Considering Currency in Decision Trees in the Context of Big Data

Completed Research Paper

Diana Hristova
Department of Management Information Systems
University of Regensburg
Universitätsstraße 31
93053 Regensburg, Germany
Diana.Hristova@wiwi.uni-regensburg.de

Abstract

In the current age of big data, decision trees are one of the most commonly applied data mining methods. However, for reliable results they require up-to-date input data, which is not always given in reality. We present a two-phase approach based on probability theory for considering currency of stored data in decision trees. Our approach is efficient and thus suitable for big data applications. Moreover, it is independent of the particular decision tree classifier. Finally, it is context-specific since the decision tree structure and supplemental data are taken into account. We demonstrate the benefits of the novel approach by applying it to three datasets. The results show a substantial increase in the classification success rate as opposed to not considering currency. Thus, applying our approach prevents wrong classification and consequently wrong decisions.

Keywords: Decision trees, Currency, Data quality, Big data mining

Introduction

In the current information age companies around the world store huge volumes of quickly changing, distributed, heterogeneous data (IBM Institute for Business Value 2012) to support decision making in areas such as marketing, investment, risk management, production, health care, etc. (Economist Intelligence Unit 2011; Giudici and Figini 2009; Hems et al. 2013; Ngai et al. 2009; Yue 2007). Such data is often called big data and is characterized by the three Vs i.e. Volume, Velocity and Variety. Data streams are a typical example for big data as they are characterized by both high volume and high velocity. Stored (big) data has business value only, if it is analyzed to discover new patterns. Thus, in a recent survey by the IBM Institute for Business Value (2012) more than 75% of the participants "with active big data efforts" (p.12) stated that they apply data analytics techniques to derive valuable insights from it. However, not only data quantity, but also its quality matters.

Stored data may be outdated due to improper update frequency (e.g. address data), attribute values may be missing due to malfunctioning sensors or be inaccurate, because of privacy protection reasons (Aggarwal et al. 2013; Liang et al. 2010) or data integration problems. As a result, if such data is used for analysis without taking its quality into account, false patterns may be identified and thus wrong decisions may be made ("Garbage in, garbage out."). According to a survey by Forbes (2010), most of the participants estimate the cost of poor data quality to more than $5 million annually. The problem is especially evident in the context of big data (Fan 2013; Li et al. 2012; Yang et al. 2012) and to demonstrate its importance IBM has added a fourth V to the characteristics of big data, which stands for Veracity (IBM Institute for Business Value 2012) and represents the uncertainty due to low data quality.

Data quality can be defined as "the measure of the agreement between the data views presented by an information system and that same data in the real world" (Orr 1998, p. 67). Data quality is a multi-dimensional concept consisting of dimensions such as currency, accuracy, completeness, etc. (Wang and Strong 1996). In this paper we focus on currency, as one of the most important among them (Experian QAS 2013; Redman 1996). We define currency as *the degree to which a previously correctly stored attribute value still corresponds to its real world counterpart at the time of analysis*. This implies that low currency causes uncertainty regarding the correspondence between the stored and the real-world attribute value which should be taken into account in data mining.

The process of Knowledge Discovery in Databases consists of five main steps: selection, pre-processing, transformation, data mining and interpretation/evaluation (Fayyad et al. 1996). Among them, (big) data mining is the application of analytic methods to search for new patterns in the pre-processed and/or transformed data. The aim of classification is to assign a stored instance characterized by the values for a set of independent attributes to one of a predefined set of classes of a given dependent attribute. Decision trees are one of the most commonly applied classification methods (Tsang et al. 2011) due to their simple interpretation (i.e. "white box") and efficiency. They consist of a set of non-leaf and leaf nodes, where each non-leaf node is characterized by a splitting independent attribute condition and each leaf node is characterized by a class of the dependent attribute (Vazirigianniset al. 2003; Witten and Frank 2005). Common applications of decision trees are credit scoring (Koh et al. 2006), fraud detection (Pathak et al. 2011), medical diagnosis (Azar and El-Metwally 2013), and sensor networks (Yang et al. 2012). In all of these applications low currency causes uncertainty in the analyzed data, which should be taken into account in the classification process. Blake and Mangiameli (2011) confirm this point by empirically showing that currency has an effect on the accuracy of classification methods.

In the literature, incorporating data uncertainty in (big) data mining is called uncertain (big) data mining (Aggarwal and Yu 2009; Chau et al. 2006; Leung and Hayduk 2013; Tsang et al. 2011: Yang and Fong 2011). For example, due to the emergence of the Internet of Things, data streams are becoming one of the most common examples of big data in practice and as a result a number of approaches have been developed for the mining of data streams (Aggarwal 2013) and in particular of uncertain data streams. Typical for the uncertain big data mining literature is that it does not focus on the source of uncertainty, but rather takes it as given without discussing how it can be derived. However, modelling this uncertainty properly (e.g. deriving a probability distribution) with the corresponding interpretation is just as important for real-world applications as incorporating it in the classification process.

In this paper we develop a probability-theory based approach for considering currency of the attribute values of stored instances when classifying them in existing decision trees. The advantage of applying

probability theory is that it allows for a mathematically-sound modelling of uncertainty. Our approach addresses the Volume, Velocity and Veracity characteristics of big data. It addresses the volume by considering currency in an efficient way; velocity by requiring less detailed historical data than existing approaches for currency; and veracity by dealing with data uncertainty. Moreover, our approach is universal, because it is independent of the particular decision tree classifier, and adaptable to the given context of application and to the structure of the decision tree.

Thus, our approach aims at extending both the data quality and the uncertain big data mining literature and to close the gap between them. The first stream of research is extended by proposing an efficient way for measuring currency in decision tress which is applicable in the context of big data. The second stream of research is extended by demonstrating how the uncertainty, which is assumed to be given in existing works, can be measured in an interpretable, efficient and context-specific way.

The practical relevance of our approach can be illustrated by an example from the field of Customer Relationship Management. Consider a financial service provider who would like to conduct a campaign for winning new customers based on their annual income. The annual income of a customer depends strongly on other personal characteristics such as education, age, and employment status and this relationship can be modelled as a decision tree. If the personal characteristics are outdated (e.g. because they were stored long before the time of the campaign), the customer may be classified in the wrong income class and thus offered the wrong product resulting in losses for the company.

Another example, which is a typical big data case, comes from the field of sensor measurements. Consider a dataset for handicapped individuals, whose movements, location, speed, etc. are measured by sensors integrated in their clothing. Based on the measurements and a decision tree classifier, their current activity is determined (Yang et al. 2012). As a result, emergency situations (e.g. an accident) can be detected and thus corresponding measures derived (e.g. sending an ambulance). If the data measured by the sensors is outdated, for example due to a delayed transmission, and if this is not considered in the classification, an emergency case can be detected either with a strong delay or not at all, causing serious consequences for the patient.

The paper is structured as follows. In the next section we give a literature review and provide the required background on the topic. In the third section our approach for incorporating currency in decision trees in the context of big data is presented. In the fourth section it is evaluated based on three different datasets representing the two applications above. In the final section main conclusions are drawn and limitations and paths for future research are discussed.

Related Work and Background

Since we extend both the literature on modelling currency in decision trees and the one on uncertain big data mining with decision trees, in the following we first present the main findings in these two streams of research as well as our contribution to them.

Modelling Currency in Decision Trees

As mentioned above, data quality is a multi-dimensional concept including characteristics such as currency, accuracy, completeness (Wang and Strong 1996). In the context of data mining, lower data quality can be seen as causing two types of uncertainty: existential and attribute-level uncertainty (Aggarwal and Yu 2009). Existential uncertainty represents the uncertainty whether an attribute value does or does not exist (i.e. completeness), while attribute-level uncertainty stands for the uncertainty regarding the existing attribute value which is possibly of poor quality (i.e. concerning accuracy, currency, etc.).

Existential uncertainty is well-studied by researchers (Dasu and Johnson 2003; Hawarah et al. 2009; Quinlan 1986; Witten and Frank 2005; Yang et al. 2012). The reason for this is that it is rather straightforward to measure and thus to identify it. To reduce existential uncertainty, missing attribute values are either replaced (i.e. imputed) based on point estimation (e.g. by the mean of the non-missing attribute values) or represented as a probability distribution over the domain of the attribute (Quinlan 1986). Already Quinlan (1986) proposes a probability-theory based approach for considering data completeness in classifying data instances in decision trees.

Attribute-level uncertainty is a relatively new topic of research. A few authors have developed metrics for measuring accuracy (Fisher et al. 2009), currency (Ballou et al. 1998; Blake and Mangiameli 2011; Even and Shankaranarayanan 2007; Heinrich and Klier 2009; Heinrich and Klier 2011; Li et al. 2012; Wechsler and Even 2012) and other attribute-level data quality dimensions (Alpar and Winkelsträter 2014; Mezzanzanica et al. 2012). Since the focus of this paper is on currency, we concentrate in the further discussion on it.

Based on our definition of currency above, it is very natural to interpret currency as probability. This is also in accordance with the literature. For example, Heinrich and Klier (2011, p. 6) interpret currency as "…the probability that an attribute value stored in a database still corresponds to the current state of its real world counterpart…" i.e. as the probability that the stored attribute value is up-to-date. Thus, we can estimate the currency of a stored attribute value with the help of historical data, which can be either publicly available (e.g. Federal Bureau of Statistics 2013) or company-internal data. To demonstrate the idea, consider the attribute $Marital\ status$ which can take the values $\{single, married, divorced, widowed\}$. If the stored attribute value is $single$, then the currency of this attribute value will be the probability that a person, who was single at the time of storage, is still single at the time of classification. Thus, currency can be defined as the conditional probability $P(single\ in\ t_1 |\ single\ in\ t_0)$, where t_1 is the time of classification and t_0 is the time of storage. Based on the Bayes' theorem (Berger 2010), the currency for the attribute value $single$ can be derived as:

$$P(single\ in\ t_1 |\ single\ in\ t_0) = \frac{P(single\ in\ t_1\ AND\ single\ in\ t_0)}{P(single\ in\ t_0)} \quad (1)$$

where the operator AND stands for the intersection of the two events. To calculate these probabilities, we follow the frequentist perspective (Berger 2010) according to which a probability "…is the proportion of the time that events of the same kind will occur in the long run." (Miller et al. 2004, p. 24). This implies that the probability of a certain event is represented by the percentage of instances in the population characterizing this event. Thus,

$$P(single\ in\ t_1 |\ single\ in\ t_0) = \frac{|single\ in\ t_1\ AND\ single\ in\ t_0||instances|}{|single\ in\ t_0||instances|} = \frac{|single\ in\ t_1\ AND\ single\ in\ t_0|}{|single\ in\ t_0|} \quad (2)$$

where $|single\ in\ t_0|$ stands for the number of instances in the population who were single at the time of storage. Analogously, $|single\ in\ t_1\ AND\ single\ in\ t_0|$ represents the number of instances in the population who were single both at the time of storage and at the time of classification. Finally, $|instances|$ stands for the total number of instances in the population.

Calculating currency in this manner provides context-free results, which are independent of the other attribute values of a stored instance. This has its advantages and is in accordance with existing metrics for currency based on probability theory (Heinrich and Klier 2009, 2011; Li et al. 2012; Wechsler and Even 2012). However, as the literature has shown, the context of application plays a role for the measurement of currency. Even and Shankaranarayanan (2007) develop context-specific metrics for different data quality dimensions including currency. They use a so called data utility measure, which reflects the business value in the specific context and give a contextual interpretation of currency as "The extent to which outdated data damages utility."(p. 83). Their metric for currency is thus represented as the relative decrease in utility due to outdated data for a particular *context*. Similarly, Heinrich and Klier (2009) propose the use of supplemental data for a more precise estimation of currency (defined as timeliness by Heinrich and Klier (2009)) adapted for the particular context of application. Supplemental data is different from metadata such as storage age in that it is attribute-value specific. Based on this approach, we consider the context of application by determining the conditional probabilities, so that the probability that an attribute value is up-to-date depends on the values of the supplemental data.

To demonstrate this idea, consider again the attribute $Marital\ status$ from above. The currency of the attribute value $single$ can be derived much more precisely, if additional information about the personal characteristics of the individual is considered. For example, the age and the gender of the person would have an influence on the changes in his/her marital status. Given a single, 20-year-old (at the time of storage) female, we can compute the currency of her marital status as the conditional probability $P(single\ in\ t_1 | single\ in\ t_0, female, 20\ in\ t_0)$, which after applying the Bayes' theorem can be rewritten as:

$$P(single\ in\ t_1 |\ single\ in\ t_0, female, 20\ in\ t_0) = \frac{P(single\ in\ t_1 AND\ single\ in\ t_0, female, 20\ in\ t_0)}{P(single\ in\ t_0, female, 20\ in\ t_0)} \quad (3)$$

Again, based on the frequentist perspective, (3) can be rewritten as:

$$P(single\ in\ t_1 |\ single\ in\ \ t_0, female, 20\ in\ t_0) = \frac{|single\ in\ t_1 AND\ single\ in\ \ t_0, female, 20\ in\ t_0|}{|single\ in\ \ t_0, female, 20\ in\ t_0|} \quad (4)$$

The result in (4) will deliver a much more precise indication about the currency of the stored attribute value than the one in (2) as it considers additional information about the stored instance. However, such a detailed historical data is not always available in reality (Velocity) and determining the conditional probability for all values can be computationally rather intensive (Volume). These challenges are addressed by our approach.

In this paper we aim to draw on the findings in the literature on modelling currency by 1) measuring currency based on the interpretation by Heinrich and Klier (2011) and 2) considering supplemental data in the measurement process. We extend the literature by a) proposing measurement methods, which are more efficient and do not require such a detailed historical data, and by b) demonstrating how the measured currency can be considered during the classification of stored instances in existing decision trees. In particular a) is very relevant in the context of big data. b) is partly based on the ideas from the field of uncertain big data mining. In the next subsection we discuss existing approaches in this field by focusing on decision trees.

Uncertain Big Data Mining for Decision Trees

In order to analyze uncertain big data, traditional data mining approaches need to be modified to consider both the characteristics of data uncertainty and these of big data. On the one hand, data uncertainty is considered in the field of uncertain data mining. However, most of these approaches are not suitable for big data because the volume and velocity of big data require more efficient methods. On the other hand, the field of big data mining concentrates on the efficiency of the algorithms (especially for data streams as one of the most common applications), but the methods there do not consider data uncertainty. The approaches in the field of uncertain big data mining can thus emerge either i) from the field of uncertain data mining by improving the efficiency of the methods, or ii) from the field of big data mining by increasing the robustness to uncertainty, or iii) by combining both of them. Figure 1 presents this idea. Since our approach is part of the first group, in the following we first describe the ideas in the field of uncertain data mining for decision trees.

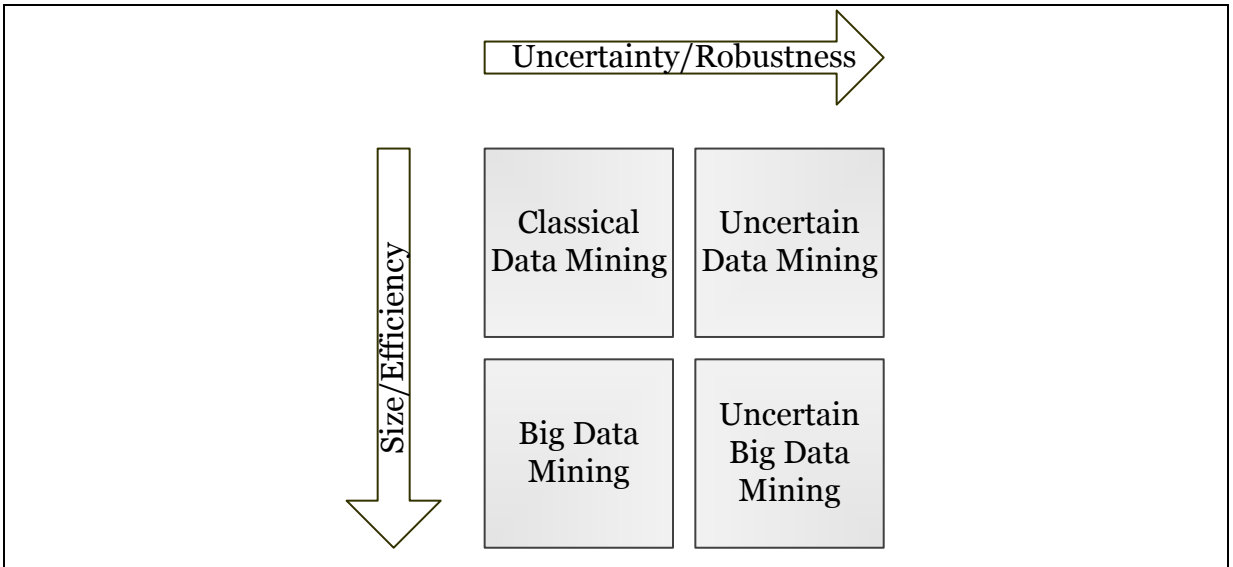

Figure 1. Literature streams

Stream of research i)

A number of authors have developed approaches that modify decision trees to work with attribute-level uncertainty, usually based on probability theory. One major group is represented by the works of Qin et al. (2009) and Tsang et al. (2011). In these approaches the independent attribute values of the data instances are assumed to be uncertain and described by a probability distribution (Qin et al. 2009). For each instance, these distributions are propagated down the tree by multiplying the probabilities characterizing a certain path of the tree. Finally, the resulting path probabilities are multiplied with the probability for a certain class at the leaf of each path and the probabilities for the same class are summed over the paths forming the class probabilities. The data instance is classified in the class with the highest class probability (majority vote). New decision trees are built by defining suitable splitting functions such as an information gain measure based on probabilistic cardinality (Qin et al. 2009). The result is a non-probabilistic tree. Similarly, Magnani and Montesi (2010) generate a table for each possible world alternative (data sources) that can occur and assign to each of these tables the probability of the corresponding alternative. This corresponds to the probability distributions in Qin et al. (2009) and Tsang et al. (2011). Then, from each of the tables (i.e. based on certain data) a decision tree is built, which would occur with the probability of the corresponding possible world alternative. In order to classify a data instance, for each class and each tree, the probability that the data instance belongs to this class is multiplied with the probability that the tree occurs. The final probability for a given class is the sum over the alternative trees, similar to the class probability in Qin et al. (2009) and Tsang et al. (2011).

The main aim of the above approaches is to consider data uncertainty in decision trees and not to provide efficient methods[1]. Thus, they need to be extended with regard to their efficiency to be applied to big data. To our knowledge, there is no such approach for decision trees in the literature (i.e. stemming *only* from the uncertain data mining literature). However, Leung and Hayduk (2013) extend the well-known UF-growth algorithm for mining uncertain frequent patterns (Leung et al. 2008) for application to big data. Their point is that the presence of uncertainty increases the search space and thus the runtime of the algorithm. Thus, Leung and Hayduk (2013) apply the MapReduce framework (Aggarwal 2013) which efficiently mines large volumes of distributed data to identify frequent patterns in the case of big data. Our approach is also based on such an idea. To better justify it, in the following we discuss the papers for uncertain big data mining, stemming from the big data mining literature. We focus on data streams, which are one of the most common applications.

Stream of research ii)

Attribute-level uncertainty in data streams can be divided into noise and concept drift uncertainty. Noise is seen as data, which "do not typically reflect the main trends but makes the identification of these trends more difficult" (Yang 2013, p.322) and can be interpreted as representing incorrect, inaccurate or outdated attribute values (Zhu and Wu 2004). Concept drift describes the change in the data distribution over time and appears when the distribution is not stationary, but "is drifting" (Hulten and Domingos 2001). For example, the relationship between annual income and employment status may change over time and as a result, the decision tree describing this relationship will change. The idea of a concept drift is related to currency in that it considers the development of data over time, but also differs from it, as it is concerned with the change in the underlying distribution over time, while currency is determined by the change of the corresponding attribute values in the real-world. A change in currency in the above example will be, if the employment status of the stored instance changes in the real world resulting in a different annual income class than the stored one.

Many of the papers that deal with classifying *certain* data streams with decision trees are based on the Hoeffding bound for tree growing upon the arrival of new data. The idea is that a new split takes place only when there is a statistically significant support for it in the data. The most famous such approach is the Very Fast Decision Tree (VFDT) (Domingos and Hulten 2000), which classifies data streams efficiently and incrementally. Since it has been shown that the presence of noise can reduce the classification accuracy and increase the tree size of decision trees for data streams (Yang and Fong 2011), existing approaches for classifying uncertain data streams are designed to be more robust against noise as compared to the ones for certain data. Examples for such approaches are the extensions of the VFDT

[1] Note, however that the authors still discuss some efficiency improvements (Tsang et al. 2011).

method (Yang and Fong 2011) and the FlexDT method. VFDT has been extended to account for noise by introducing a dynamic tie threshold based on the changing mean of the Hoeffding bound (Yang and Fong 2011). Hashemi and Yang (2009) apply fuzzy sets in the FlexDT method to mitigate the effect of noise on the classification accuracy. The approach is similar to the approaches by Qin et al. (2009) and Tsang et al. (2011), but is based on fuzzy set theory.

In order to consider the occurrence of a concept drift, approaches for classifying data streams use new instances to test the suitability of the existing decision trees. For example, Yang et al. (2012) and Hulten and Domingos (2001) update the sufficient statistics of the tree based on new instances and as a result decide if the tree structure should be changed. Hulten and Domingos (2001) grow gradually a new subtree which eventually replaces the old one. Hashemi and Yang (2009) backpropagate new instances up the tree and update the fuzzy parameters for potential changes in the data. If necessary, they grow a new subtree based on Hulten and Domingos (2001). Finally, Wang et al. (2003) apply ensemble classifiers to classify data streams and adjust their weights based on the values of new instances. As mentioned above, our focus is on the currency of the attribute values and not the stationarity of its distribution over time.

To sum up, the approaches for uncertain big data mining stemming from the big data mining literature are designed to be robust against noise and thus currency. However, they do not examine the source of the noise and do not model it explicitly[2]. This is the research gap we aim to close.

Stream of research iii)

An approach that has emerged from both the big data mining and the uncertain data mining literature is the one by Liang et al. (2010). They apply the idea for probabilistic cardinality (Qin et al. 2009) to model the uncertainty in data streams and update the sufficient statistics for new samples similar to Yang et al. (2012) and Hulten and Domingos (2001). However, they do not model the source of uncertainty.

Typical for all the presented approaches in the uncertain big data mining literature is that uncertainty is considered to exist, but without providing an interpretation for it. The approaches in the uncertain data mining literature assume that it is given in the form of a probability distribution without discussing the derivation of this distribution. The works dealing with the classification of uncertain data streams do not model uncertainty explicitly[2], but either modify their methods to increase their robustness against noise or update the decision trees in the case of a concept drift. Our approach contributes to the uncertain big data mining literature in *Stream of research i)*. We model currency as a probability distribution and classify stored instances by propagating their probabilities down the tree similar to Qin et al. (2009) and Tsang et al. (2011). We extend the literature by a) giving an interpretation of the probabilities in the form of currency, b) showing how these probabilities can be efficiently determined for an application in big data, and c) deriving them in a context-specific way with the use of supplemental data. We have chosen probability theory, because: 1) it allows for a mathematically-sound modelling of uncertainty, 2) it is a common well-founded approach for modelling currency in the literature and 3) the literature on uncertain data mining is mainly based on it. In the next section we present our approach.

A Method for Considering Currency in Decision Trees

In this section we present our two-phase approach, which considers the currency of stored instances when classifying them in *existing* decision trees. As a result, we do not concentrate on the process of building the tree, but only on the classification of stored instances in existing decision trees. Therefore, our approach is independent of the particular decision tree classifier (e.g. CHAID, C 4.5). The considered independent attributes can be numerical or categorical, while categorical dependent attributes are assumed, which is a standard assumption in the uncertain data mining literature[3] (e.g. Liang et al. 2010; Qin et al. 2009).

---

[2] An exception here is the paper by Hashemi and Yang (2009), who apply fuzzy set theory, but we focus on probability theory.

[3] Otherwise, a leaf will not be characterized by a given class of the dependent attribute, but by a probability distribution and thus final classification will not be possible.

Notation

In order to describe the approach we first introduce some necessary notation. As mentioned above, decision trees aim at classifying instances according to the values of a set of independent attributes in one of the classes of the dependent attribute. A path in the tree begins at the root and ends in one of its leaves. It is thus represented by a sequence of the splitting conditions of the independent attributes and the corresponding class of the dependent attribute. Let $A_i^I \in D_i^I, i \in \{1, \dots, n\}$ be the set of independent attributes with their corresponding domains D_i^I. Let, in addition, $d_i^{j(k)}, j(k) \in \{1(1), \dots, m(k)\}, k \in \{1, \dots, t\}$ represent the disjoint splitting independent attribute conditions for the attribute A_i^I at depth k of the tree (i.e. $\bigcup_{j=1}^{m} d_i^{j(k)} = D_i^I$) where t is the maximal depth of the tree. We call $d_i^{j(k)}, j(k) \in \{1(1), \dots, m(k)\}, k \in \{1, \dots, t\}$ *splitting subsets*. Let $A^D \in D^D$ analogously be the dependent attribute with its corresponding domain D^D and disjoint classes $c_l, l \in \{1, \dots, p\}$, $\bigcup_{l=1}^{p} c_l = D^D$. A path leading to a leaf node with the class c_l is then given by a sequence $path_{lu} = \left\{d_{i1}^{j(1)}, d_{i2}^{j(2)}, \dots, d_{it}^{j(t)}, c_l\right\}, \{i1, \dots it\} \subseteq \{1, .. n\}$, where u represents the different paths leading to the same class. A stored data instance is given by $G = \{s_1, \dots, s_n\}$, where $s_i \in D_i^I, \forall i$ represent the stored values of the corresponding independent attributes.

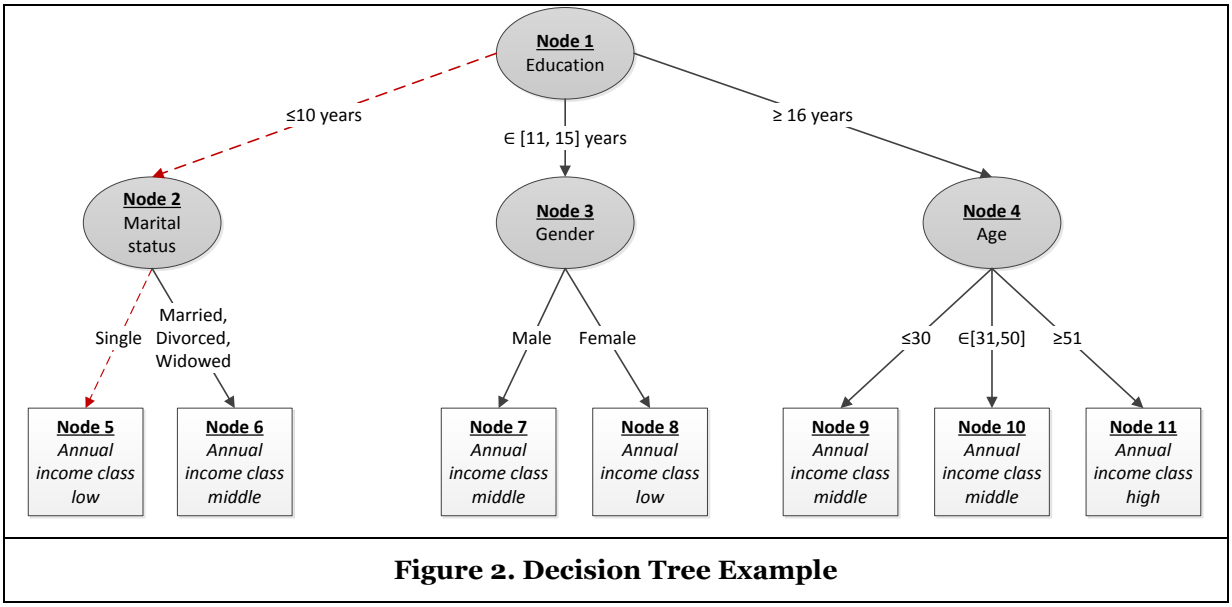

Figure 2. Decision Tree Example

To illustrate the idea, consider the tree in Figure 2, where the instances need to be classified according to the income of the person. The independent attributes are $Education\ (in\ years)$, $Marital\ status$, $Gender$, and $Age\ (in\ years)$, the dependent attribute is $Income$ with $c_1 = low, c_2 = middle, c_3 = high$. An example for a path is $path_{11} = \{\leq 10\ years, single, low\}$ (marked in Figure 2 with a dashed line) and an example for a stored instance that would follow this path (i.e. it would be classified in the class $c_1 = low$) is $G = \{10\ years, single, male, 18\ year\ old\}$.

Since the instance which must be classified, was stored some time ago, it may be the case that some or all of its values are outdated at the time of classification. Not considering the currency of these values may result in wrong classification and thus wrong decisions. For example, for the instance $G = \{10\ years, single, male, 18\ year\ old\}$, the individual may have got married or completed his university education (at least 16 years of education) in the meantime and thus be in the class of medium or high earners. If this is not considered, wrong decisions and thus economic losses may occur.

First Phase

In order to consider currency in decision trees, we first need to measure it and this is the *first* phase of our approach. For this aim, we follow the interpretation from above, where currency is seen as the probability

that the stored attribute value is up-to-date. Thus, as discussed in the second section, currency can be determined independently of the given decision tree based on the percentage of instances in the population which had the stored attribute value (e.g. single) at the time of storage and still have it at the time of classification (e.g. are still single).

This way of determining currency derives the probability for every possible value in the domain of the independent attribute separately. However, it is rather impractical, especially in the context of big data, since it requires reliable and detailed historical data and is computationally very intensive. Moreover, for decision trees, this high precision of the estimations is very rarely needed. The reason is that the splitting subsets in decision trees are often sets or intervals rather than single values. This is especially true for numerical attributes such as *Education* or Age, but may also occur with categorical attributes such as *Marital status* or *Industry*. For example, in Figure 2 the splitting subsets for the attribute *Marital status* are given by the sets $d_2^{1(2)} = \{single\}$ and $d_2^{2(2)} = \{married, divorced, widowed\}$, and for *Education* by the intervals $d_1^{1(1)} = [0, 10]$, $d_1^{2(1)} = [11, 15]$, $d_1^{3(1)} = [16, \infty)$. As a result, for a correct classification, it is not anymore crucial that the stored attribute value is up-to-date. Rather, it is enough that even if the value changed in reality, it still belongs to the same splitting subset. For example, in Figure 2, if the stored attribute value is *married* and the person is *divorced* at the time of classification, the correct classification of the stored instance will not be affected.

Thus, we propose an approach for considering currency in decision trees which is based on the splitting subsets of the particular tree. To describe the idea, consider for some $i \in \{1, .., n\}$ the attribute value s_i from the stored instance $G = \{s_1, \dots, s_n\}$ with the splitting subset(s)[4] $d_i^{j*(k)}$ such that $s_i \in d_i^{j*(k)}$ i.e. the attribute A_i^I is at depth k of the tree. We call $d_i^{j*(k)}$ *storage splitting subset(s)*. Then the currency of s_i is the probability that the instances which attribute values were in $d_i^{j*(k)}$ upon storage are still in it upon classification. We denote this probability by $p_i^{j*(k)}$. If the stored attribute value is up-to-date, then $p_i^{j*(k)} = 1$. This probability can be derived from historical data based on the Bayes' theorem. For example, in Figure 2 and $s_1 = 9\ years$, $p_1^{1(1)}$ will be the probability that during the time span between storing and classifying the instance, a person with less or equal to **ten** years of education at the time of storage did not study **more** than ten years until the time of classification. This implies that a stored instance with $s_1 = 9\ years$ will follow the path to $d_1^{1(1)} = [0, 10]$ with a probability $p_1^{1(1)}$. As a result, it will follow the path to $d_1^{2(1)} = [11, 15]$ or to $d_1^{3(1)} = [16, \infty)$ with a total probability of $1 - p_1^{1(1)}$. The exact probability with which the stored instance will follow each of these two paths is then the probability that the stored attribute value changed between storage and classification to a value, which belongs to one of the two splitting subsets. For example, for $d_1^{2(1)}$ this is the probability that a person who had less or equal to ten years of education at the time of storage, has received between eleven and fifteen years of education until the time of classification. It can be derived analogously to $p_1^{1(1)}$ from historical data.

Note that, in order to determine all these probabilities, we only need to know the corresponding storage splitting subset(s) $d_i^{j*(k)}$ and the structure of the tree. Thus, they can be derived independently of the stored instances (for all splitting subsets $d_i^{j*(k)}$) resulting in higher efficiency. Let, for each possible storage splitting subset $d_i^{j*(k)}$ (i.e. each splitting subset of the tree), $p_i^{j(k)}$ represent the so derived probabilities for each splitting subset $d_i^{j(k)}$ (including $d_i^{j*(k)}$) at the time of classification. We call $p_i^{j(k)}$ the *node* probability of $d_i^{j(k)}$ for the storage splitting subset $d_i^{j*(k)}$. Then, for a given stored instance $G = \{s_1, \dots, s_n\}$ and for each of its stored attribute values $s_i, i \in \{1, \dots, n\}$, only the corresponding storage splitting subset(s) $d_i^{j*(k)}$ with $s_i \in d_i^{j*(k)}$ need(s) to be identified to derive the node probabilities for $G = \{s_1, \dots, s_n\}$. Figure 3 provides the node probabilities for the tree in Figure 2 and the stored instance $G = \{10\ years, single, male, 18\ year\ old\}$.

[4] Note that an independent attribute may happen to be a splitting independent attribute more than once in a tree.

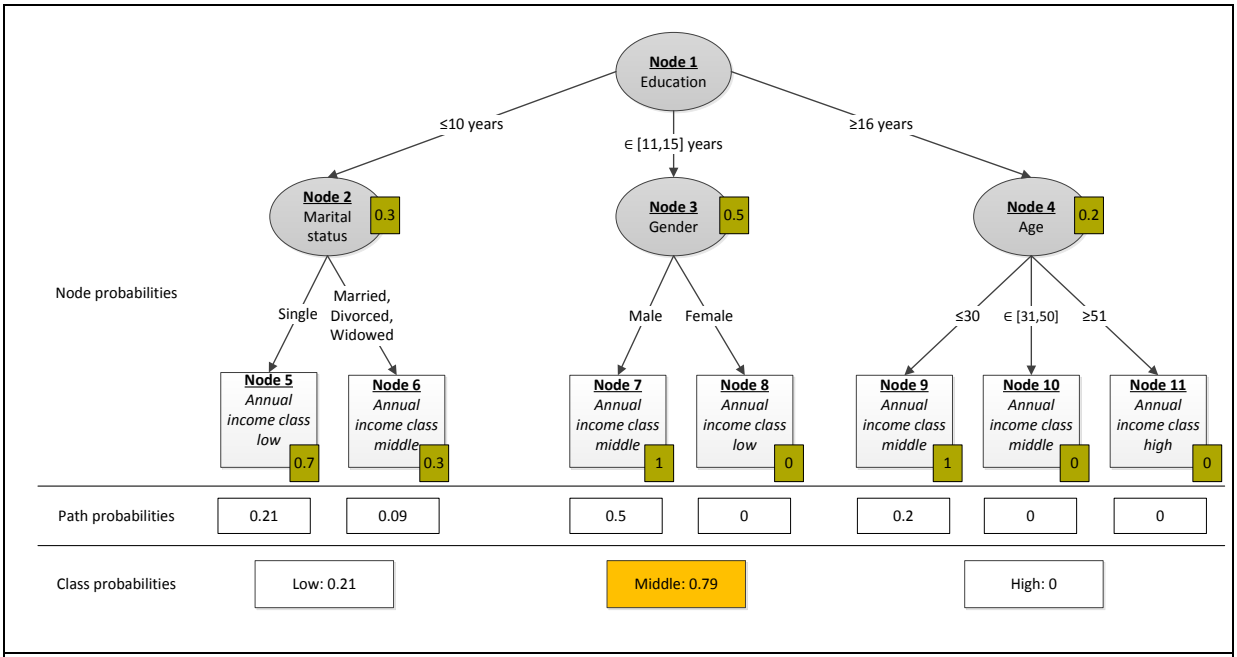

Figure 3. An Example for Considering Currency in a Decision Tree

This completes the description of the derivation of the node probabilities and thus the *first* phase of the algorithm. By deriving the node probabilities according to the splitting subsets of the decision tree, we model them more efficiently than the existing approaches for measuring currency. In addition, such a derivation considers the structure of the decision tree and thus the context of application. Finally, as opposed to the approaches in the uncertain big data mining literature, these probabilities have an interpretation based on the currency of the stored attribute values.

Second Phase

The *second* phase consists of classifying the stored data instance, which follows a certain path to reach a leaf with a class for the independent attribute. Based on the results from the *first* phase, for a given stored instance, we can assign to each path $path_{lu} = \left\{d_{i1}^{j(1)}, d_{i2}^{j(2)}, \dots, d_{it}^{j(t)}, c_l\right\}, \{i1, \dots it\} \subseteq \{1, .. n\}$ a sequence of the corresponding node probabilities $\left\{p_{i1}^{j(1)}, p_{i2}^{j(2)}, \dots, p_{it}^{j(t)}\right\}, \{i1, \dots it\} \subseteq \{1, .. n\}$. For example, in Figure 3, the node probabilities for the path $path_{11} = \{\leq 10\ years, single, low\}$ are given by {0.3,0.7}. This implies that the first splitting subset in the path will be followed with a probability of 0.3 and the second one will be followed with a probability of 0.7. In the next step, we need to determine the total probability that the stored instance follows a particular path, which we call *path* probability. We derive it, based on the uncertain data mining literature, by multiplying the node probabilities of the splitting subsets of the path. In the example above, the path probability that the instance $G = \{10\ years, single, male, 18\ year\ old\}$ follows $path_{11}$ is 0.21. Figure 3 provides the path probabilities for this instance and all the possible paths in the tree.

In order to classify the stored instance in a particular class, we consider the path probabilities for each of the classes. In Figure 3 the stored instance will be classified in the income class *low* either if it follows $path_{11}$ (i.e. a probability of 0.21) or if it follows $path_{12} = \{\in [11,15]\ years, female, low\}$ (i.e. a probability of 0). To determine the probability with which a given instance belongs to a class, we sum the probabilities of all the paths leading to this class (e.g. in Figure 3 for the class "low" 0.21+0) and call this *class* probability. Note that summation is possible because the splitting subsets are disjoint sets. Finally, the stored data instance is classified in the class with the highest class probability. In Figure 3 the instance $G = \{10\ years, single, male, 18\ year\ old\}$ is assigned to the class *middle* with a class probability of 0.79.

Classifying the instance in the class with the highest class probability corresponds to the majority vote from the literature concentrating on the combination of multiple classifiers (Fred 2001; Kittler et al. 1998; Kuncheva 2004; Seewald et al. 2001). The idea is that the different paths, an instance could follow, represent the different classifiers and the class of a path and the path probability stand for the result and the probability of each of the classifiers, respectively. According to the majority vote, the instance is assigned to the class with the highest class probability among the classifiers. In such cases ties are either resolved arbitrarily (Kuncheva 2004; Street and Kim 2001) or based on the prior probability of the particular class, which is the number of instances in the training set that belong to this class (Seewald et al. 2001). In case the prior probabilities are also equal, then the class is chosen arbitrarily. We resolve ties arbitrarily, as we believe that it is important that our method remains independent of the particular training set, not least due to efficiency reasons. This completes the description of the *second* phase of our approach, which is based on the ideas from the uncertain data mining literature. In Figure 4 the whole approach is summarized.

Phase I: Derivation of the node probabilities

1.For each splitting subset $d_i^{j*(k)}, i \in \{1, \dots, n\}, k \in \{1, \dots, t\}, j* \in \{1, \dots, m\}$ of the tree

i. Based on historical data, determine the node probability $p_i^{j*(k)}$ that an attribute value which belonged to $d_i^{j*(k)}$ at the time of storage still belongs to it at the time of classification

ii. For all $d_i^{j(k)}$ with $j \neq j*$ determine the node probabilities $p_i^{j(k)}$ that an attribute value which belonged to $d_i^{j*(k)}$ at the time of storage belongs to $d_i^{j(k)}$ at the time of classification

iii. Store the node probabilities $p_i^{j(k)}, j \in \{1, \dots, m\}$ from 1.a or 1.b with $d_i^{j*(k)}$

2. For each stored instance $G = \{s_1, \dots, s_n\}$

i. For each stored attribute value $s_i, i \in \{1, \dots, n\}$

a. Identify the storage splitting subset(s) $d_i^{j*(k)}$ s.t. $s_i \in d_i^{j*(k)}$

b. For each $d_i^{j*(k)}$ from a. assign to the splitting subsets $d_i^{j(k)}, j \in \{1, \dots, m\}$ the probabilities $p_i^{j(k)}, j \in \{1, \dots, m\}$ stored with $d_i^{j*(k)}$ in 1.iii

Phase II: Classification of the stored instance

1. For each path $path_{lu} = \left\{d_{i1}^{j(1)}, d_{i2}^{j(2)}, \dots, d_{it}^{j(t)}, c_l\right\}, \{i1, \dots it\} \subseteq \{1, .. n\}, l \in \{1, \dots, p\}$ in the tree with the corresponding probabilities $\left\{p_{i1}^{j(1)}, p_{i2}^{j(2)}, \dots, p_{it}^{j(t)}\right\}, \{i1, \dots it\} \subseteq \{1, .. n\}$ derived in Step I.1.d determine the path probability $prob_{lu} = \prod_{k=1}^{t} p_{ik}^{j(k)}$

2. For each class of the dependent attribute $c_l, l \in \{1, \dots, p\}$, sum the probabilities for the paths of the type $path_{lu}$ to determine the class probability

3. Classify the data instance in the class $c_l, l \in \{1, \dots, p\}$ with the highest class probability (in case of a tie choose the class randomly)

Figure 4. A Two-Phase Method for Incorporating Currency in Decision Trees

In order to demonstrate the efficiency of our approach, we compare its complexity to the complexity of the approach presented in the second section. It is enough to compare only the complexities of the calculation of the node probabilities (i.e. Step 1 of Phase I) as we assume that in the other steps the two approaches are identical[5]. We calculate the complexity for our approach based on Figure 4. The complexity for Steps

[5] The approach in the second section was not developed for the application to decision trees. This is an assumption we make to be able to compare the two approaches.

1.i-1.iii depends on the available historical data. If the historical dataset consists of N instances, for each $d_i^{j*(k)}$ the algorithm will take $O(N)$ for each splitting subset. Thus, Step 1 requires $O(NnMAXk^2)$, where $MAXk$ represents the maximal number of splitting subsets for a given attribute and n stands for the number of attributes. In the approach from the second section, for each attribute the probabilities for all possible stored attribute values need to be determined. This results in a complexity in $O(NnMAXvMAXk)$, where $MAXv$ represents the maximal number of different stored independent attribute values for a given attribute. Already here the complexity of our approach is lower or equal (in the worst case) to in the second section, because $MAXk \leq MAXv$, especially for numeric attributes. This proves the efficiency of our approach. In the next subsection we show how the node probabilities can be more precisely and context-specifically modelled based on supplemental data.

Supplemental Data

In the approach presented above, the node probabilities are determined for each splitting subset without considering the values of the other attributes. This implies, for example in Figure 3, for the splitting subset $single$ and for the instance $G = \{10\ years, single, male, 18\ year\ old\}$ that the probability for a person staying single in the time span between storage and classification is independent of supplemental data such as the years of education, gender, and age of this person. However, as mentioned in the second section, the literature has shown that considering supplemental data can lead to a more precise, context-specific estimation. Thus, it is reasonable to modify the derivation of the node probabilities for a given instance so that not only $d_i^{j*(k)}$ is considered, but also the other stored values. Based on the discussion above, for a given stored instance $G = \{s_1, \dots, s_n\}$ we would derive $p_i^{j(k)}$ as the probability that s_i belongs to $d_i^{j(k)}$ at the time of classification provided that the values of the other attributes were $\{s_1, \dots s_{i-1}, s_{i+1}, \dots s_n\}$ at the time of storage.

This approach considers high amount of additional information in the derivation of the probabilities, but it is again very impractical in the context of big data since very detailed historical data is required to determine all the probabilities. For example, for the instance $G = \{10\ years, single, male, 18\ year\ old\}$ and the attribute value *single*, we need historical data on the males, who were 18-year-old, single, and had ten years of education at the time of storage, and who got married until the time of classification. To avoid this problem and make the approach better applicable in the context of big data, we consider as supplemental data only the stored attribute values which *precede* the stored value in the path. Moreover, we do not calculate the probabilities based on single values such as *18-year-old*, but rather use the corresponding splitting subset of the tree (e.g. ≤ 30-year-old). Thus, we still determine the node probabilities $p_i^{j(k)}$ based on supplemental data, but with fewer and less granular conditions. For example, for $G = \{10\ years, single, male, 18\ year\ old\}$ and the stored attribute value $single$, the node probability $p_2^{1(2)}$ for the path $path_{11} = \{\leq 10\ years, single, low\}$ will be derived by considering all singles who had less or equal to 10 years of education at the time of storage without posing any additional restrictions on their gender or age.

This approach is very reasonable in the context of decision trees, since the order of the independent attributes represents their importance with respect to the classification of the dependent attribute. The root is the most important attribute and with increasing depth the importance decreases. Moreover, a path in the tree is also a sequence of conditions, where each consecutive condition is considered, only if the preceding one is fulfilled and regardless of the stored attribute values, which are not part of the path. For example, for $G = \{10\ years, single, male, 18\ year\ old\}$ and the path $path_{11} = \{\leq 10\ years, single, low\}$, the fact that the person is an 18-year-old male is not relevant for the classification and would thus not influence the result when incorporating currency.

In some cases, due to missing historical data, it may happen that a given node probability is zero. This is especially the case when supplemental data is used, as detailed historical data is needed, which is not always available in reality. The problem is then that these zero probabilities are propagated down the tree and no instance is classified in the corresponding class. To solve this "zero-frequency-problem" we apply a smoothing technique by transforming the zero probabilities with a Laplace transformation. This approach is often used in data mining for such situations (Witten and Frank 2005).

To sum up, in this section we presented our novel approach for considering currency in decision trees. The presented two-phase algorithm is interpretable, universal, context-specific, and efficient. It is interpretable, because currency is interpreted as probability. In addition, it is universal, because it is independent of the particular decision tree classifier. Moreover, it is context-specific, because both determining the node probabilities and considering supplemental data strongly depend on the stored instance and on the structure of the tree. Finally and most importantly in the context of big data, it is efficient because 1) it only modifies the process of classifying stored instances and not the decision tree algorithm itself, 2) it determines the node probabilities based on the set of splitting subsets and not on a single value and 3) it incorporates supplemental data based on the tree structure and not on all the stored values. In the next section our approach is evaluated with three real-world datasets.

Evaluation

In this section we evaluate our approach by applying it to three different datasets for the two applications presented in the introduction. The first dataset consists of the publicly available panel data from the SOEP (2012)[6] where for each year the personal characteristics (age, gender, marital status, education, industry, employment status, annual income, etc.) of the same individuals are stored. This dataset represents the Customer Relationship Management scenario, in which a financial service provider conducts a campaign to attract new customers based on their annual income. It derives the relationship between the annual income of the customers and their personal characteristics, based on an up-to-date database of existing customers, in the form of a decision tree. Since the CRM campaign took place some time ago, some of the stored personal characteristics of the people who took part in it may be outdated resulting in a wrong income class. Since the company targets the customers according to their income, this may lead to the wrong product being offered and thus economic losses.

The second dataset is from the UCI Machine Learning Repository (Bache and Lichman 2014), which is a source commonly used in the big data mining literature. The dataset is called "Localization Data for Posture Reconstruction" and contains the results from sensors attached to four different parts of the body of individuals who were performing different activities in a number of sessions. For each sensor, its x, y and z coordinates are given. This dataset stands for the scenario mentioned in the introduction for handicapped individuals, where the aim is to determine the current activity of the patient based on the measurements coming from the sensors. The data is used to build the relationship between the sensors and their values on the one side and the current activity on the other side in the form of the decision tree. Based on this tree, instances stored some time ago (this time span can also be as small as a couple of minutes) are classified to determine the current activity of the individuals. If the currency of the sensor values is not taken into account, wrong activities may result causing the wrong responses, which may have devastating consequences.

Finally, the third dataset was generated by a mobile application, developed for the purpose of this paper and can be provided upon request. This dataset also represents the scenario for handicapped individuals, but as opposed to the second one, it contains the values of all the sensors of a mobile device, thus extending the scenario by considering the development in mobile technologies. Nowadays, many companies develop mobile applications that are applied for activity recognition[7]. In the case of elderly and handicapped care, such applications can increase the probability of an appropriate response and also result in higher acceptance as it is enough to have a mobile device and do not need many different sensors attached to the body. Thus, in this scenario, the relationship between the current activity of a person and the values from the sensors delivered from the mobile application is examined in the form of a decision tree. As for the second dataset, the stored sensor values may be outdated (e.g. due to wrong transmission, sensor error, etc.) resulting in the wrong activity and thus a wrong response. In the next subsection we provide the evaluation procedure we follow to demonstrate the advantages of our approach and describe the three datasets in detail.

---

[6] The data used in this publication was made available to us by the German Socio-Economic Panel Study (SOEP) at the German Institute for Economic Research (DIW), Berlin.

[7] E.g. https://actitracker.com/

Data and Evaluation Procedure

In order to evaluate our approach we define for each of the three datasets a point in time in which the data was stored (t_0) and another, later point in time in which the classification takes place (t_1). Then each of the three datasets is randomly divided into three *mutually exclusive* subsets, which are then used to derive the *training and test* set (approx. 50% of the instances), the *currency* set (approx. 25% of the instances) and the *validation* set (approx. 25% of the instances).

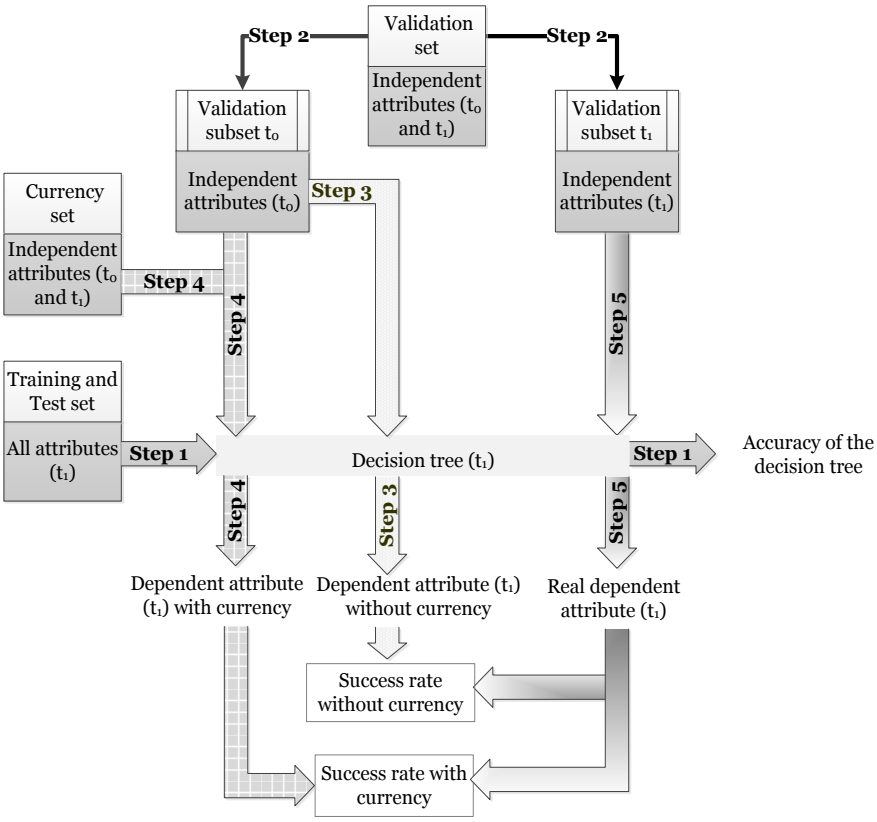

Figure 5. Evaluation Procedure

The evaluation procedure is presented in Figure 5. In **Step 1** the decision tree is built and its accuracy is tested based on the *training and test* set. This set contains the values of the independent attributes and of the dependent attribute of the instances for t_1 and represents the real world values at the time of classification. In **Step 2** the *validation* set, which contains the values for the independent attributes of the individuals for both t_0 and t_1, is divided into two *validation* subsets according to the point of acquisition. In **Step 3** the decision tree is applied to the *validation* subset for t_0 and the result is the class of the dependent attribute based on the values of the independent attributes for t_0 and the decision tree for t_1. In **Step 4** the *currency* set is used to determine the node probabilities for the independent attribute values in t_1, given the stored independent attribute values in t_0. This set contains the values for the independent attributes for both t_0 and t_1. The node probabilities are used in this step to classify the stored instances from the *validation* subset for t_0 in the decision tree for t_1. The result is the class of the dependent attribute based on the decision tree for t_1 and the values of the independent attributes in t_0 after considering the currency of the stored attributes. To determine the success rate of our approach, in **Step 5** the decision tree is applied to the *validation* subset for t_1 and the resulting classes (corresponding to the real-world classes of the independent attribute in t_1) are compared with the classes from **Step 3** and **Step**

4. We additionally restrict the validation and the currency sets to instances, for which the attribute value corresponding to the root of the tree changed to demonstrate our approach. Note that in the case when currency is considered, we additionally examine the results for incorporating supplemental data. In the following we describe the three datasets in detail.

The first dataset is the panel data SOEP (2012) from which we consider t_0=2001 and t_1=2011, as 2011 is the last year the dataset. We filter only the individuals that are contained in both datasets resulting in 9522 instances. The dependent variable for this dataset is the annual income of the individuals and the independent variables are: *In Training,* which shows if the individual is currently in training; *Employment*, which gives the employment status of the individual; *Education*, which stands for the number of years of education; *cc*; and *Gender*. Since the dependent attribute for the decision tree is categorical, we categorize the annual income in classes with the software tool IBM Corp. Released (2011) so that each class contains an approximately equal part of the individuals in the panel data. Note that if this is not done, as mentioned above, the result will not be a distribution over the classes, but rather a distribution over the distributions of the income in the different leaves. Thus determining the final class will not be possible, which is crucial in real-world applications.

The second dataset is the "Localization Data for Posture Reconstruction" consisting of 164860 instances, where each instance consists of the person who was measured (5 people), the session (5 sessions per person), the type of sensor (4 sensors), the time of measurement, a unique timestamp, the sensor's three coordinates, and the current activity of the individual (4 activities). The independent variables are *Person*, *Sensor*, *xCoordinate*, *yCoordinate* and *zCoordinate*. The dependent variable is *Activity*. t_0 was chosen to represent the first two sessions, while t_1 was chosen to represent the second two sessions.

The third dataset was gathered with a mobile device with an Android 4.4.2 operating system and a Quad-Core Qualcomm Snapdragon 600 processor. The dataset consists of 1839062 instances in two sessions (for t_0 and t_1). The dependent attribute is again *Activity* (4 classes based on Wu et al. 2011) and the independent attributes are the values of the sensors of the device (18 sensors). Most sensors had three values and thus for a better comparison to the second dataset, we omitted the sensors with less or more than three values. In the next subsection we present the decision tree.

Decision Trees

We applied to the training and test set of each of the three datasets above both the CHAID and the C4.5 algorithm (Step 1 in Figure 5). The CHAID is based on the chi-squared test of independence and results in non-binary trees (Kass 1980). The C4.5 is based on an information gain measure and also results in non-binary trees (Quinlan 1993). We used IBM Corp. Released (2011) with default options and an adjustable depth (depending on the dataset) for the CHAID algorithm, and WEKA (Hall et al. 2009) with reduced-error pruning and a minimum number of instances (depending on the dataset) for C4.5. For CHAID all the p-values were significant at the 5%-level. We tested the models with both a randomly chosen test set (33 %) and a cross validation (10-fold). The accuracy of the decision trees is presented in Table 1. We can see that depending on the dataset, the method, and the validation, the results may differ, but generally the accuracies are reliable enough for the models to be considered for further analysis.

Table 1. Accuracy of the Decision Trees

Dataset/Method	CHAID a	CHAID b	C4.5 a	C4.5 b
1	71%	74%	73%	74%
2	65%	67%	71%	73%
3	75%	77%	77%	78%
Legend: *a = Test set, b = Cross validation*				

Performance of the New Approach

As presented in Figure 5, to evaluate our approach we conduct Steps 3-5 based on the models from Table 1. The results are presented in Table 2. As we can see, considering currency leads always to higher success rate than not doing so and also considering supplemental data can additionally increase the success rate.

The success rate of supplemental data would be even higher, if the trees contained attributes that strongly changed according to the additional information, which is not the case in the models in Table 1.

Table 2. Comparison of the Success Rates of the Approaches

Dataset/Method	CHAID i	CHAID ii	CHAID iii	C4.5 i	C4.5 ii	C4.5 iii
1 a	6%	33%	33%	5%	31%	31%
1 b	7%	29%	29%	7%	29%	29%
2 a	33%	59%	60%	30%	54%	60%
2 b	32%	62%	62%	30%	44%	48%
3 a	45%	63%	63%	51%	73%	73%
3 b	46%	64%	65%	37%	53%	53%

Legend: *a = Test set, b = Cross validation,*
i =without currency, ii =with currency, iii =with currency and supplemental data

Finally, to demonstrate the efficiency of our approach, we measured the runtimes of computing the node probabilities with our approach and with the approach presented in the second section. Note that, as mentioned above, to conduct this comparison certain assumptions need to be made and the approach of the second section needs to be additionally modified. We measured the runtimes on two different machines (M1 with Intel(R) Core™ i5-2520M CPU @ 2.50GHz, 4.00 GB RAM and M2 with Intel(R) Core™2 Duo CPU T6570 @ 2.10GHz, 4.00 GB RAM) to test the reliability of the results. As we can see and as shown in the third section, our approach is almost always faster than the one presented in the second section which proves its efficiency.

Table 3. Runtimes of the Approaches in milliseconds

Method/Dataset	1a(M1 /M2)	1b(M1 /M2)	2a(M1 /M2)	2b(M1 /M2)	3a(M1 /M2)	3b(M1 /M2)
CHAID I ii	47/ 78	47/ 94	281/ 280	297/ 577	5546/ 10358	6156/ 11840
CHAID II ii	47/ 78	47/ 78	218/ 250	203/ 296	5437/ 10093	6062/ 11606
C4.5 I ii	47/ 109	62/ 109	187/ 218	203/ 296	1484/ 2153	1797/ 3183
C4.5 II ii	47/ 78	63/ 109	172/ 203	172/ 296	1312/ 2121	1907/ 3135
CHAID I iii	62/ 94	78/ 171	875/ 1689	972/ 1873	100159/ 443448	102324/ 459578
CHAID II iii	62/ 94	62/ 109	844/ 1436	953/ 1592	99273/ 441532	101944/ 456993
C4.5 I iii	63/ 109	156/ 250	33797/ 68267	61817/ 159574	677299/ 2484836	748587/ 3192598
C4.5 II iii	46/ 78	156/ 128	33344/ 66785	61750/ 128992	602771/ 2479723	717455/ 3141878

Legend: *a = Test set, b = Cross validation, ii =with currency, iii =with currency and supplemental data, I = approach from the second section, II= our approach*

Availability of an Up-to-Date Training and Test Set

In our approach and thus in the evaluation above, we implicitly assume the existence of an up-to-date Training and Test set for building a decision tree that describes the current relationships. If this is not the case and there is a concept drift in the data, then the relationships presented in the tree may be outdated. Such a situation may occur when in the CRM-scenario the company does not possess the up-to-date information of its current customers, as a customer is not obliged to inform the insurer regarding all changes in his/her personal characteristics. It may also occur if there is no up-to-date available data about the sensor measurements and the corresponding activities in the handicapped-people-scenario. In such a situation, the approaches from the uncertain big data mining literature presented above (e.g. Tsang et al. 2011) can be applied to modify the decision tree. This will reduce the accuracy and increase the runtime of the approach with increasing negative effects for more outdated data. However, the results will still be better than not considering currency as the decision tree in both cases is the same and only the classification of stored instances is determined by the two approaches.

Conclusion

In this paper we present an approach for considering currency in the decision tree classification method in the context of big data. Our idea is based both on the literature for modelling currency and on the one for uncertain big data mining in decision trees. Based on the first stream of research, we demonstrate how currency can be derived for the consideration in decision trees in an efficient and context-specific way. In particular, we show how the probability, which commonly represents currency in the literature, can be derived with little historical data and depending on the structure of the decision tree. This makes our approach suitable for the context of big data. In addition, we demonstrate how this probability can be refined through the use of supplemental data, but again in an efficient and decision-tree-specific way. Based on these considerations, we determine the node probabilities for a given stored instance and a decision tree and apply the ideas from the uncertain data mining literature to classify these stored instances. We extend this stream of research by showing how the uncertainty, which is assumed to be given by the authors there, can be derived in an interpretable, efficient and context-specific way, where the interpretation is given by the currency of stored data. The applicability and the contribution of our approach are demonstrated based on three different datasets, two of which stemming from the context of big data. The results demonstrate that our approach leads to a substantial improvement in the classification accuracy as opposed to not considering currency and also that it is more efficient than the approach presented in the second section.

Our method also has some limitations. First of all, it is developed for decision trees. However, the problem of input data of low quality is just as relevant for other data mining methods such as clustering, for example. The idea to develop an approach based on the data quality and the uncertain data mining literature can be applied there in a similar fashion. In addition, in some cases (especially with supplemental data) historical data may not be provided. An alternative idea would be to consider expert estimations instead. These can be modelled with fuzzy set theory based on fuzzy decision trees (Yuan and Shaw 1995) and fuzzy metrics for currency (Heinrich and Hristova 2014). Moreover, in the future our approach can be additionally optimized by using new methods such as the MapReduce framework (Aggarwal 2013) for even more efficient application in the context of big data. Finally, applying the approach to other types of big data such as data streams or unstructured data, for example, will be an additional challenge for future research.

References

Aggarwal, C. C., and Yu, P. S. 2009. "A survey of uncertain data algorithms and applications," *IEEE Transactions on Knowledge and Data Engineering* (21:5), pp. 609-623.

Aggarwal, C. C., Ashish, N., and Sheth, A. 2013. "The internet of things: A survey from the data-centric perspective," in *Managing and mining sensor data*: Springer, pp. 383–428.

Aggarwal, C. C. 2013. *Managing and mining sensor data*, New York: Springer

Alpar, P., and Winkelsträter, S. 2014. "Assessment of data quality in accounting data with association rules," *Expert Systems with Applications* (41:5), pp. 2259–2268.

Azar, A. T., and El-Metwally, S. M. 2013. "Decision tree classifiers for automated medical diagnosis," *Neural Computing and Applications* (23:7-8), pp. 2387-2403.

Ballou, D., Wang, R., Pazer, H., Tayi, G. K. 1998. "Modeling information manufacturing systems to determine information product quality," *Management Science* (44:4), pp. 462-484.

Bache, K. and Lichman, M. 2014. "*UCI Machine Learning Repository*," http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science.

Berger, J. O. 2010. *Statistical Decision Theory and Bayesian Analysis*, New York, NY: Springer.

Blake, R., and Mangiameli, P. 2011. "The Effects and Interactions of Data Quality and Problem Complexity on Classification*," Journal of Data and Information Quality* (2:2), pp. 1–28.

Chau M., Cheng R., Kao B., and Ng, J. 2006. „Uncertain Data Mining: An Example in Clustering Location Data," in *Advances in Knowledge Discovery and Data Mining*, Ng W., Kitsuregawa M., Li J. and Chang K. (eds), vol 3918. Springer Berlin Heidelberg, pp 199-204.

Dasu, T., and Johnson, T. 2003. *Exploratory data mining and data cleaning*, Wiley Online Library.

Domingos, P., and Hulten G. 2000. "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '00)*, ACM, New York, NY, USA, pp. 71-80.

Economist Intelligence Unit 2011. *Levelling the Playing Field: How Companies Use Data for Competitive Advantage.*

Even, A., and Shankaranarayanan, G. 2007. "Utility-driven assessment of data quality," *ACM SIGMIS Database* (38:2), pp. 75-93.

Experian QAS 2013. The Data Advantage: How accuracy creates opportunity.

Fan, W. 2013. "Querying Big Social Data," in *Big Data*, G. Gottlob, G. Grasso, D. Olteanu, and C. Schallhart (eds.): Springer Berlin Heidelberg, pp. 14-28.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. 1996. "From data mining to knowledge discovery in databases," *AI magazine* (17:3), pp. 37.

Federal Bureau of Statistics 2013. *Annual Abstract of Statistics (in German).*

Fisher, C. W., Lauria, E. J., and Matheus, C. C. 2009. "An accuracy metric: Percentages, randomness and probabilities," *Journal of Data and Information Quality (JDIQ)* (1:3), pp. 16.

Forbes, I. 2010. *Managing Information in the Enterprise: Perspectives for Business Leaders.*

Fred, A. 2001. "Finding consistent clusters in data partitions," in *Multiple classifier systems*: Springer, pp. 309–318.

Giudici, P., and Figini, S. 2009. *Applied Data Mining for Business and Industry*. John Wiley and Sons.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. 2009. „The WEKA Data Mining Software: An Update," *SIGKDD Explorations* (11:1), pp. 10-18 .

Hashemi, S., and Yang, Y. 2009. "Flexible decision tree for data stream classification in the presence of concept change, noise and missing values," *Data Mining and Knowledge Discovery* (19:1), pp. 95-131.

Hawarah L., Simonet A., Simonet M. 2009. "Dealing with Missing Values in a Probabilistic Decision Tree during Classification," in *Mining Complex Data*, Zighed D., Tsumoto S., Ras Z., Hacid H. (eds), vol. 165. Springer Berlin Heidelberg, pp. 55-74.

Heinrich, B., and Klier, M. 2009. „A Novel Data Quality Metric for Timeliness considering Supplemental Data," in *Proceedings of the 17th European Conference on Information Systems*, University of Verona, pp. 2701-2713.

Heinrich, B., and Klier, M. 2011. „Assessing data currency-a probabilistic approach," *Journal of Information Science* (37:1), pp. 86-100.

Heinrich, B. and Hristova, D. 2014. „A Fuzzy Metric for Currency in the Context of Big Data," in *Proceedings of the 22nd European Conference on Information Systems*, Tel Aviv, Israel.

Hems, A., Soofi, A., and Perez, E. 2013. *How innovative oil and gas companies are using big data to outmaneuver the competition*. A Microsoft White Paper.

Hulten, G., Spencer, L. and Domingos, P. 2001. "Mining time-changing data streams," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 97.

IBM Corp. Released 2011. *IBM SPSS Statistics for Windows. Version 20.0*. Armonk, NY: IBM Corp.

IBM Institute for Business Value 2012. *Analytics: The real-world use of big data, How innovative enterprises extract value from uncertain data.*

Kass, G. V. 1980. "An exploratory technique for investigating large quantities of categorical data," *Applied statistics*, pp. 119-127.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. 1998. "On combining classifiers*," IEEE Transactions on Pattern Analysis and Machine Intelligence*, (20:3), pp. 226–239.

Kuncheva, L. I. 2004. *Combining pattern classifiers: methods and algorithms*: John Wiley & Sons.

Koh, H. C., Tan, W. C., and Goh, C. P. 2006. "A two-step method to construct credit scoring models with data mining techniques," *International Journal of Business and Information* (1:1), pp. 96-118.

Leung, C. K.-S., Mateo, M. A. F., and Brajczuk, D. A. 2008. "A tree-based approach for frequent pattern mining from uncertain data," in *Advances in Knowledge Discovery and Data Mining*: Springer, pp. 653–661.

Leung, C.-S., and Hayduk, Y. 2013. "Mining Frequent Patterns from Uncertain Data with MapReduce for Big Data Analytics," in *Database Systems for Advanced Applications*, W. Meng, L. Feng, S. Bressan, W. Winiwarter, and W. Song (eds.): Springer Berlin Heidelberg, pp. 440-455.

Li, F., Nastic, S., and Dustdar, S. 2012. "*Data Quality Observation in Pervasive Environments,*" in *Proceedings of the IEEE 15th International Conference on Computational Science and Engineering (CSE)*, 5-7 Dec. 2012, Nicosia, pp. 602-609.

Liang, C., Zhang, Y., and Song, Q. 2010. "Decision Tree for Dynamic and Uncertain Data Streams," *Journal of Machine Learning Research-Proceedings Track* (13), pp. 209-22.

Magnani, M., and Montesi, D. 2010. "Uncertainty in Decision Tree Classifiers," in *Scalable Uncertainty Management*, Lecture Notes in Computer Science, Vol. 6379, pp. 250-263.

Mezzanzanica, M. Boselli, R., Cesarini, M., and Mercorio, F. 2012. "Towards the use of Model Checking for performing Data Consistency Evaluation and Cleansing," in *Proceedings of the 17th International Conference on Information Quality (ICIQ 2012).*

Miller, I., Miller, M., and Freund, J. E. 2004. *John E. Freund's mathematical statistics with applications*, Upper Saddle River, NJ: Prentice Hall.

Ngai, E. W., Xiu, L., and Chau, D. C. 2009. "Application of data mining techniques in customer relationship management: A literature review and classification," *Expert Systems with Applications* (36:2), pp. 2592-2602.

Orr, K. 1998. "Data quality and systems theory," *Communications of the ACM* (41:2), pp. 66-71.

Pathak, A. N., Sehgal, M., and Christopher, D. 2011. "A Study on Fraud Detection Based on Data Mining Using Decision Tree," *International Journal of Computer Science* (8:3), pp. 258-261.

Qin, B., Xia, Y., and Li, F. 2009. "DTU: a decision tree for uncertain data," in *Advances in Knowledge Discovery and Data Mining*, Theeramunkong, T., Kijsirikul, B., Cercone, N. and Ho, T. (eds), Springer, pp. 4-15.

Quinlan, J. R. 1993. *C4. 5: programs for machine learning*: Morgan Kaufmann.

Redman, T. C. (1996). Data Quality for the Information Age. Boston, MA: Artech House.

Seewald, A. K., and Fürnkranz, J. 2001. "An evaluation of grading classifiers," in *Advances in Intelligent Data Analysis*: Springer, pp. 115–124.

SOEP 2012. *Socio-Economic Panel Study (SOEP), Data for the years 1984-2011, Version 28*. doi:10.5684/soep.v28.

Street, W. N., and Kim, Y. 2001. "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 377–382.

Tsang, S., Kao, B., Yip, K. Y., Ho, W.-S. and Lee, S. D. 2011. "Decision trees for uncertain data," *IEEE Transactions on Knowledge and Data Engineering* (23:1), pp. 64-78.

Vazirigiannis, M., Halkidi, M., and Gunopulos, D. 2003. *Uncertainty handling and quality assessment in data mining,* Springer.

Wang, H., Fan, W., Yu, P. S., and Han, J. 2003. "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03)*, ACM, New York, NY, USA, pp. 226-235.

Wang, R. Y., and Strong, D. M. 1996. "Beyond accuracy: What data quality means to data consumers," *Journal of Management Information Systems* (12:4), pp. 5-33.

Wechsler, A., and Even, A. 2012. "Using a Markov-Chain Model for Assessing Accuracy Degradation and Developing Data Maintenance Policies," in *Proceedings of the AMCIS 2012*, Paper 3.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann.

Wu, J., Jiang, C., Houston, D., Baker, D., and Delfino, R. 2011. "Automated time activity classification based on global positioning system (GPS) tracking data," *Environ Health* (10), p. 101.

Yang, H., and Fong, S. 2011. "Moderated VFDT in Stream Mining Using Adaptive Tie Threshold and Incremental Pruning," in *Data Warehousing and Knowledge Discovery*, A. Cuzzocrea, and U. Dayal (eds.): Springer Berlin Heidelberg, pp. 471-483.

Yang, H., Fong, S., Sun, G., and Wong, R. 2012. "A Very Fast Decision Tree Algorithm for Real-Time Data Mining of Imperfect Data Streams in a Distributed Wireless Sensor Network," *International Journal of Distributed Sensor Networks* (2012), pp. 1–16.

Yang, H. 2013. "Solving Problems of Imperfect Data Streams by Incremental Decision Trees," *Journal of Emerging Technologies in Web Intelligence* (5:3),pp 322-331.

Yuan, Y., and Shaw, M. J. 1995. "Induction of fuzzy decision trees," *Fuzzy Sets and Systems* (69:2), pp. 125-139.

Yue, D., Wu, X., Y, W., Li, Y., and Chu, C-H. 2007. "A review of data mining-based financial fraud detection research," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, WiCom 2007, pp. 5519-5522.

Zhu, X., and Wu, X. 2004. "Class Noise vs. Attribute Noise: A Quantitative Study," *Artificial Intelligence Review* (22:3), pp. 177-210.