, relying solely on the observed macroscopic symptoms on the stored fruit. The system is designed as a recommender engine which collects the feedback of the user (i.e., the evidence) on a specific apple fruit (i.e., the target apple), in order to suggest a suitable diagnosis (i.e., a set of recommended diseases). The reasoning mechanism is performed by a Bayesian Network (BN) based on an ad-hoc knowledge base, constructed with the help of a domain expert (as described in 3.2).

Specifically, the system collects user's feedback about the target apple by asking a set of dynamic multiple-choice questions related to the macroscopic features of the observed symptoms (e.g., the shape of the rot, the origin of the infection, etc.). Each question is illustrated with exemplary pictures, facilitating also non-expert users in their understanding. Each question is mapped to a specific variable in the BN model. This part of the system is dynamic, since the system incrementally adapts the questions path based on the previous answers given by the user. For instance, when the system gets the information that spores are visible on the infected apple, it will inquiry the user about further features of those spores (i.e., their mass distribution, colour, and origin). Furthermore, the system provides full flexibility to the user, i.e., it allows to navigate the questions path back and forth in order to revise previous answers, to provide multiple answers, or to skip questions in case of lacking confidence.

3.2. Knowledge Elicitation for Bayesian Network

In order to build a diagnostic reasoning system based on Bayesian network (i.e., both the network structure and the CPTs) two options are available: learn from the data or elicit the knowledge from the domain literature or the experts, or any combination of the above. At the best of our knowledge, no datasets are publicly available to learn significant relationships among apple diseases and macroscopic symptoms. Thus, we started by analysing a large OWL ontology which captures the entire life cycle of apple cultivation, production, handling, and storage, presented in [5]. Hence, we extracted a smaller quantitative part of the presented ontology suitable for our goal, which allows a simple reasoning mechanism connecting symptoms to diseases, thanks to a set of SWRL rules [6].

The graphical structure of this ontology is represented in Figure 1. At the best of our knowledge, the difficult task to (semi)-automatically construct a BN from a domain ontology is still under-explored in the literature. Few practical, heuristic solutions can be found [7, 8], which can hardly be applied to our case. The main limitation of such an effort lays in the fact that the two frameworks differ in the purpose they are used for. An ontology is more suitable to describe concepts and qualitative relationships (of different nature), while the BN requires quantitative definitions (i.e., probabilistic) of correlation relationships related to the reasoning mechanism of phenomena [9].

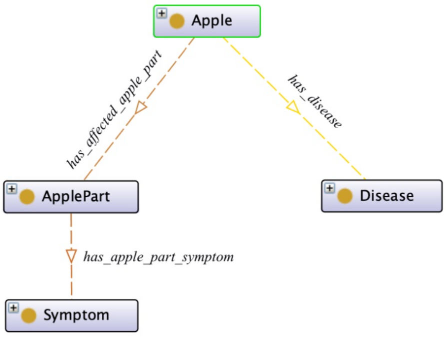


Figure 1: The initial ontology for BN-DSSApple.

We overcame this problem by directly interviewing a domain expert for the construction of the knowledge base. Specifically, we divided this task into two distinct phases: during the first phase, we identified the random variables (i.e., the macroscopic symptoms) which are relevant in the diagnostic process; during the second phase, we determined the probability values (i.e., the CPTs) quantitatively linking the diseases to the symptoms. We firstly asked the domain expert to review the available ontology, enrich and adapt it in order to obtain an effective tool for the diagnosis of post-harvest diseases of apple based on visible macroscopic symptoms on it. After few rounds of interaction, we agreed with a set of 27 discrete random variables (12 boolean and 15 categorical) related to macroscopical symptoms and signs that could be observed on the infected apple skin and pulp, together with two hidden (target) variables, namely *Disease* and *Stage*. We assumed that a target apple could be infected by one and only one disease and thus, the random variable *Disease* encodes the whole set of bacterial diseases of our study, namely the 7 diseases *Val(Disease) = {alternaria_rot, alternaria_spot, bitter_rot, botrytis, mucor_rot, neofabraea, penicillium}*. The *Stage* random variable was introduced to facilitate the experts' probability elicitation task. The variable represents three discrete and symbolic stages of advancement of the post-harvest infection, namely *Val(Stage) = {early, medium, late}*. This workaround allows the expert to visualize a specific condition of the disease and thus specify a more reliable likelihood of the symptoms.

The final *BN-DSSApple* graph is reported in Figure 2. The central nodes in the network, bolded and empty, represent the two hidden diagnosis variables, namely *Disease* and *Stage*. On the top part of the network, coloured in grey, are the nodes related to the *lesion properties*. On the right-most part, colored in yellow, are the *rot properties*, while on the left-most part, colored in green, are the *lesion origin* nodes. Finally, in the central-bottom part, colored in orange, are represented the nodes related to the *lesion type* and *other symptoms*, under those, colored in cyan, the nodes representing the properties of the other symptoms.

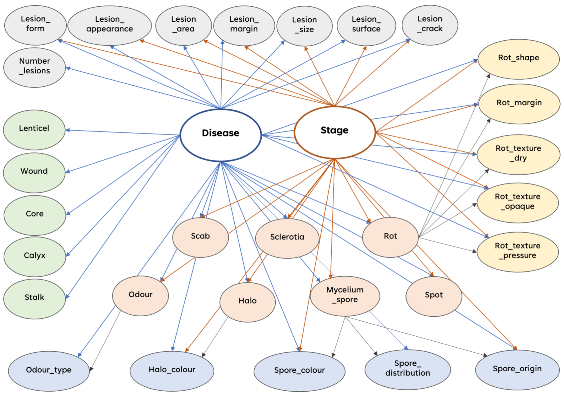


Figure 2: The graph of the Bayesian Network for *DSSApple*.

In the second phase, we interviewed the domain expert in order to define the quantitative probabilistic dependencies among variables. For simplicity, we decided to start from a situation where all the symptom variables are conditionally independent among each other, given the states of *Disease* and *Stage*. Furthermore, they all depends from the two hidden variables responsible for the assessment of the diagnosis (i.e., *Disease* and *Stage*). We indicate the *Disease* variable as $D \in \mathscr{D}$, where $\mathscr{D}$ defines the set of hidden variables for the model. $Val(D) = \{d^1, d^2, \dots d^n\}$ represents the set of states of the variable D, where d^i is the i-th state of the *Disease* variable (i.e., the i-th disease in our pool). The *Stage* variable is referred as $T \in \mathscr{D}$ and $Val(T) = \{t^1, t^2, \dots t^m\}$ represents the set of states of variable T, where t^i is the i-th state of the *Stage* variable. All other (observed) variables in the model are referred as symptom variables and they belong to the set $\mathscr{S}$. A generic symptom variable $S_i \in \mathscr{S}$ is represented by a set of states $Val(S_i) = \{s_i^1, s_i^2, \dots s_i^q\}$, where s_i^j is the j-th state of the symptom variable S_i. Moreover, we adapted the procedures described in [10] for eliciting expert probabilities of our network. Specifically, we adopted a mixed symbolic questionnaire to facilitate the expert expressing the conditional probability of each event. In more details, two techniques were applied depending on the support of the variable. For boolean variables (for each symptom variable $S_i \in \mathcal{S}$ such that $Val(S_i) = \{true, false\}$), the expert was invited to answer the question: *"How frequently do you observe symptom $S_i = true$, given that you have an apple infected by disease $D = d_l$ at stage $T = t_j$?"*. We allowed her to select one option on a pre-defined 6-point scale, including *Always (A)*, *Very often (V)*, *Often (O)*, *Sometimes (S)*, *Rarely (R)*, and *Never (N)*. The expert had to fill a form, providing the answer for each combination of $d_l \in D \times t_j \in T$. The symbolic scale is converted into an actual probability $P(S_i = true|D = d_l, T = t_j)$ according to the scheme reported in Table 1. The complementary probability is consequentially defined as $P(S_i = false|D = d_l, T = t_j) = 1 - P(S_i = true|D = d_l, T = t_j)$.

answer	$P(S_i = true\|d_l, t_j)$
Always (A)	0.999
Very often (V)	0.8
Often (O)	0.6
Sometimes (S)	0.3
Rarely (R)	0.01
Never (N)	0.001

Table 1
Scale to convert expert knowledge into actual probabilities. How frequently do you observe symptom $S_i = true$, given that you have apples infected by disease d_l at stage t_j?

For categorical variables (i.e., each symptom variable $S_i \in \mathcal{S}$ such that $Val(S_i) = \{s_i^1, s_i^2, \ldots, s_i^m\}$, where $m > 2$), such a process would have been too burdensome for the expert. Thus, we decided to adopt a lighter, yet effective, approach. For each categorical symptom variable $S_i \in \mathcal{S}$, given a specific disease $D = d_l$ at stage $T = t_j$, the expert was invited to simply indicate which values of $Val(S_i)$ are likely to be observed. Furthermore, we agreed on a 3-point symbolic annotation to denote the likelihood of each reported value, namely, *common* (no parenthesis), *less common* (one parenthesis), and *rare* (two parenthesis). The assumption underneath this choice is that many symptom values are never observed under some conditions (i.e., resulting CPTs are sparse) and could be ignored to speed up the elicitation process. In order to convert likelihood annotations into actual probability distribution values we adopted the following heuristic. Please consider a random variable R with $Val(R) = \{a, b, c, d\}$, which is annotated as follows by the the expert: *a: common*, *b: less common*, *c: rare*, and d is ignored; then $P(a) = 2P(b) = 4P(c) = 1.0$ and $P(d) = 0.0$. Furthermore, a small value $\epsilon = 0.001$ is added to each probability value in order to avoid null probabilities, then values are normalized such that $\sum_{r \in Val(R)} P(r) = 1.0$. This process completely defines a probability distribution for the categorical random variable R.

3.3. Recommendation Mechanism

In this section, we detail how a ranked list of recommended diseases (i.e., a diagnosis) is computed after the user provides the feedback on a target apple, answering the questions asked by the system.

The reasoning mechanism of the BN allows to perform the inference, namely, to estimate the posterior probability distribution on a target unobserved variable (i.e., the *Disease* variable D), given any set $\mathbf{S} \in \mathcal{S}$ of observed variables as provided by the user (i.e., the *evidence* $\mathbf{E}$). The evidence set $\mathbf{E}$ is constructed incrementally by the application. At each step, the application requests the user to answer a multiple-choice question, related to a symptom variable $S_i \in \mathcal{S}$. When the user submits the observed state $s_i^j \in Val(S_i)$, *BN-DSSApple* includes the new information into the evidence set, $\mathbf{E} \cup S_i = s_i^j$. At the end, of this elicitation process, the application will have access to the complete information provided by the user on the infected target apple, she wants to diagnose. It is important to mention that the BN inference mechanism is robust to missing values, hence, the user is not forced to provide observations for every symptom variable $S_i \in \mathcal{S}$ in the model. Thus, if the user skips the question related to variable $S_m \in \mathcal{S}$, the evidence set $\mathbf{E}$ will not include an observation for that variable, $S_m \notin \mathbf{E}$. Thus, the goal of the reasoning system is to provide a probability over the set of candidate diseases (i.e., the possible diagnosis). We estimate the posterior probability distribution $P(D|\mathbf{E})$ through an algorithm called *loopy belief propagation* [11]. The loopy belief propagation is an approximate message-passing method to perform inference on graphical models. In few words, the algorithm iteratively updates the marginal distribution $P(N)$ of a node $N \in \mathcal{G}$, by updating the outgoing message, at the current iteration, from the node N to each of its neighbors $\mathbf{V} \in \mathcal{G}$ in terms of the previous iteration's incoming messages from $\mathbf{V}$.

In our recommendation engine, after completing the evidence collection process for a target apple a, the posterior probability computed by the BN when evidence $\mathbf{E}$ is provided, is considered as a *diagnosis score* $s(d_i)_a$ for each disease $d_i \in D$. Namely, this probability distribution represents the confidence of the system over each disease $d_i \in D$ being the correct diagnosis for the target apple a. More formally, given the provided evidence set $\mathbf{E} = \{S_1 = s_1^o, S_2 = s_2^p, \ldots S_l = s_l^q\}$, defined as the set of each observed state $s_i^j \in Val(S_i)$ for each random variable $S_i \in \mathcal{S}$, the diagnosis score related to target apple a for disease $d_i \in D$ is computed as:

$$s(d_i)_a = P(D = d_i|\mathbf{E}) \tag{1}$$

The ranked list of the k suggested diseases $R^k = \{d^1, d^2, \ldots, d^k\}$ shown to the user is then based on the score for each disease, such that $s(d^i) \geq s(d^{i+1})$. The parameter k controls for the flexibility of the system to show more or less recommended diseases to the user. In our evaluation, the parameter is fixed to $k = 3$.

3.4. Transferability and Likelihood Evidence

In knowledge-based modeling, but also with standard supervised learning, we often face the problem of transferring such a model on a different environment (i.e., providing external validity). This type of situation is referred to as the *transferability* problem [12, 13]. For instance, it might be difficult to allow a vast set of users, with different expertise level, to effectively exploit a diagnostic expert model, based on domain-specific knowledge. In our application, the knowledge base of *BN-DSSApple* has been built with the information derived from domain literature and empirical knowledge of a domain expert. Nevertheless, different sets of users, with less experience in the field, might perceive the same attributes (i.e., the symptoms) in a different way. In fact, the user perception is mediated by her personal experience and specific knowledge biases. This mismatch invalidates the effectiveness and hence the diagnostic performance of *BN-DSSApple*. In this section, we formalize the problem of transferability and we propose a practical solution to bridge the gap between the expert model and the user perception.

In our scenario, the transferability problem is defined as the mismatch between the BN probability distributions (CPTs) defined by the expert, and the probability distributions derived by the usage of the system. Formally, the expert during the knowledge elicitation phase (as described in Section 3.2) implicitly defined a complete set of probability $\mathscr{P}^{exp} = \{P(\mathbf{S}|D = d_1), P(\mathbf{S}|D = d_2), \ldots P(\mathbf{S}|D = d_n)\} \subseteq \mathscr{P}$, for each set of symptom random variables $\mathbf{S}$, given the target disease $D = d_i$. At testing time, the users of our application produced a set of u observations $\mathscr{E} = \{(\mathbf{E}_1, d_1), (\mathbf{E}_2, d_2), \ldots (\mathbf{E}_u, d_u)\} \subseteq \mathscr{S} \times \mathscr{D}$, where $\mathbf{E}_i = \{S_1 = s_1^o, S_2 = s_2^p, \ldots S_l = s_l^q\}$, represent the evidence provided by a user during the i-th diagnosis session, as a set of instantiations of symptom variables, and d_i is the corresponding ground-truth disease. These set of user observations define a different set of probabilities $\mathscr{P}^{usr} = \{P(\mathbf{S}|D = d_1), P(\mathbf{S}|D = d_2), \ldots, P(\mathbf{S}|D = d_n)\} \subseteq \mathscr{P}$, which is generally different from the one defined by the expert, $\mathscr{P}^{usr} \neq \mathscr{P}^{exp}$. The problem becomes the one to find a transferability function $T(.)$ to be applied to the expert model such that $\mathscr{P}^{usr} = T(\mathscr{P}^{exp})$.

The problem of transferability is long-lasting in machine learning and statistics and it has been addressed in causal terms, referred to as *transportability* [12, 14], as well as in statistical terms, in the context of supervised learning, where it is also known as *covariate shift* or *sample selection bias* [15, 16]. One of the most common approaches applies a direct correction to the learned probability distribution based on the estimates on the testing set [13]. Specifically inspired by the work presented in [17], we proposed a methodology, referred to as *likelihood evidence* and tailored to our BN-based application, to correct the expert-defined distribution $\mathscr{P}^{exp}$ towards the one derived by users $\mathscr{P}^{usr}$. We define the likelihood evidence (or likelihood finding) for each random symptom variable $S_i \in \mathscr{S}$ of our *BN-DSSApple*. Specifically, when a symptom variables S_i is observed and thus instantiated by a user, we assume that a certain degree of uncertainty is associated with it (i.e., the difference of knowledge and expertise between the user and the expert). We define the actual user observation with another random variable O_i, such that $Val(O_i) = Val(S_i)$, to distinguish it from the variable as it should be observed by an expert S_i. We represent the uncertainty degree with a likelihood ratio $L(S_i)$, formally defined as:

$$L(S_i = s_i^j) = P(O_i = o_i^l | S_i = s_i^j) \tag{2}$$

which represents the probability of a user observing value $o_i^l \in Val(O_i)$ given that, in the same situation, the expert would have observed $s_i^j \in Val(S_i)$. Thus, we enrich our BN by adding, for each symptom variable S_i, a virtual likelihood evidence node O_i that encodes the likelihood ratio $L(S_i)$, with $pa(O_i) = \{S_i\}$. The added set of random variable $\mathscr{O} = \{O_1, O_2, \ldots O_t\}$ is now the one observed by the user while providing the evidence $\mathbf{E}$ on the questions asked by the application, while the random variables in $\mathscr{S}$ become hidden. We finally need to define a new set of conditional probability tables $P(O_i|S_i)$ for each pair $(S_i, O_i) \in \mathscr{S} \times \mathscr{O}$. We adopt a direct estimation of these probabilities from the observed interactions of users with a set of apples $\mathscr{A}$ for which we know the actual observed value by the expert. Namely, for each state $s_i^j \in Val(S_i)$ of each variable $S_i \in \mathscr{S}$ we define a subset of $\mathscr{A}_{s_i^j} \subseteq \mathscr{A}$ for which the value of the symptoms variable S_i observed by the expert is $S_i = s_i^j$. Thus, the conditional probability of the observed value $O_i = o_i^l$ by the users is defined as:

$$P(O_i = o_i^l | S_i = s_i^j) = \frac{1}{|\mathscr{A}_{s_i^j}|} \sum_{a_i \in \mathscr{A}_{s_i^j}} \mathbb{1}_{a_i}(o_i^l) \tag{3}$$

where $\mathbb{1}_{a_i}(o_i^l)$ is an indicator function which is equal to 1 if the user observed $O_i = o_i^l$ in apple a_i, and 0 otherwise. The defined conditional probability for the likelihood ratio is also referred as *consensus* among expert and users.

4. Experiments

4.1. User Study Evaluation

We conducted a large user study to evaluate the effectiveness of *BN-DSSApple* in recommending the correct diagnosis. Specifically, we divided the user study into three distinct phases to test the system behaviour under different circumstances. The task submitted to the users involved in our study was the same in all cases. The user received a "bucket" of infected apples, for which she had to find the correct diagnosis leveraging *BN-DSSApple*. Each target apple was simulated as a set of two high-definition photos depicting an internal and an external view of the target apple, and for which the ground-truth disease was collected in lab by genome sequencing. In each diagnostic round, the user had to carefully inspect the target apple and interact with the system by providing information (i.e., the evidence) about the symptoms and signs she was able to identify on the apple. At the end, *BN-DSSApple* returned a ranked list of three suggested diagnosis, i.e., the three diseases with the highest posterior given the available evidence, as computed by the BN. The three phases of the presented study differed in the number of users, their expertise level, and the number of distinct target apples involved. In details, we performed:

- **Single Expert Study (SES)**: a domain expert (the one which collaborate in the construction of the BN) interacted with the system to diagnose 21 target apples in a time-span of around 2 weeks.
- **Single User Study (SUS)**: a single user (a MSc student in Biology), interacted with the system during the course of an internship, lasting around 3 months, to diagnose 131 target apples.
- **Multiple Users Study (MUS)**: a group of 11 students of a Phytopatology class interacted with the system to diagnose a bucket of 7 target apples each. The apples were randomly sampled from the same set of 21 apples used for SES. The activity lasted for a total of 4 hours.

In Table 2 we summarize the different characteristics of the three user studies performed.

	# users	*expertise*	*# apples*	*time-span*
SES	1	high	21	2 weeks
SUS	1	high-medium	131	3 months
MUS	11	medium-low	21	4 hours

Table 2
Characteristics of the three user studies: Single Expert Study (SUS), Single User Study (SUS), and Multiple User Study (MUS).

4.2. Results

In Table 3 we report the results of the three user studies in terms of *recall@k*. To better formalize this metric, please consider a situation in which a set N of n diagnosis is performed by *BN-DSSApple*. The set N is composed by n ranked lists of recommended diagnosis, namely $N = \{R^k_{a_1}, R^k_{a_2}, \dots R^k_{a_n}\}$, where a_i represents the i-th apple processed by the system. A generic $R^k_{a_i} = \{d^1_{a_i}, d^2_{a_i}, \dots, d^k_{a_i}\}$ is a ranked list of k suggested diagnosis $d^j_{a_i}$ for apple a_i with a specific ground truth disease t_{a_i}. Thus, we formally define *recall@k* as:

$$recall@k = \frac{1}{n} \sum_{R^k_{a_i} \in N} \mathbb{1}_{R^k_{a_i}}(t_{a_i}) \tag{4}$$

Where the function $\mathbb{1}_{R^k_{a_i}}(t_{a_i})$ is an indicator function which is equal 1 if $t_{a_i} \in R^k_{a_i}$ and 0 otherwise.

	SES	**SUS**	**MUS**	**ZeroR**
recall@1	.905	.489	.286	.143
recall@2	1.	.656	.403	.286
recall@3	1.	.763	.571	.429

Table 3
Recall@k for the three user studies performed, Single Expert Study (SES), Single User Study (SUS), and Multiple Users Study (MUS). The ZeroR benchmark is also reported.

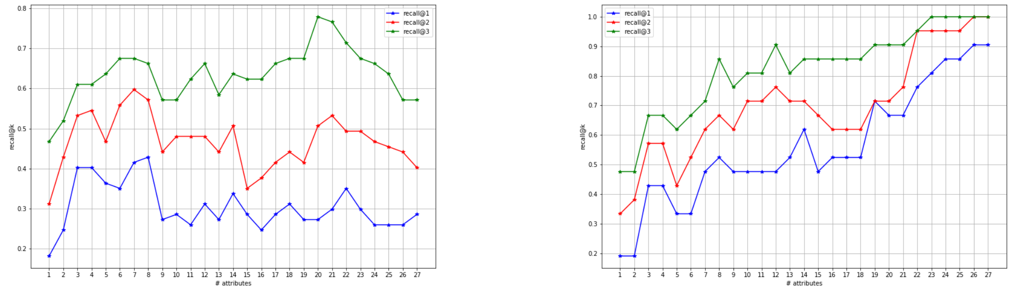
From the results presented in Table 3 we highlight how the theoretical effectiveness of the *BN-DSSApple* model is very high. Specifically, an expert user (SES), with strong knowledge in the domain of post-harvest diseases of apples and a good capability of correctly identify symptoms on a diseased apple, is able to reach a recall@1 above the 90%. The performance of the system increases up to 100% of recall when evaluated at a larger cut-off of suggested diseases. Of course, we have to consider that in the SES evaluation, we are in the ideal situation where the expert user knows exactly how to look and evaluate the symptoms requested by *BN-DSSApple*. A more realistic situation is depicted by the SUS evaluation. In this situation, a single user with a medium-high level of expertise had months of time to interact with the system by evaluating a very large set of apples (131). The performance of the system for the recall@1 are still convincing (49%), i.e. correct disease identification by half of all diagnoses. The other metrics testify how the system is not able to scale-up well for further cut-off of recall, achieving 66% of recall@2 and 76% of recall@3 (the correct disease is within the first 3 recommendations in 3/4 of the cases). Finally, *BN-DSSApple* showed some limits in the situation where the users have a limited expertise and training, and a limited amount of time (few hours) to use the system as in the MUS evaluation. In addition to the time and skill aspect, also less intrinsic motivation to interact as accurate as possible with the system could be a partial explanation for the deviation. In this case, the measured recall of the system is significantly lower than the one of the two previous evaluations. Particularly, the recall@1 doesn't reach the 30%, while the best result is achieved by the recall@3 with a value of 57% (slightly more than half of the diagnosis include the correct disease in the top-3 recommendations). Nevertheless, despite the poor performances of *BN-DSSApple* in MUS, the collected results are still superior to the ZeroR benchmark, namely, a classifier which always suggest the class with a priori higher probability. Important to notice that the reported results for ZeroR are related to the situation in which the class (ground-truth disease) distribution is perfectly balanced, like for SES and MUS. In the comparison with ZeroR, MUS evaluation for *BN-DSSApple* shows the double of recall@1 (28.6% against 14.3%), while recall@2 and recall@3 are closer but still significantly better (+12% and +14%, respectively). The main cause of this mismatch of performances among expert and averaged users can be identified in the problem of transferability of a knowledge-aware model. In the remaining of this section, we are going to empirically analyze and explain such a phenomenon, and test possible solutions to correct and alleviate it.

Foremost, we want to understand the impact of each expert-defined attribute in the model. In Table 4 we report the ranked list of attributes, based on the likelihood ratio (i.e., consensus) computed between users of MUS and the expert of SES (which we consider as a ground-truth) in identifying the symptoms on the same set of 21 target apples. It is interesting to notice how the users are effective in identifying the principal symptoms and signs, presented by the application as boolean variables. Namely, *Sclerotia* (99%), *Rot* (96%), and *Spot* (95%) present a very high level of agreement with the domain expert, while *Mycelium_spore* (81%) and *Halo* (78%) receive an high consensus. Vice versa, some qualitative attributes related to the appearance or the consistency of the lesion and the rot are among the hardest to be correctly recognized by the users (i.e., they show a poor consensus with the expert). For example, *Lesion_appearance* and *Rot_texture_pressure* achieve a consensus below the 50%, while *Lesion_margin*, *Lesion_area*, and *Rot_texture_opaque* are below 65%. Nevertheless, other categorical variables more related to quantitative aspects of the lesion are easier for the users to be spotted. This is the case of the variables *Lesion_size*, *Lesion_surface*, *Lesion_form*, and *Lesion_crack* which show a consensus between 84% and 79%. Finally, it is interesting to notice the behavior of the variables of the *Lesion origin* category. Most of them are quite easy to be identified by the user, with a consensus above the 90% with the expert. Nevertheless, two of them, namely *Wound* and *Lenticel*, are equally difficult to be recognized with a consensus of around 59%. This is probably due to the fact that the two origins might be perceived as quite similar and could be confused, without a careful inspection of the apple skin.

rank	*attribute*	*consensus*
1	Sclerotia	0.988
2	Calyx	0.985
3	Rot	0.964
4	Spot	0.950
5	Stalk	0.926
6	Core	0.917
7	Spore_distribution	0.872
8	Lesion_size	0.837
9	Lesion_surface	0.837
10	Number_lesions	0.817
11	Mycelium_spore	0.809
12	Lesion_form	0.792
13	Lesion_crack	0.790
14	Halo	0.782
15	Rot_shape	0.760
16	Rot_texture_dry	0.755
17	Halo_colour	0.750
18	Rot_margin	0.740
19	Spore_colour	0.731
20	Spore_origin	0.694
21	Lesion_margin	0.636
22	Lesion_area	0.623
23	Rot_texture_opaque	0.607
24	Wound	0.594
25	Lenticel	0.588
26	Lesion_appearance	0.417
27	Rot_texture_pressure	0.321

Table 4
Attributes ranking based on the rate of agreement (i.e., consensus) of the users of MUS with the domain expert of SES.

In Figure 3 we plot the recall@k achieved by *BN-DSSApple* for MUS and SES, by incrementally selecting the attributes based on the consensus ranking reported in Table 4. On the x-axis, we report the number of attributes in each model configuration. Namely, the *i*-th value represents the BN model built with the attribute set $\mathscr{A}_i = \{a^1, a^2, \dots a^{i-1}, a^i\}$, where the rank j of attribute a^j is defined by expert consensus, as reported in Table 4. From the graph in Figure 3a for MUS evaluation, we immediately notice how the model achieves the best performances for recall@1 and recall@2 with around 8-9 attributes. A larger set of attributes is detrimental, causing a drop of recall of at least 10% in both situations. Interesting to notice how these performances seem to recover with the models based on 21-22 attributes, without reaching the optimal level. In fact, for the recall@3 metric the global optimum is achieved by the model with



(a) (b)

Figure 3: Recall@k by incremental selection of attributes based on ranking of Table 4 for MUS (a) and SES (b).

20 attributes, with a significant improvement of around 10% on the smaller attribute set configurations. Opposite considerations emerge from the graph in Figure 3b for SES evaluation. In this case, the recall@k metrics are linearly correlated to the number of attributes, and the best performances are always achieved with the full set of attributes. This means that the expert is able to correctly instantiate even the harder variables, by understanding the status of an infected apple. Furthermore, this "hard-to-recognize" attributes are necessary to significantly improve the diagnostic effectiveness of the model and reach the highest performances in term of recall@k. For instance, in both recall@2 and recall@3 the BN model registers around +20% improvement by considering the full set of 27 attributes instead of just considering 21 attributes (i.e., by discarding the 6 "hardest" attributes, with lowest consensus).

	BN	**TRAIN-BN**	**BEST-ATTR**	**LH-EV**
recall@1	.286	.312	.429 (8)	.351
recall@2	.403	.468	.597 (9)	.623
recall@3	.571	.636	.779 (20)	.766

Table 5
Recall@k for MUS when applying the plain BN-DSSApple (BN), the trained BN-DSSApple on MUS data (TRAIN-BN), the incremental best attribute selection (BEST-ATTR), and the BN-DSSApple with likelihood evidence (LH-EV). In BEST-ATTR column, we report the results for the optimal attribute set, with the number of selected attributes in parenthesis.

Finally, in Table 5 we compare the recall@k results for the MUS evaluation of the improved versions of the BN model, in order to cope with the transferability problem discussed in Section 3.4. Firstly, the smallest improvement is provided by the trained BN model (dubbed as TRAIN-BN), where the parameters are fine-tuned on MUS data with the Maximum Likelihood Estimation (MLE) algorithm. The recall@1 improvement is marginal (around +2.5%), while recall@2 shows a +6.5% with respect to the plain BN model. We already commented the large improvements achieved by selecting the optimal attribute set (BEST-ATTR model), whereas the gain in recall is between +14% and +21%. Of course, this analysis is derived a posteriori, where the optimal number of attributes is fixed after the evaluation. For this reason, the achievement of the model equipped with likelihood evidence (LH-EV, methodology detailed in Section 3.4, where expert ground-truth data are derived from SES) is even greater. For recall@1 the LH-EV outperforms TRAIN-BN of around +4%, while being inferior to BEST-ATTR by around -8%. For recall@2, instead, the likelihood evidence achieves the best result outperforming also BEST-ATTR by a +2.5%. Finally, for recall@3 the LH-EV model significantly outscores TRAIN-BN (+13%), while being comparable with the results of BEST-ATTR.

5. Conclusions

This case study focused on knowledge elicitation and construction as well as discussed the application of likelihood evidence to enhance performance and transferability of the knowledge-based recommendation system *BN-DSSApple*. Major limitations of the presented approach concern the fact that the knowledge base is fully based on qualitatively probability elicitation from a single human expert. Furthermore, transferability problem of the crafted BN must be additionally investigated. Further development of the method to other domains as well as additional testing is required. Currently, deployment for real-life evaluation is ongoing. In future work, the integration of additional evidence like microscopic images of fungal spores will be considered.

References

[1] T. B. Sutton, H. S. Aldwinckle, A. Agnello, J. F. Walgenbach (Eds.), Compendium of apple and pear diseases and pests, 2 ed., APS press, 2014.

[2] P. Maxin, M. Williams, R. W. Weber, Control of fungal storage rots of apples by hot-water treatments: A northern european perspective, Erwerbs-Obstbau 56 (2014) 25–34.

[3] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, Adaptive computation and machine learning, MIT Press, 2009. URL: https://books.google.co.in/books?id=7dzpHCHzNQ4C.

[4] U. B. Kjaerulff, A. L. Madsen, Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis, 1st ed., Springer Publishing Company, Incorporated, 2010.

[5] A. Niederkofler, S. Baric, G. Guizzardi, G. Sottocornola, M. Zanker, Knowledge models for diagnosing postharvest diseases of apples, in: Proceedings of the Joint Ontology Workshops 2019 Episode V: The Styrian Autumn of Ontology, Graz, Austria, September 23-25, 2019, volume 2518 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2518/paper-ODLS6.pdf.

[6] M. Zanker, M. Jessenitschnig, W. Schmid, Preference reasoning with soft constraints in constraint-based recommender systems, Constraints 15 (2010) 574–595.

[7] M. B. Messaoud, P. Leray, N. B. Amor, Semcado: A serendipitous strategy for causal discovery and ontology evolution., Knowl.-Based Syst. 76 (2015) 79–95. URL: http://dblp.uni-trier.de/db/journals/kbs/kbs76.html#MessaoudLA15.

[8] A. M. Kalet, J. N. Doctor, J. H. Gennari, M. H. Phillips, Developing bayesian networks from a dependency-layered ontology: A proof-of-concept in radiation oncology, Medical Physics 44 (2017) 4350–4359. doi:10.1002/mp.12340.

[9] S. Fenz, An ontology-based approach for constructing bayesian networks, Data Knowl. Eng. 73 (2012) 73–88. URL: http://dx.doi.org/10.1016/j.datak.2011.12.001. doi:10.1016/j.datak.2011.12.001.

[10] L. C. van der Gaag, S. Renooij, C. L. M. Witteman, B. M. P. Aleman, B. G. Taal, How to elicit many probabilities, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI'99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, p. 647–654.

[11] A. T. Ihler, J. W. Fischer III, A. S. Willsky, Loopy belief propagation: Convergence and effects of message errors, J. Mach. Learn. Res. 6 (2005) 905–936.

[12] J. Pearl, E. Bareinboim, Transportability of causal and statistical relations: A formal approach, in: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, ICDMW '11, IEEE Computer Society, USA, 2011, p. 540–547. URL: https://doi.org/10.1109/ICDMW.2011.169. doi:10.1109/ICDMW.2011.169.

[13] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, Knowl. Based Syst. 80 (2015) 14–23.

[14] A. Subbaswamy, S. Saria, Counterfactual normalization: Proactively addressing dataset shift using causal mechanisms, in: R. Silva, A. Globerson, A. Globerson (Eds.), 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, volume 2, Association For Uncertainty in Artificial Intelligence (AUAI), 2018, pp. 947–957. 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018 ; Conference date: 06-08-2018 Through 10-08-2018.

[15] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, B. Scholkopf, Correcting sample selection bias by unlabeled data, in: Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS'06, MIT Press, Cambridge, MA, USA, 2006, p. 601–608.

[16] M. Sugiyama, S. Nakajima, H. Kashima, P. v. Bünau, M. Kawanabe, Direct importance estimation with model selection and its application to covariate shift adaptation, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07, Curran Associates Inc., Red Hook, NY, USA, 2007, p. 1433–1440.

[17] A. B. Mrad, V. Delcroix, S. Piechowiak, P. Leicester, M. Abid, An explication of uncertain evidence in bayesian networks: likelihood evidence and probabilistic evidence - uncertain evidence in bayesian networks, Appl. Intell. 43 (2015) 802–824. URL: https://doi.org/10.1007/s10489-015-0678-6. doi:10.1007/s10489-015-0678-6.