Adam Mickiewicz University in Poznań
Faculty of Mathematics and Computer Science

Ensemble classification of incomplete data – a non-imputation approach with an application in ovarian tumour diagnosis support

Grupowa klasyfikacja danych niekompletnych – podejście nieimputacyjne z zastosowaniem we wspomaganiu diagnostyki guzów jajnika

Andrzej Wójtowicz

Dissertation for the degree of Doctor of Philosophy
in Mathematics in the field of Information Science

Praca doktorska na stopień doktora nauk matematycznych
w zakresie informatyki

Supervisor:
Prof. Dr hab. Maciej Wygralak

Auxiliary supervisor:
Dr Krzysztof Dyczkowski

Poznań, 2017

Acknowledgements

I would like to gratefully acknowledge the contribution of many people to the conception and completion of this thesis. I would like to thank Professor Maciej Wygralak and Dr Krzysztof Dyczkowski for their support in planning and carrying out the research. I am deeply grateful to Dr Anna Stachowiak and Dr Patryk Żywica for their help in harnessing and selecting medical data, as well as in the implementation of aggregation operators. I would like to thank Professor Dariusz Szpurek, Dr hab. Rafał Moszyński and Dr Sebastian Szubert for their advice and help concerning the medical problem. I am grateful to Maciej Prill for maintaining an excellent computing environment. Finally, I would like to thank my family for their generous support.

The research was partially supported by the Microsoft Research Award and the Innovation Incubator programme of the Polish Ministry of Higher Education.

Abstract

In this doctoral dissertation I focus on the problem of classification of incomplete data. The motivation for the research comes from medicine, where missing data phenomena are commonly encountered. The most popular method of dealing with data missingness is imputation; that is, inserting missing data on the basis of statistical relationships among features. In my research I choose a different strategy for dealing with this issue. Classifiers of a type previously developed can be transformed to a form which returns an interval of possible predictions. In the next step, with the use of aggregation operators and thresholding methods, one can make a final classification. I show how to make such transformations of classifiers and how to use aggregation strategies for interval data classification. These methods improve the quality of the process of classification of incomplete data in the problem of ovarian tumour diagnosis. Additional analysis carried out on external datasets from the University of California, Irvine (UCI) Machine Learning Repository shows that the aforementioned methods are complementary to imputation.

Streszczenie

W niniejszej pracy doktorskiej zająłem się problemem klasyfikacji danych niekompletnych. Motywacja do podjęcia badań ma swoje źródło w medycynie, gdzie bardzo często występuje zjawisko braku danych. Najpopularniejszą metodą radzenia sobie z tym problemem jest imputacja danych, będąca uzupełnieniem brakujących wartości na podstawie statystycznych zależności między cechami. W moich badaniach przyjąłem inną strategię rozwiązania tego problemu. Wykorzystując opracowane wcześniej klasyfikatory można przekształcić je do formy, która zwraca przedział możliwych predykcji. Następnie, poprzez zastosowanie operatorów agregacji oraz metod progowania, można dokonać finalnej klasyfikacji. W niniejszej pracy pokazuję jak dokonać ww. przekształcenia klasyfikatorów oraz jak wykorzystać strategie agregacji danych przedziałowych do klasyfikacji. Opracowane przeze mnie metody podnoszą jakość klasyfikacji danych niekompletnych w problemie wspomagania diagnostyki guzów jajnika. Dodatkowa analiza wyników na zewnętrznych zbiorach danych z repozytorium uczenia maszynowego Uniwersytetu Kalifornijskiego w Irvine (UCI) wskazuje, że przedstawione metody są komplementarne z imputacją.

Contents

1 Introduction

In this thesis I elaborate on a problem of importance in medicine. This work is a result of collaboration with specialists from the Division of Gynaecological Surgery, Poznan University of Medical Sciences. The main goal was to support physicians in the process of prediction of ovarian tumour malignancy. Recent statistics show that the mortality rate is still alarming in some member states of the European Union, as shown in Figure 1.1. The latest statistics from the United States show that ovarian cancer is among the top five leading types of cancer deaths [1].

One of the issues that can delay effective medical treatment is a shortage of experienced gynaecologists. In general, years of experience are necessary to become a professional who is able to correctly detect and classify tumours in their early stages. For this reason, it is desirable to equip inexperienced physicians with an effective preoperative model. In recent years, two possible approaches have emerged. Their aim is to approximate the model of subjective assessment [4]. In the first approach, through scoring systems, points are assigned for the presence of certain features in a patient. If the sum of the assigned points exceeds a threshold, this is taken to indicate malignancy of the tumour. This approach, due to its simplicity and effectiveness, has resulted in a wide range of such models [5]–[7].

The second approach exploits more sophisticated mathematical models. The basic concept utilises rule-based systems through simple schemes of reasoning [8] and rough sets [9]. Recent developments in the field of machine learning have led to the construction of new models, from logistic regression [10]–[12], through artificial neural networks [13], [14], support vector machines and Bayesian networks [15], [16], to neuro-fuzzy networks [17].

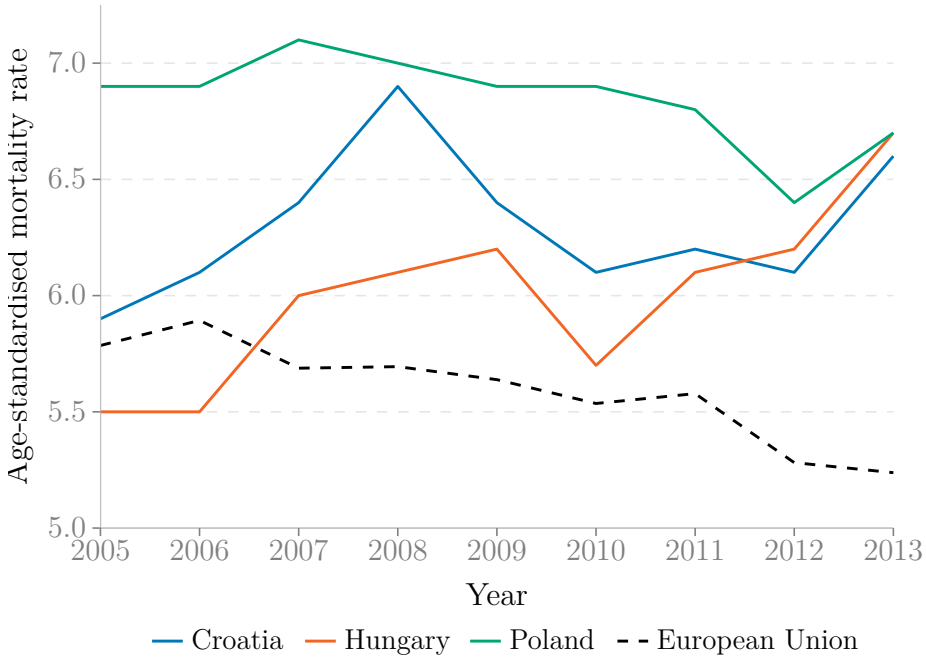

Figure 1.1: Ovarian tumour mortality rates in the European Union and selected member states between 2005 and 2013. The age-standardised rate (world) is expressed per 100 000 persons at risk. Source: [2], [3].

There are also approaches that take benefits from the two aforementioned solutions. The risk malignancy index model (RMI) combines a scoring system with a formal mathematical model [18], whereas the Gynecologic Imaging Report and Data System (GI-RADS) is a rule-based scoring system [19].

Notably, the medical community has recently made additional efforts to establish a discrimination algorithm on the basis of ultrasound images [20]–[22]. Although the accuracy of this approach is reasonably high, problems currently arise in connection with the sample size and variability of tumours. Nevertheless, image-based solutions show great future promise due to the emergent development of deep learning algorithms.

Such a variety of models has resulted in different levels of classification performance and different sets of considered features [23]. The general characteristics of the models can also be particularised. Some of them have a tendency to be more likely to classify tumours as benign (*liberal* models) and some to classify them as malignant (*conservative* models).

Another problem with the models is that they assume a complete and reliable set of input features. This is in contradiction with the phenomenon of data uncertainty in medicine [24]. Missing values might result from the health status of a patient, making a particular examination impossible to process, from the financial costs of an examination, or from the fact that an institution is not equipped with the necessary medical devices. These and related circumstances should be taken into account when a prediction model is being constructed.

In many real-life problems one can handle missing data through the process of imputation, that is inserting values on the basis of statistical relations among features. Although such an approach might be reasonable in general when applied to a whole dataset, it is a hazardous methodology when applied to particular cases. This matter has recently been raised and discussed in the medical community [25]. Moreover, there is particular interest in the availability of a simple-to-apply method designed for non-expert practitioners [26]. Hence, there is a need to develop new classification methods that do not rely on imputation in case of missing data. Another motivating factor is the fact that, despite recent rapid development in artificial intelligence and machine learning, physicians still diagnose illnesses correctly twice as often as computer algorithms [27].

In this research I focus on a sub-field of supervised learning, namely the binary classification of incomplete data. In particular, I search for non-imputation methods designed for the classification of incomplete data. I shall demonstrate that the aggregation of interval data enables a reduction in the impact of information incompleteness on the quality of classification in the problem of ovarian tumour differentiation. The research is focused on real-world applications; thus, we can assume the finiteness of the domains of features and other numerical subsets.

The main research objective is to develop a new procedure for ensemble classification of incomplete data. More specifically, in this thesis I:

- develop algorithms for the uncertaintification of classifiers, so that they return an interval of possible predictions in case of missing data;
- describe an original method of aggregation and thresholding of a set of interval decisions;

- evaluate the proposed solution in the problem of supporting ovarian tumour diagnosis;
- evaluate the proposed solution on commonly used machine learning datasets.

The work has the following structure. In Chapter 2 I describe basic definitions used throughout the document. In Chapter 3 I propose an original method for transforming classifiers to uncertaintified form, and I describe how to aggregate and make decisions based upon interval predictions. Normally, through imputation we check the performance of the classification process when we complete the dataset. A more interesting case is the investigation of classifiers when we remove even more data; this is the topic of the next two chapters. In Chapter 4 I assess the proposed methodology in the context of supporting ovarian tumour diagnosis. In Chapter 5 I evaluate the methodology on UCI repository datasets. Finally, in Chapter 6 I summarise the results and conclusions relating to the developed methods.

In the appendices I list the aggregation operators used (Appendix A), thresholding strategies (Appendix B), complexity analysis of the developed algorithms (Appendix C) and detailed results for evaluation on UCI repository datasets (Appendix D).

Finally, this thesis expands on material reported in previous articles that I have published in collaboration with members of the faculty and medical project. The article [28] focuses on the imprecision of data obtained by a gynaecologist during examinations. The article [29] describes a medical dataset and performance results of common ovarian tumour classifiers. The algorithms for decision-making in case of data incompleteness are elaborated in [30]–[32] – the results from those papers are contained in Chapter 3. Since the research is focused on medical applications, an overview of the implemented *OvaExpert* system appears in [33]–[35]. Additional approaches to medical classification using similarity measures and cardinality can be found in [35], [36] respectively.

2 Basic definitions

In this chapter I would like to give definitions and explanations of mathematical terms and algorithms used throughout the dissertation. Some of them are illustrated by examples. The following definitions relate to data mining, machine learning and imputation. The definitions given are based on the state-of-art literature related to data mining [37], statistical learning [38]–[41], performance evaluation [42] and imputation [43].

2.1 Elements of a dataset

Let us define some essential concepts related to sets of data.

An **instance** is a vector $\mathbf{x} = (x_1, \ldots, x_n)$ such that $x_i \in X_i$. An element x_i of the instance $\mathbf{x}$ is called an **attribute** (or a **feature**) and n is a number of attributes that describe the instance.

An instance is an input vector to a classification algorithm. The domain X_i of the attribute can be either **numeric** or **categorical**.[1] In the former case, the domain is either a closed interval $[a, b]$ of real numbers ($X_i \subset \mathbb{R}$) or a finite subset of integers ($X_i \subset \mathbb{Z}$); notably, $\min X_i$ and $\max X_i$ exist. In the latter case, values of an attribute are pre-specified by a finite set of possibilities (e.g. $X_i = \{\text{“a”}, \text{“b”}, \ldots, \text{“z”}\}$).

A **domain of an instance** is defined as $X = X_1 \times \ldots \times X_n$.

A **class** is an outcome value $y \in Y$ associated with an instance $\mathbf{x}$. Throughout this work we consider only a binary classification, hence $Y = \{y_1, y_2\}$, where the y_i's are pre-specified by a finite set of possibilities.

[1] Although in the literature there are more levels of distinction of attributes, in most practical cases of machine learning problems this division is sufficient [see 37, chapter 2].

A **dataset** is a collection D of instances associated with classes. The number of instances in the dataset D is denoted as $|D|$.

Example 2.1. An excerpt from a dataset describing quality of wine ([see 44], [45]) is contained in Table 2.1.

Table 2.1: Excerpt from *wine quality* dataset

No.	pH	alcohol	free sulphur dioxide	colour	quality
1	3.25	9.0	54	white	bad
2	2.82	13.2	14	white	good
3	3.36	10.1	4	red	bad
4	3.03	10.2	19	white	bad
...	...	...	...	...	...
6497	3.29	10.1	12	red	good

Three attributes are numeric:

1. pH (X_1),
2. alcohol (X_2),
3. free sulphur dioxide (X_3).

One feature – colour (X_4) – is categorical. The classes – quality – are denoted by Y. The domain of the features and the class variables are the following:

$$\begin{aligned} X_1 &= [2.72, 4.2]\,, \\ X_2 &= [8.0, 14.9]\,, \\ X_3 &= [1, 289]\,, \\ X_4 &= \{\text{“red”}, \text{“white”}\}\,, \\ Y &= \{\text{“good”}, \text{“bad”}\}\,. \end{aligned}$$

The number of instances $|D|$ is equal to 6497.

2.2 Classification models

Let us define functions that can operate on instances from a dataset. More specifically, we need functions that can predict a class for a given set of features.

A **scoring function** is a function f such that $f : X \to \mathbb{R}$.

A **classification model** (or a **classifier**) is a function g such that $g : X \to Y$ with a **threshold** (or **cutoff**) $\theta \in \mathbb{R}$, such that

$$g(\mathbf{x}) = \begin{cases} y_1, & \text{if} \quad f(\mathbf{x}) > \theta \\ y_2, & \text{otherwise} \end{cases} .$$

The function g has a construction such that, firstly, it returns a *raw* prediction $f(\mathbf{x})$, which can be interpreted as a score, probability or possibility of belonging to a class; and secondly, with the use of some threshold value it assigns one of two possible classes. In practice, a classifier can output raw predictions from the unit interval $[0, 1]$. Note that a classifier does not have to use all of the attributes in the prediction process.

The following definitions give classic examples of classification models.

2.2.1 Scoring system

A **scoring function of a scoring system** is a function f^{sco} such that

$$f^{\text{sco}}(\mathbf{x}) = \sum_{i=1}^{n} q_i(x_i) \,,$$

where $q_i : X_i \to Q_i \subset \mathbb{N}_0$, $\min Q_i$ and $\max Q_i$ exist.

The interpretation of the function q_i is that it assigns some amount of points for the value of an attribute x_i. A common case is when the q_i's are defined as m step functions that assign non-negative points, i.e.

$$q_i(x_i) = \sum_{j=1}^{m} \gamma_j s_j(x_i) \,,$$

$$s_j(x_i) = \begin{cases} 1, & \text{if} \quad x_i \in S_j \\ 0, & \text{otherwise} \end{cases} ,$$

where the γ_j's are non-negative points given for the value of x_i, the s_j's are step functions, and the S_j's are domains of the step functions for giving specific points, where $\bigcup_{j=1}^{m} S_j = X_i$, $S_j \cap S_k = \varnothing$ for $j \neq k$.

Observe that $f^{\text{sco}} : X \to [0, \sum_{i=1}^{n} \max Q_i]$ for the aforementioned assumptions on the functions q_i.

A **scoring system** is a function g^{sco} with a threshold $\theta^{\text{sco}} \in \mathbb{R}$ such that

$$g^{\text{sco}}(\mathbf{x}) = \begin{cases} y_1, & \text{if} \quad f^{\text{sco}}(\mathbf{x}) > \theta^{\text{sco}} \\ y_2, & \text{otherwise} \end{cases} .$$

Example 2.2. Let us define the following scoring system:

$$g_1(\mathbf{x}) = \begin{cases} \text{“good”}, & \text{if} \quad f_1(\mathbf{x}) > \theta_1 \\ \text{“bad”}, & \text{otherwise} \end{cases} ,$$

where $\theta_1 = 3$ and f_1 is given as a set of rules representing decreasing step functions (see Table 2.2).

Table 2.2: Rules of the example scoring system

Feature	Range	Points
pH	$[2.72, 3.0)$	0
	$[3.0, 3.7)$	1
	$[3.7, 4.2]$	2
alcohol	$[8.0, 9.0)$	0
	$[9.0, 14.9)$	1
free sulphur dioxide	$[1, 20)$	3
	$[20, 90)$	2
	$[90, 289]$	0

Suppose we have an instance $\mathbf{x}_1 = (2.9, 9.5, 15)$, where the elements of the vector correspond respectively to the pH, alcohol and free sulphur dioxide attributes. Then we have

$$f_1(\mathbf{x}_1) = 4 > \theta_1, \text{ thus, } g_1(\mathbf{x}_1) = \text{“good”} .$$

The instance $\mathbf{x}_1$ is classified as good wine.

Example 2.3. Given the rapid development of machine learning techniques, one may wonder whether scoring systems are still relevant and useful. Undoubtedly, their power lies in their computational simplicity and acceptable level of effectiveness. Banking institutions use this approach to estimate customer credit ratings, but the exact formulae used

are not publicly available. A scoring system with a known formula that is extensively used on a daily basis is the Polish government system of profiling support for the unemployed [46]. A respondent answers 24 questions and the system assigns points (from 0 to 10) for each answer. The sum of the points indicates one of three possible profiling support groups, i.e. what kind of support the employment agency ought to provide to the unemployed person.

2.2.2 Logistic regression

A **scoring function of a logistic regression** is a function $f^{\text{lgr}} : X \to (0, 1)$ such that

$$f^{\text{lgr}}(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{uv})},$$

where

$$\mathbf{u} = (u_0, u_1, \ldots, u_n),$$
$$\mathbf{v} = (1, x_1, \ldots, x_n)^\mathsf{T},$$

$\mathbf{v}$ is a parameter vector, $\mathbf{u}$ is a weight vector and $\mathbf{u}$,$\mathbf{v} \in \mathbb{R}^{n+1}$.

A **logistic regression** is a function g^{lgr} with a threshold $\theta^{\text{lgr}} \in (0, 1)$ such that

$$g^{\text{lgr}}(\mathbf{x}) = \begin{cases} y_1, & \text{if} \quad f^{\text{lgr}}(\mathbf{x}) > \theta^{\text{lgr}} \\ y_2, & \text{otherwise} \end{cases}.$$

Example 2.4. Let us define the following logistic regression:

$$g_2(\mathbf{x}) = \begin{cases} \text{“good”}, & \text{if} \quad f_2(\mathbf{x}) > \theta_2 \\ \text{“bad”}, & \text{otherwise} \end{cases},$$

where f_2 is given as the function f^{lgr},

$$\mathbf{u} = (-8.6, -0.33, 0.85, -0.06),$$
$$\mathbf{v} = (1, x_1, x_2, x_3)^\mathsf{T},$$

x_1, x_2 and x_3 denote values of pH, alcohol and free sulphur dioxide respectively, and $\theta_2 = 0.6$.

For the instance $\mathbf{x}_1$ from Example 2.2 we have

$$f_2(\mathbf{x}_1) \approx 0.08 < \theta_2, \text{ thus, } g_2(\mathbf{x}_1) = \text{“bad”}.$$

The instance is classified as bad wine.

2.2.3 Classification tree

Let us define a binary tree (T, E) with $t \in T$ nodes, the set of edges E and height ρ. Leaves determine membership of a class, all non-leaves are splitting rules, and all nodes except for the root have assigned probabilities of belonging to the classes y_1 and y_2.

A **scoring function of a classification tree** is a function $f^{\text{tree}} : X \to [0, 1]$ in the following form: given a binary tree (T, E) and instance $\mathbf{x}$, start from the root and go down to the leaves according to the splitting rules and values of attributes; when a terminal node (leaf) is reached, return a probability of belonging to the class y_2.

A **classification tree** is a function g^{tree} with a threshold $\theta^{\text{tree}} \in (0, 1)$ such that

$$g^{\text{tree}}(\mathbf{x}) = \begin{cases} y_1, & \text{if} \quad f^{\text{tree}}(\mathbf{x}) > \theta^{\text{tree}} \\ y_2, & \text{otherwise} \end{cases}.$$

Note that θ^{tree} is typically equal to 0.5.

Example 2.5. Let us define the following classification tree:

$$g_3(\mathbf{x}) = \begin{cases} \text{“good”}, & \text{if} \quad f_3(\mathbf{x}) > \theta_3 \\ \text{“bad”}, & \text{otherwise} \end{cases},$$

where $\theta_3 = 0.5$ and f_3 is given as the binary tree depicted in Figure 2.1. For the instance $\mathbf{x}_1$ from Example 2.2 we have

$$f_3(\mathbf{x}_1) = 0.82 > \theta_3, \text{ thus, } g_3(\mathbf{x}_1) = \text{“good”}.$$

The instance is classified as good wine.

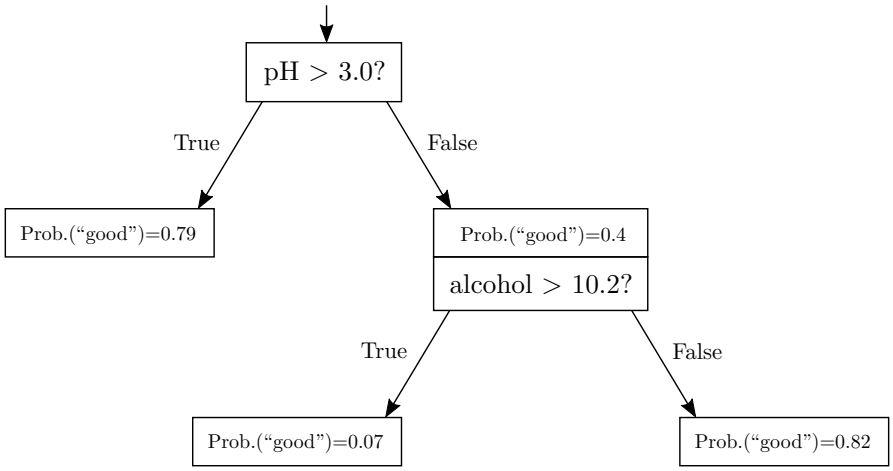

Figure 2.1: Example classification tree

2.2.4 Ensemble of classifiers

An **ensemble classification model** (or an **ensemble classifier**) is a function h with a threshold $\theta^{\text{ens}} \in \mathbb{R}$, such that

$$h(\mathbf{x}) = \begin{cases} y_1, & \text{if} \quad f_0\left(f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})\right) > \theta^{\text{ens}} \\ y_2, & \text{otherwise} \end{cases}.$$

We assume that an ensemble classifier uses a collection of scoring functions. However, in many practical solutions a function f_i can be replaced with a corresponding g_i ($i \geq 1$). In consequence, g_0 may be, for example, a simple majority vote.

Example 2.6. Let us consider the scoring functions f_1, f_2 and f_3 defined in Examples 2.2, 2.4 and 2.5 respectively. We define an ensemble classifier with $\theta_0 = 0.5$, such that

$$h_0(\mathbf{x}) = \begin{cases} \text{“good”}, & \text{if} \quad f_0\left(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\right) > \theta_0 \\ \text{“bad”}, & \text{otherwise} \end{cases},$$

$$f_0(a, b, c) = \frac{a/6 + b + c}{3}.$$

For the instance $\mathbf{x}_1$ from Example 2.2 we have

$$f_0(\mathbf{x}_1) \approx \frac{0.67 + 0.08 + 0.82}{3} \approx 0.52 > \theta_0, \text{ thus, } h_0(\mathbf{x}_1) = \text{“good”}.$$

The instance is classified as good wine.

2.3 Performance measures

Let us define a binary **confusion matrix** of a classifier g on a dataset D. With Table 2.3, one can calculate performance measures of a classifier.

Table 2.3: Binary confusion matrix

		Predicted class of $\mathbf{x}$	
		y_1	y_2
Actual class of $\mathbf{x}$	y_1	True negative (TN)	False positive (FP)
	y_2	False negative (FN)	True positive (TP)

The most common measures are **accuracy** (ACC) and two metrics with a single-class focus, **sensitivity** (SEN) and **specificity** (SPE):

$$\text{ACC} = \frac{\#\text{TP} + \#\text{TN}}{\#\text{TP} + \#\text{TN} + \#\text{FP} + \#\text{FN}},$$

$$\text{SEN} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}},$$

$$\text{SPE} = \frac{\#\text{TN}}{\#\text{TN} + \#\text{FP}}.$$

Along with the three metrics, let us define an additional performance measure, **decisiveness** (DEC):

$$\text{DEC} = \frac{\#\text{instances in } D \text{ for which } g \text{ is able to predict classes}}{\text{total } \#\text{instances in } D}.$$

A **cost matrix** is a numerical matrix where each value corresponds with the confusion matrix and reflects a reward or loss for a particular classifier decision. Use of the cost matrix might be useful when different misclassification types have imbalanced importance or weight.

Example 2.7. Table 2.4 shows an example cost matrix. True positives and true negatives have no reward, false positives have a cost of 2, and a false negative has a cost of 5.

Table 2.4: Example cost matrix

		Predicted class of $\mathbf{x}$	
		y_1	y_2
Actual class of $\mathbf{x}$	y_1	0	2
	y_2	5	0

2.4 Error estimation methods

In the process of learning of the classifier on a large dataset, one may use a traditional dataset division into training-test or training-validation-test sets. However, in many situations the available dataset is not large. In this case, with a desired performance measure PERF, the ***k*-fold cross-validation** algorithm can be used – a descriptive listing of steps is given in Algorithm 2.1.

Another problem arises when the distribution of classes is imbalanced. In this case a **stratified *k*-fold cross-validation** algorithm can be applied, as described in Algorithm 2.2. In this version the folds preserve the approximate global distribution of classes.

In a practical case one may wish to choose one model from a set of possible models. A model selection procedure can be combined with cross-validation into **nested *k*-fold cross-validation**. The algorithm is described in Algorithm 2.3. Notice that in this procedure the k-fold cross-validation can be replaced with a stratified version.

Algorithm 2.1: k-fold cross-validation

1 Divide D of size of m instances into k non-overlapping subsets D_i of approximate size $\frac{m}{k}$.
2 **For each** fold $i \in \{1, \ldots, k\}$:
3 $\quad D^{\text{train}} = D \setminus D_i$.
4 $\quad D^{\text{test}} = D_i$.
5 $\quad$ Train classifier g_i on D^{train}.
6 $\quad$ Obtain performance measure PERF_i of g_i achieved on D^{test}.
7 Report average performance measure across folds, i.e. $\frac{1}{k}\sum_{i=1}^{k} \text{PERF}_i$.
8 Learn classifier g on D.

Algorithm 2.2: Stratified k-fold cross-validation

1 Divide D into two datasets each containing only one class, i.e. D_{y_1} and D_{y_2}.
2 Generate k non-overlapping subsets $D^i_{y_1}$ and $D^i_{y_2}$ with approximately the same number of instances of each class in all k subsets.
3 Merge consecutive subsets of $D^i_{y_1}$ and $D^i_{y_2}$, in order to obtain k subsets reflecting the original class distribution.
4 Perform k-fold cross-validation on these k subsets.

Algorithm 2.3: Nested k-fold cross-validation

1 Divide D into k folds D_i.
2 **For each** fold $i \in \{1, \ldots, k\}$:
3 Divide $D \setminus D_i$ into k folds D_j
4 Perform model selection of g_i on the folds D_j using k-fold cross-validation.
5 Learn g_i on $D \setminus D_i$.
6 Obtain performance measure $\textsc{Perf}_i$ of g_i achieved on D_i.
7 Report average performance measure across folds, i.e. $\frac{1}{k}\sum_{i=1}^{k} \textsc{Perf}_i$.
8 Perform model selection of g on the folds of D using k-fold cross-validation.
9 Learn classifier g on D.

2.5 Imputation

In many cases the instances may have some missing attributes. Reasons for this anomaly may relate to, for example, malfunction of a measuring device, corruption of a storage device, human error in data input, etc. In practical applications a missing value is commonly denoted by the symbol `NA`. As a consequence, an attribute x_i has the extended domain $X_i \cup \{\texttt{NA}\}$.

In case of missing data, a conventional classifier is often unable to make a prediction. Naturally, one can avoid this problem by choosing a classifier with an embedded method of handling missing data, e.g. a binary classification tree where nodes also check whether an attribute is available. Unfortunately, this is not always applicable in real-world problems; moreover, end users often naively assume that all attributes will be complete in the future.

For this reason, several methods of dealing with missing data have been developed in recent years. An extensive overview of statistical data editing and imputation can be found in [43]. The simplest and most straightforward method of dealing with missing data might be case-wise deletion of instances with missing values from the dataset, but in this case the data loss may be too great to be acceptable. A different approach may involve inserting a median or mode of the attribute, but this is too naive when relations among attributes are complex. The most practical approach to imputation is through random forests [47] and multivariate imputation by chained equations [48]. An imputation method will be denoted by Imp throughout this dissertation.

3 Interval classification procedure

In the previous chapter I defined basic terms and definitions relating to datasets and classifiers. Normally, if an instance has missing values, one can handle this problem using either imputation or a special form of classifier with a native capability of dealing with missing values. However, this is not the case in the medical problem being considered here. In this chapter I present a novel method of dealing with this problem. A general outline of the procedure was published in [32].

3.1 Interval modelling

In order to handle missing data for an attribute, the domain of the attribute must be extended by the element `NA`. That is, $\mathbf{x} = (x_1, \ldots, x_n)$ where $x_i \in X_i \cup \{\texttt{NA}\}$. This standard approach has two major drawbacks. Firstly, a new separate value must be introduced to represent missing data. Secondly, often such a value cannot be handled by classical classifiers, which leads to inability to make any prediction. However, this issue may be modelled in a different way. For the sake of simplicity we shall focus on numeric attributes.

Let us introduce an interval version of the domain of the attribute X_i, which is denoted as the set of all nonempty closed subintervals of X_i, i.e.

$$\hat{X}_i = \mathcal{I}_{X_i} = \{[a, b] : [a, b] \subseteq X_i\}.$$

We can define an interval domain of the interval instance, i.e. $\hat{X} = \hat{X}_1 \times \ldots \times \hat{X}_n$. Now, for each instance $\mathbf{x} \in X$ (with or without missing values) we can define its interval equivalent, i.e.

$$\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n) = ([\underline{x}_1, \bar{x}_1], \ldots, [\underline{x}_n, \bar{x}_n]) \in \hat{X}\,,$$

where

$$\underline{x}_i = \begin{cases} x_i, & \text{if} \quad x_i \neq \texttt{NA} \\ \min\limits_{x \in X_i} x, & \text{if} \quad x_i = \texttt{NA} \end{cases},$$

$$\bar{x}_i = \begin{cases} x_i, & \text{if} \quad x_i \neq \texttt{NA} \\ \max\limits_{x \in X_i} x, & \text{if} \quad x_i = \texttt{NA} \end{cases}.$$

These definitions allow us to describe the value of each attribute in a uniform way by an interval. With this approach the instances in the dataset can be similarly processed by a classifier. In the practical case, the representation of the attribute can be twofold: a set of all possible values (if the value is not present) or a point (if the value is present). This representation has an additional practical advantage: it stores and encodes within the data the possible ranges for missing values. This might be particularly helpful when the source description of attributes is not available in a general preprocessing step.

Example 3.1. Let us consider two instances:

$$\mathbf{x}_2 = (\texttt{NA}, 9.5, 3)\,,$$
$$\mathbf{x}_3 = (2.8, \texttt{NA}, 32)\,,$$

where the respective attributes in the vectors are pH, alcohol and free sulphur dioxide, with the domains defined in Example 2.1. The interval representations of the instances are as follows:

$$\hat{\mathbf{x}}_2 = ([2.72, 4.2], [9.5, 9.5], [3, 3])\,,$$
$$\hat{\mathbf{x}}_3 = ([2.8, 2.8], [8.0, 14.9], [32, 32])\,.$$

3.2 Uncertaintification of classifiers

In the next step we have to enable the scoring functions to work with the interval representation of instances. We say that a vector $\mathbf{x}$ is an embedded vector of $\hat{\mathbf{x}}$, denoted by $\mathbf{x} \in_{\mathsf{E}} \hat{\mathbf{x}}$, when for all $i \in \{1, \ldots, n\}$ the attribute $x_i \in \hat{x}_i$. We can define an **uncertaintified scoring function** as

$$\hat{f}(\hat{\mathbf{x}}) = \{f(\mathbf{x}) : \mathbf{x} \in_{\mathsf{E}} \hat{\mathbf{x}}\}\,. \tag{3.1}$$

The resultant interval represents all possible predictions that can be made based on values of an instance in which every missing value is replaced with all possible values for that attribute. The more incomplete the instance, the more uncertain the prediction. Observe that in many cases it is still possible to make a proper decision, since some quantity of missing values is acceptable and will not affect the final result significantly.

The result of reasoning based on the interval representation can also be denoted as an interval, i.e.

$$\hat{f}(\hat{\mathbf{x}}) = \left[\min_{\mathbf{x} \in_{\mathsf{E}} \hat{\mathbf{x}}} f(\mathbf{x}), \max_{\mathbf{x} \in_{\mathsf{E}} \hat{\mathbf{x}}} f(\mathbf{x})\right]\,. \tag{3.2}$$

These two definitions are equivalent whenever the scoring function is continuous. In other cases, Formula (3.2) gives a very good approximation of Formula (3.1), and we therefore adopt Formula (3.2) as the definition of $\hat{f} : \hat{X} \to \mathcal{I}_{[0,1]}$. We can assume that the value 0.5 will serve as a separating point for classes y_1 and y_2.

We can also define an **uncertaintified classification model**:

$$\hat{g}(\hat{\mathbf{x}}) = \begin{cases} y_1, & \text{if} \quad \hat{f}(\hat{\mathbf{x}}) \subset (0.5, 1] \\ y_2, & \text{if} \quad \hat{f}(\hat{\mathbf{x}}) \subset [0, 0.5] \\ \texttt{NA}, & \text{otherwise} \end{cases}\,.$$

The interpretation of this classifier is that if the returned interval prediction is wholly greater than, or wholly less than or equal to, 0.5, then it can still assign a class to the instance. Otherwise, the interval is too wide and includes the separation point, hence no decision should be made.

The following subsections describe an algorithmic approach to the uncertaintification of scoring functions.

3.2.1 The case of scoring system

Suppose we have a scoring system g^{sco} with a threshold θ^{sco} as defined in Section 2.2.1 (i.e. with increasing step functions). We can calculate $\hat{f}(\hat{\mathbf{x}})$ directly as follows:

$$\hat{f}(\hat{\mathbf{x}}) = [f^{\text{sco}}(\underset{\sim}{\mathbf{x}}), f^{\text{sco}}(\widetilde{\mathbf{x}})]\,,$$

where

$$\begin{aligned}
\underset{\sim}{\mathbf{x}} : \underset{\sim}{x}_i &= \begin{cases} x_i, & \text{if} \quad x_i \neq \texttt{NA} \\ a \in \underset{x \in [\underline{x}_i, \bar{x}_i]}{\arg\min}\, q_i(x), & \text{if} \quad x_i = \texttt{NA} \end{cases}, \\
\widetilde{\mathbf{x}} : \widetilde{x}_i &= \begin{cases} x_i, & \text{if} \quad x_i \neq \texttt{NA} \\ b \in \underset{x \in [\underline{x}_i, \bar{x}_i]}{\arg\max}\, q_i(x), & \text{if} \quad x_i = \texttt{NA} \end{cases},
\end{aligned} \tag{3.3}$$

where a, b are arbitrary elements of the resulting sets.

Recall that $f^{\text{sco}} : X \to [0, \sum_{i=1}^{n} \max Q_i]$. Hence, we have to normalise the result of $\hat{f}(\hat{\mathbf{x}})$ so that it is contained in $\mathcal{I}_{[0,1]}$, and rescale through θ^{sco} in order to make the value 0.5 the separating point for classes y_1 and y_2, i.e.

$$\hat{f}(\hat{\mathbf{x}}) = \left[\xi\left(f^{\text{sco}}(\underset{\sim}{\mathbf{x}})\right), \xi\left(f^{\text{sco}}(\widetilde{\mathbf{x}})\right)\right]\,,$$

where

$$\xi(c) = \begin{cases} \dfrac{0.5c}{\theta^{\text{sco}}}, & \text{if} \quad c \leq \theta^{\text{sco}} \\ \dfrac{0.5(c - \theta^{\text{sco}})}{(\sum_{i=1}^{n} \max Q_i) - \theta^{\text{sco}}} + 0.5, & \text{if} \quad c > \theta^{\text{sco}} \end{cases}.$$

Example 3.2. Let us consider g_1 and f_1 from Example 2.2 and $\mathbf{x}_2$, $\mathbf{x}_3$ from Example 3.1. By Formula (3.3), for the first instance we obtain

$$\begin{aligned}
\underset{\sim}{\mathbf{x}}_2 &= (2.72, 9.5, 3)\,, \\
\widetilde{\mathbf{x}}_2 &= (4.2, 9.5, 3)\,,
\end{aligned}$$

and for the second one we obtain

$$\underset{\sim}{\mathbf{x}}_3 = (2.8, 8.0, 32)\,,$$
$$\widetilde{\mathbf{x}}_3 = (2.8, 14.9, 32)\,.$$

Notice that for $\mathbf{x}_2$ we choose arbitrary elements of the resulting sets arg min and arg max. Now we can compute lower and upper numeric prediction bounds, i.e.

$$\hat{f}_1(\hat{\mathbf{x}}_2) = \left[\xi\left(f_1(\underset{\sim}{\mathbf{x}}_2)\right), \xi\left(f_1(\widetilde{\mathbf{x}}_2)\right)\right] \approx [0.67, 1]\,,$$
$$\hat{f}_1(\hat{\mathbf{x}}_3) = \left[\xi\left(f_1(\underset{\sim}{\mathbf{x}}_2)\right), \xi\left(f_1(\widetilde{\mathbf{x}}_2)\right)\right] \approx [0.33, 0.5]\,.$$

We can also compute predictions made by the uncertaintified classifier, i.e.

$$\hat{f}_1(\hat{\mathbf{x}}_2) \approx [0.67, 1] \subset (0.5, 1], \text{ hence } \hat{g}_1(\hat{\mathbf{x}}_2) = \text{“good”}\,,$$
$$\hat{f}_1(\hat{\mathbf{x}}_3) \approx [0.33, 0.5] \subset [0, 0.5], \text{ hence } \hat{g}_1(\hat{\mathbf{x}}_3) = \text{“bad”}\,.$$

Observe that despite the missing values, we can still make a prediction by means of $\hat{f}_1$ and $\hat{g}_1$. In fact, no matter what the real value is for the missing one, it does not influence the prediction.

3.2.2 The case of logistic regression

Suppose we have a logistic regression g^{lgr} with a threshold θ^{lgr} and weights $\mathbf{u}$, as defined in Section 2.2.2. We can calculate $\hat{f}(\hat{\mathbf{x}})$ directly as follows:

$$\hat{f}(\hat{\mathbf{x}}) = [f^{\text{lgr}}(\underset{\sim}{\mathbf{x}}), f^{\text{lgr}}(\widetilde{\mathbf{x}})]\,,$$

where

$$\begin{aligned}
\underset{\sim}{\mathbf{x}} : \underset{\sim}{x}_i &= \begin{cases} x_i, & \text{if} \quad x_i \neq \texttt{NA} \\ \min\limits_{x \in [\underset{\sim}{x}_i, \widetilde{x}_i]} x, & \text{if} \quad x_i = \texttt{NA} \wedge u_i > 0 \\ \max\limits_{x \in [\underset{\sim}{x}_i, \widetilde{x}_i]} x, & \text{if} \quad x_i = \texttt{NA} \wedge u_i < 0 \end{cases}, \\
\widetilde{\mathbf{x}} : \widetilde{x}_i &= \begin{cases} x_i, & \text{if} \quad x_i \neq \texttt{NA} \\ \max\limits_{x \in [\underset{\sim}{x}_i, \widetilde{x}_i]} x, & \text{if} \quad x_i = \texttt{NA} \wedge u_i > 0 \\ \min\limits_{x \in [\underset{\sim}{x}_i, \widetilde{x}_i]} x, & \text{if} \quad x_i = \texttt{NA} \wedge u_i < 0 \end{cases}.
\end{aligned} \tag{3.4}$$

Since $f^{\text{lgr}} : X \to (0,1)$, there is no need to normalise the output range to be within $\mathcal{I}_{[0,1]}$. However, there might be a need to rescale through $\theta = \theta^{\text{sco}}$ so that the value 0.5 will serve as a separating point for classes y_1 and y_2, i.e.

$$\hat{f}(\hat{\mathbf{x}}) = \left[\phi\left(f^{\text{lgr}}(\underset{\sim}{\mathbf{x}})\right), \phi\left(f^{\text{lgr}}(\widetilde{\mathbf{x}})\right)\right],$$

where

$$\phi(a) = \begin{cases} \dfrac{0.5a}{\theta}, & \text{if} \quad a \leq \theta \\ \dfrac{0.5(a-\theta)}{1-\theta} + 0.5, & \text{if} \quad a > \theta \end{cases}.$$

Example 3.3. Let us consider g_2 and f_2 from Example 2.4 and $\mathbf{x}_2$, $\mathbf{x}_3$ from Example 3.1. By Formula (3.4), for the first instance we obtain

$$\begin{aligned} \underset{\sim}{\mathbf{x}}_2 &= (4.2, 9.5, 3), \\ \widetilde{\mathbf{x}}_2 &= (2.72, 9.5, 3), \end{aligned}$$

whereas for the second one

$$\begin{aligned} \underset{\sim}{\mathbf{x}}_3 &= (2.8, 8.0, 32), \\ \widetilde{\mathbf{x}}_3 &= (2.8, 14.9, 32). \end{aligned}$$

Now we can calculate lower and upper numeric prediction bounds, i.e.

$$\begin{aligned} \hat{f}_2(\hat{\mathbf{x}}_2) = \left[\phi\left(f_2(\underset{\sim}{\mathbf{x}}_2)\right), \phi\left(f_2(\widetilde{\mathbf{x}}_2)\right)\right] &\approx [\phi(0.11), \phi(0.17)] \\ &\approx [0.09, 0.14], \\ \hat{f}_2(\hat{\mathbf{x}}_3) = \left[\phi\left(f_2(\underset{\sim}{\mathbf{x}}_3)\right), \phi\left(f_2(\widetilde{\mathbf{x}}_3)\right)\right] &\approx [\phi(0.01), \phi(0.77)] \\ &\approx [0.01, 0.71]. \end{aligned}$$

We can also calculate predictions made by the uncertaintified classifier:

$$\begin{aligned} &\hat{f}_2(\hat{\mathbf{x}}_2) \approx [0.09, 0.14] \subset [0, 0.5], \text{ hence } \hat{g}_2(\hat{\mathbf{x}}_2) = \text{“bad”}, \\ &\hat{f}_2(\hat{\mathbf{x}}_3) \approx [0.01, 0.71], \text{ hence } \hat{g}_2(\hat{\mathbf{x}}_3) = \texttt{NA}. \end{aligned}$$

3.2.3 The case of classification tree

Suppose we have a binary tree (T, E), as defined in Section 2.2.3. This time we have to calculate lower and upper bounds in a different way. Let $\hat{x}_i \in \mathcal{I}_{[0,1]}$ and suppose that node t has a splitting rule utilising x_i. During the prediction process we visit nodes from root to leaves, and according to the splitting rules we eventually reach leaf (node) t. We check whether the values of $\hat{x}_i$ satisfy the splitting rule, and then continue the procedure by concurrently visiting left and right sub-nodes. If all attributes x are available, the procedure ends with a single probability of belonging to class y_1, i.e. $\{p_1\}$. However, this time the modified procedure returns a set of probabilities $P = \{p_1, p_2, \dots\}$. We can calculate $\hat{f}(\hat{\mathbf{x}})$ with the use of P, i.e.

$$\hat{f}(\hat{\mathbf{x}}) = \left[\min_{p \in P} p, \max_{p \in P} p\right] .$$

Since $f^{\text{tree}} : X \to [0, 1]$, there is no need to normalise the output range to be within $\mathcal{I}_{[0,1]}$. Moreover, 0.5 is usually a built-in thresholding value for splitting into two classes, hence the application of a normalising function is also unnecessary. Nevertheless, such a normalisation can be performed, if needed, using the function ϕ.

Example 3.4. Let us consider g_3 and f_3 from Example 2.5 and $\mathbf{x}_2$, $\mathbf{x}_3$ from Example 3.1. We can walk through the tree and obtain interval prediction boundaries, i.e.

$$\hat{f}_3(\hat{\mathbf{x}}_2) = (0.79, 0.82) ,$$
$$\hat{f}_3(\hat{\mathbf{x}}_3) = (0.07, 0.82) .$$

We can also calculate predictions made by the uncertaintified classifier:

$$\hat{f}_3(\hat{\mathbf{x}}_2) = [0.79, 0.82] \subset (0.5, 1], \text{ hence } \hat{g}_3(\hat{\mathbf{x}}_2) = \text{“good”} ,$$
$$\hat{f}_3(\hat{\mathbf{x}}_3) = [0.07, 0.82], \text{ hence } \hat{g}_3(\hat{\mathbf{x}}_3) = \texttt{NA} .$$

3.2.4 Practical guidelines

The aforementioned procedures of uncertaintification show that each type of classifier needs a different approach to force it to return an interval prediction. Since there are many types of prediction models, customising and

describing an uncertaintification procedure for each model is ineffectual in terms of both mathematical notation and computer programming.

Fortunately, the problem of uncertaintification can be thought of as an optimisation problem, where we have to determine minimum and maximum values of a scoring function f for a given specific instance $\mathbf{x}$. In case of a missing attribute x_i we have to set in the optimisation procedure the boundaries given by the domain X_i. In general, it might be very impractical or even impossible to obtain a derivative of a function. For this reason, derivative-free optimisation methods are preferable, e.g. the Nelder–Mead method [49] or particle swarm optimisation [50].

So far we have considered only numeric attributes; however, the question is what to do if a categorical attribute is missing. Although in this case we have to check all possible substitutions, this task can be done independently. Such an operation can be easily programmed and performed concurrently.

Lastly, in case of a missing value, in the interval modelling step we consider the whole possible range of values from a feature domain. Nevertheless, if we have additional knowledge allowing us to discard some possible values or sub-ranges (e.g. a particular value of one attribute may not occur with a given configuration of a second attribute), then we can narrow the attribute intervals. This approach might be particularly useful, since it results in narrower prediction intervals and more confident predictions.

3.3 Aggregation of scoring functions

Assume that we have at our disposal m different classifiers $g_1, \ldots, g_m$. In order to improve performance in the classification of new instances, we can restate the problem as one of group decision-making and information aggregation [51]. We can create a collection of predictions of classifiers with the use of a special construction of ensemble classifier h (and its extension to interval inputs) via aggregation and thresholding.

An n-argument **numeric aggregation operator**[1] is a mapping $\text{AGG} : [0,1]^n \to [0,1]$ with the following property of monotonicity and boundary conditions [51], [52]:

1. if $a_i \leq b_i$ for all $i \in \{1, \ldots, n\}$, then
$$\text{AGG}(a_1, \ldots, a_n) \leq \text{AGG}(b_1, \ldots, b_n),$$
2. $\text{AGG}(0, \ldots, 0) = 0$,
3. $\text{AGG}(1, \ldots, 1) = 1$.

Observe that the above definition can be extended to an **interval aggregation operator**, where the function operates on unit intervals, i.e. $\widehat{\text{AGG}} : \mathcal{I}^n_{[0,1]} \to \mathcal{I}_{[0,1]}$. An intelligible definition of the interval aggregation operator can be found in [53]. Let us denote by L a lattice of non-empty intervals $L = \{[a,b] | (a,b) \in [0,1]^2, a \leq b\}$ with the partial order $\leq_L$ defined as $[a,b] \leq_L [c,d] \Leftrightarrow a \leq c$ and $b \leq d$. The top and bottom elements are respectively $1_L = [1,1]$, $0_L = [0,0]$. A function $f_L : L^n \to L$ is an aggregation function if it is monotone with respect to $\leq_L$ and satisfies $f_L(0_L, \ldots, 0_L) = 0_L$ and $f_L(1_L, \ldots, 1_L) = 1_L$. Here, f_L is equivalent to $\widehat{\text{AGG}}$.

There are four main classes of aggregation operators [51]:

1. averaging, i.e. $\text{AGG}(a_1, \ldots, a_n) \in [a_1 \wedge \ldots \wedge a_n,\ a_1 \vee \ldots \vee a_n]$,
2. conjunctive, i.e. $\text{AGG}(a_1, \ldots, a_n) \leq a_1 \wedge \ldots \wedge a_n$,
3. disjunctive, i.e. $\text{AGG}(a_1, \ldots, a_n) \geq a_1 \vee \ldots \vee a_n$,
4. mixed, i.e. those which do not belong to any of the above mentioned classes.

A detailed list of the aggregation operators used in this dissertation is given in Appendix A.

Maintaining the assumptions given in Section 3.1 and Section 3.2, we can use m normalised interval scoring functions $\hat{f}_i : \hat{X} \to \mathcal{I}_{[0,1]}$ and aggregate their results by means of either a numeric or an interval aggregation operator, i.e. $\text{AGG} : [0,1]^m \to [0,1]$ or $\widehat{\text{AGG}} : \mathcal{I}^m_{[0,1]} \to \mathcal{I}_{[0,1]}$ respectively. Since we operate on intervals produced by $\hat{f}_i$, the numeric aggregation operator AGG has to work on representatives of the input intervals, e.g. the lower bounds or midpoints of the intervals (see Section A.2). In the case of the interval aggregation operator $\widehat{\text{AGG}}$, it can utilise whole input intervals.

[1] We will use terms *aggregation functions* and *aggregation operators* interchangeably in this thesis.

Example 3.5. Let us consider $\hat{f}_1$, $\hat{f}_2$, $\hat{f}_3$ from Examples 3.2–3.4 and $\hat{\mathbf{x}}_2$, $\hat{\mathbf{x}}_3$ from Example 3.1. We can aggregate the results produced by the uncertaintified scoring functions with the use of the simple arithmetic mean. In the numeric mode of aggregation we can operate on, for example, the midpoints of the intervals:

$$\begin{aligned}\textsc{Agg}\left(\hat{f}_1(\hat{\mathbf{x}}_2), \hat{f}_2(\hat{\mathbf{x}}_2), \hat{f}_3(\hat{\mathbf{x}}_2)\right) &= \frac{\frac{0.67+1}{2} + \frac{0.09+0.14}{2} + \frac{0.79+0.82}{2}}{3} \\ &\approx 0.59\,.\end{aligned}$$

$$\begin{aligned}\textsc{Agg}\left(\hat{f}_1(\hat{\mathbf{x}}_3), \hat{f}_2(\hat{\mathbf{x}}_3), \hat{f}_3(\hat{\mathbf{x}}_3)\right) &= \frac{\frac{0.33+0.5}{2} + \frac{0.01+0.71}{2} + \frac{0.07+0.82}{2}}{3} \\ &\approx 0.41\,.\end{aligned}$$

For the interval mode of aggregation we can use interval arithmetic to operate on whole intervals:

$$\begin{aligned}\widehat{\textsc{Agg}}\left(\hat{f}_1(\hat{\mathbf{x}}_2), \hat{f}_2(\hat{\mathbf{x}}_2), \hat{f}_3(\hat{\mathbf{x}}_2)\right) &= \left[\frac{0.67+0.09+0.79}{3}, \frac{1+0.14+0.82}{3}\right] \\ &\approx [0.52, 0.65]\,.\end{aligned}$$

$$\begin{aligned}\widehat{\textsc{Agg}}\left(\hat{f}_1(\hat{\mathbf{x}}_3), \hat{f}_2(\hat{\mathbf{x}}_3), \hat{f}_3(\hat{\mathbf{x}}_3)\right) &= \left[\frac{0.33+0.01+0.07}{3}, \frac{0.5+0.71+0.82}{3}\right] \\ &\approx [0.14, 0.68]\,.\end{aligned}$$

3.4 Thresholding

The result of the aggregation step can be either a numeric value or an interval. To perform this, we need two different classes of functions:

1. a **numeric thresholding strategy**, i.e. $\tau : [0,1] \to \{y_1, y_2, \texttt{NA}\}$,
2. an **interval thresholding strategy**, i.e. $\hat{\tau} : \mathcal{I}_{[0,1]} \to \{y_1, y_2, \texttt{NA}\}$.

A detailed list of the thresholding strategies used in this dissertation is given in Appendix B. A combination of an aggregation operator with a thresholding strategy is called an **aggregation strategy** and denoted as AggStr.

As in Section 3.2, the separating point for classes y_1 and y_2 is chosen to be 0.5. Observe that these functions support the case where the aggregation step delivers either a numeric value or an interval that is not sufficient to make a reliable decision (resulting in the value `NA`).

Example 3.6. Let us continue Example 3.5. For numeric aggregation we can use the following thresholding strategy:

$$\tau_{0.05}(a) = \begin{cases} \text{“good”} & \text{if } a > 0.55 \\ \text{“bad”} & \text{if } a \leq 0.45 \\ \texttt{NA} & \text{otherwise} \end{cases}.$$

For the interval instances we have the following predictions:

$$\tau_{0.05}\left(\text{AGG}\left(\hat{f}_1(\hat{\mathbf{x}}_2), \hat{f}_2(\hat{\mathbf{x}}_2), \hat{f}_3(\hat{\mathbf{x}}_2)\right)\right) = \tau_{0.05}(0.59) = \text{“good”}\,,$$
$$\tau_{0.05}\left(\text{AGG}\left(\hat{f}_1(\hat{\mathbf{x}}_3), \hat{f}_2(\hat{\mathbf{x}}_3), \hat{f}_2(\hat{\mathbf{x}}_3)\right)\right) = \tau_{0.05}(0.41) = \text{“bad”}\,.$$

For interval aggregation we can use

$$\hat{\tau}_{0.01}\left([a,b]\right) = \begin{cases} \text{“good”} & \text{if } a > 0.51 \\ \text{“bad”} & \text{if } b \leq 0.49 \\ \texttt{NA} & \text{otherwise} \end{cases}.$$

For the interval instances we have the following predictions:

$$\begin{aligned} \hat{\tau}_{0.01}\left(\widehat{\text{AGG}}\left(\hat{f}_1(\hat{\mathbf{x}}_2), \hat{f}_2(\hat{\mathbf{x}}_2), \hat{f}_3(\hat{\mathbf{x}}_2)\right)\right) &= \hat{\tau}_{0.01}\left([0.52, 0.65]\right) \\ &= \text{“good”}\,, \\ \hat{\tau}_{0.01}\left(\widehat{\text{AGG}}\left(\hat{f}_1(\hat{\mathbf{x}}_3), \hat{f}_2(\hat{\mathbf{x}}_3), \hat{f}_3(\hat{\mathbf{x}}_3)\right)\right) &= \hat{\tau}_{0.01}\left([0.14, 0.68]\right) \\ &= \texttt{NA}\,. \end{aligned}$$

3.5 Summary of the proposed approach

In the proposed method we have shown how to perform the process of prediction uncertaintification. With the use of this step, the classifiers can return interval predictions for instances with missing values. Then, through aggregation and thresholding, we can rely on independent classifiers and collaboratively make a single decision. By this step we have introduced a novel behaviour, namely the ability to refrain from making

a final decision if the predictions are mutually exclusive or the predictions are too uncertain. A visualisation of this step is seen in Figure 3.1. A combination of all four steps is listed in Algorithm 3.1.

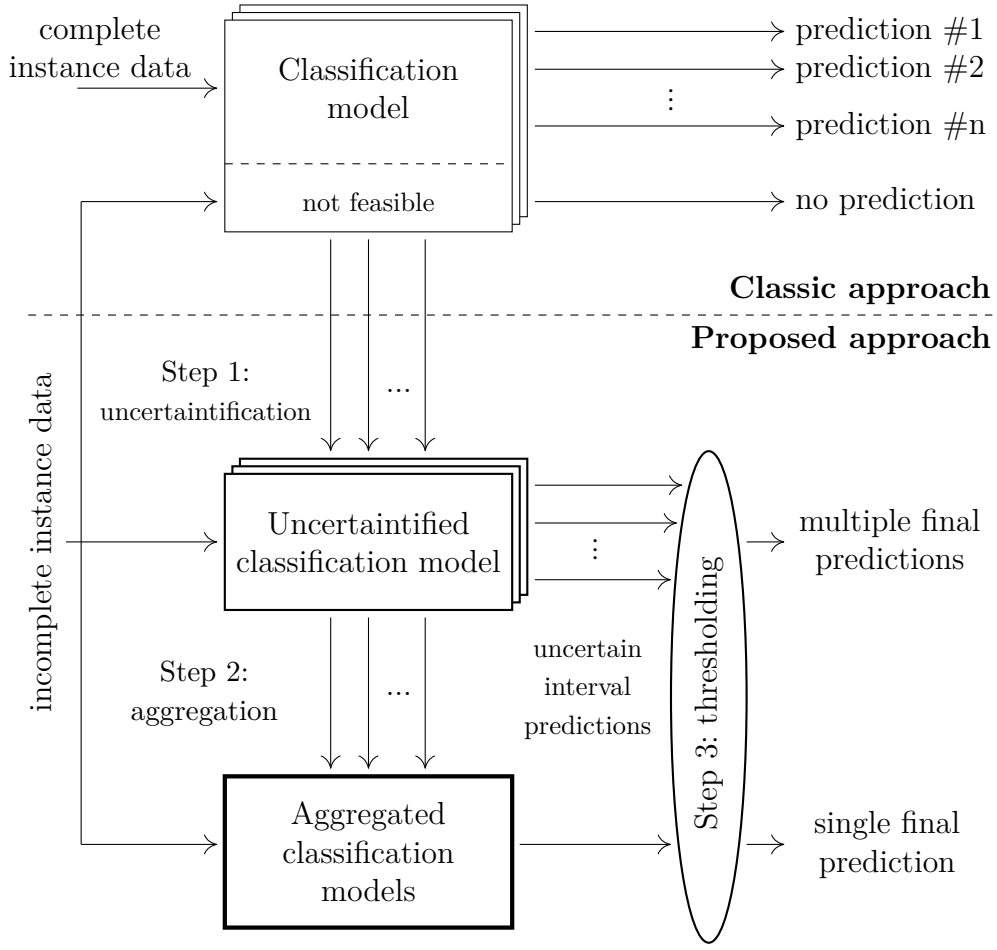

Figure 3.1: A graphical summary of the classical and proposed approaches. Rectangles represent diagnostic classification models at different stages. Vertical arrows represent classification model transformations, i.e. uncertaintification and aggregation. The third step (thresholding) is depicted as an ellipse. Horizontal arrows represent the flow of data of instances and predictions.

Algorithm 3.1: Ensemble classification through aggregation strategy

Input : dataset D of n instances, scoring functions f_i's, compatible aggregation strategy, i.e. $(\textsc{Agg}, \tau)$ or $(\widehat{\textsc{Agg}}, \hat{\tau})$

Output: n predictions $\{y_1, y_2, \texttt{NA}\}$

1 Transform attributes of instances in D to the interval form.

2 Transform scoring functions f_i into uncertaintified scoring functions $\hat{f}_i$.

3 Get interval predictions of $\hat{f}_i$ on transformed D.

4 Aggregate interval predictions by $\textsc{Agg}$ or $\widehat{\textsc{Agg}}$.

5 Return, thresholded by τ or $\hat{\tau}$, aggregated interval predictions.

4 Medical evaluation

In Chapter 3 I presented a method of interval classification for use in case of missing data. This procedure is well-suited to the medical problem described in Chapter 1. In this chapter I show how the proposed approach can be applied in supporting ovarian tumour diagnosis. The results were published in [32].

4.1 Subject of evaluation

The study group consisted of 388 patients diagnosed and treated for ovarian tumours in the Division of Gynaecological Surgery, Poznan University of Medical Sciences, between 2005 and 2015. The distribution of benign and malignant tumours was 61% and 39% respectively. A majority of the patients (56%) had a complete set of attributes as required by diagnostic scales, 40% of the patients had missing values in the range $(0\%, 50\%]$, and the remaining 4% of the patients had more than 50% of values missing. The distribution of missing values depending on malignancy is depicted in Figure 4.1.

Six diagnostic models were selected for the evaluation procedure: two scoring systems [6], [7] and four regression models [10], [54], [55]. Table 4.1 shows the usage of attributes by the models. The features consisted of two groups, the first comprising attributes that are always available, and the second comprising attributes that might have missing values. The diagnostic models were subjected to the uncertaintification procedure described in Section 3.2.

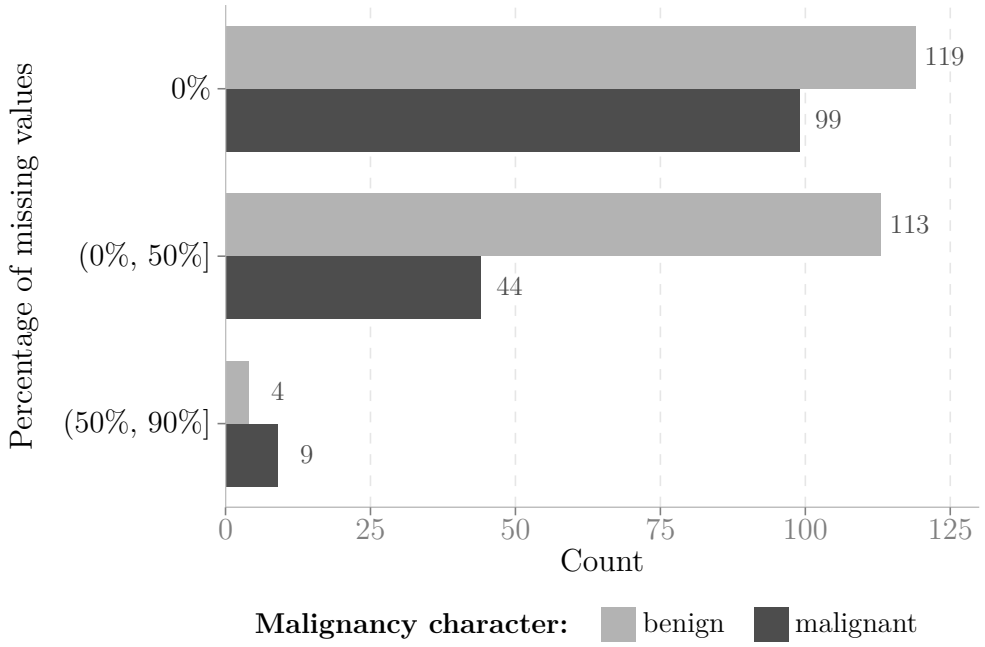

Figure 4.1: Distribution of patients in terms of percentage of missing values

Four groups of aggregation operators were selected for the aggregation step: weighted averages, OWA, integrals and t-operations (see Appendix A). Each group was evaluated in two scenarios, i.e. whether the subject of aggregation was whole intervals or numerical representatives of intervals. This step was described in Section 3.3.

Outputs of the aggregation step were thresholded using the methods listed in Section 3.4. The value of 0.5 served as a raw thresholding point of classification as malignant or benign tumour. The resulting intervals or numerical values were checked to determine whether they were greater or lower than $0.5 \pm \epsilon$, where $\epsilon \geq 0$. For a resulting interval one can distinguish three intervals associated with a benign, unknown (`NA`) and malignant output, i.e. $[0, 0.5 - \epsilon]$, $[0.5 - \epsilon, 0.5 + \epsilon]$ and $[0.5 + \epsilon, 1]$ respectively. One can check which of these intervals has the largest common part with the input interval or whether the intersected region is greater than the sum of the remaining two intervals. The thresholding strategies used are described in Appendix B.

Attribute	Diagnostic model					
	SM g_1[7]	Alc. g_2[6]	LR1 g_3[10]	LR2 g_4[10]	Tim. g_5[54]	RMI g_6[55]
age	-	-	✓	✓	-	✓
menopausal status	✓	-	-	-	✓	✓
pain during examination	-	-	✓	-	-	-
hormonal therapy	-	-	✓	-	-	-
hysterectomy	-	-	-	-	-	✓
ovarian cancer in family	-	-	✓	-	-	-
lesion volume	✓	-	✓	-	-	-
internal cyst walls	✓	-	✓	✓	-	-
septum thickness	✓	-	-	-	-	-
echogenicity	✓	✓	-	-	-	-
localisation	✓	-	-	-	-	✓
ascites	✓	-	✓	✓	-	✓
papillary projections	-	✓	-	-	✓	-
solid element size	-	✓	✓	✓	-	✓
blood flow location	-	✓	✓	✓	-	-
resistance index	-	✓	-	-	-	-
acoustic shadow	-	-	✓	✓	-	-
amount of blood flow	-	-	✓	-	✓	-
CA-125 blood marker	-	-	-	-	✓	✓
lesion quality class	-	-	-	-	-	✓

Table 4.1: Attributes used by the selected preoperative diagnostic models. Features in the first group are always available; the second group may have missing values.

4.2 Assumptions on dataset partitioning

The evaluation procedure was based on the classic data division into training and test sets. Since the dataset varied in terms of levels of missing data, special steps were performed to split the data. For some levels of data missingness there were too few patients to perform a reliable division. This could lead to a situation where at some stage of training or testing there were discontinuities in the levels of missing data. Since the goal is to construct a classification procedure for all levels, a different approach was chosen.

The test set consisted of the patients with real missing data and some proportion of patients with a complete set of attributes. The training set was formed from patients with a complete set of features, and the missing data were simulated. It is impossible to reconstruct the actual process by which missing data occur during examination; the simulations

therefore assumed random data missingness. In addition, the true distribution of levels of data missingness is also unknown, so in the training phase different levels of missing data were simulated uniformly. Given these steps, both training and test sets had none of the aforementioned discontinuities in the levels of missing data.

Moreover, the true distribution of tumour malignancy in the population is also unknown. According to a recent review of classification procedures, the distributions vary widely among study groups [23]. Therefore, in this evaluation an equal distribution of malignancy was assumed. In the repeated random sampling of patients and obscuring of data, the same proportions of benign and malignant cases were selected.

4.3 Evaluation procedure

The training set consisted of 200 patients with no missing data. The test set consisted of 175 patients: the remaining 18 patients with no missing data, together with those who had missing values in the range $(0\%, 50\%]$. The aforementioned subgroups of 200 and 18 patients had the same distribution of tumour malignancy. Patients with more than 50% of values missing were excluded from the study. The partition of the datasets is depicted in Figures 4.2 and 4.3.

In the training phase we select the parameters of the aggregation operators and thresholding strategies. The levels of missing data in the simulation step vary from 0% to 50% with a step size of 5%. For each level, 1000 repetitions were made of the following procedure:

1. randomly select from the training set 75 patients with benign tumours and 75 patients with malignant tumours,
2. obscure (remove) a given percentage (level) of patients' features,
3. calculate interval-valued diagnoses with uncertaintified diagnostic models,
4. calculate final diagnosis with aggregation strategies.

All results were averaged over the repetitions and the levels of missing data. The numerical parameters of the aggregation strategies were optimised on a reasonable set of values, selected by an expert. All steps of the training phase are depicted in Figure 4.4. The result of this phase

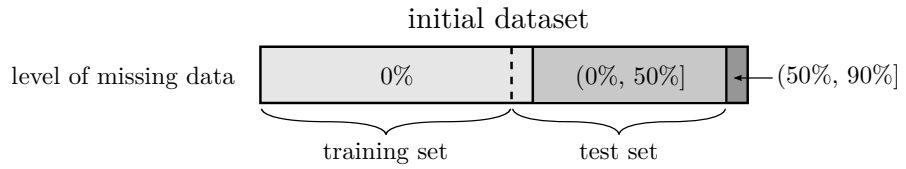

Figure 4.2: The division of the medical dataset. Patients with more than 50% missing values were not included in the experiment.

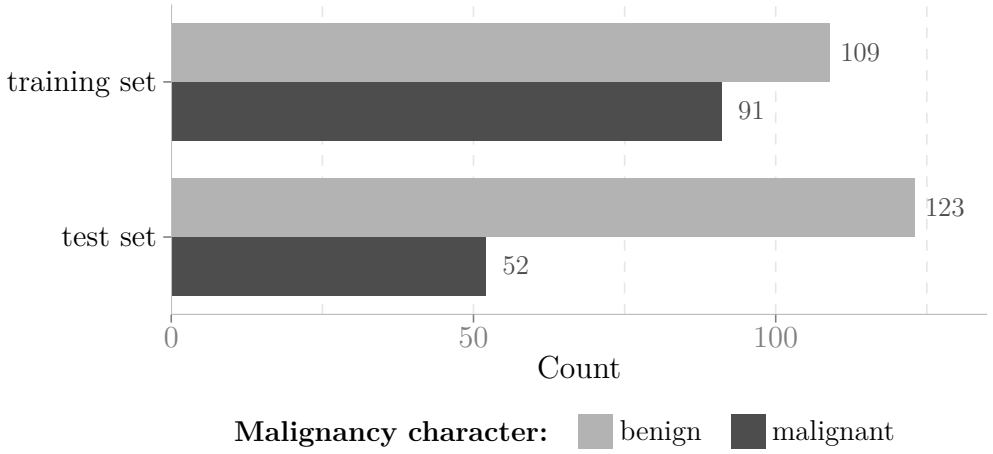

Figure 4.3: Class distribution in the medical training and test sets

is a set of optimised aggregation strategies which performs well on the simulated missing data.

In the test phase the selected aggregation strategies are checked on the dataset with actual missing values. A stratified bootstrapping with 500 replications is used to estimate the uncertainty of the performance [56].

4.4 Criteria of performance evaluation

The evaluation procedure aims to identify an aggregation strategy that provides accurate diagnosis with the highest possible decisiveness. In the given medical problem, the aggregation strategy should ensure both very high sensitivity and specificity. In some cases the diagnostic models may result in ambiguous decisions, hence the aggregation strategy should not perform a classification by chance; in such a case the patient should be referred to an experienced gynaecologist. A few percent of patients having no recommendation for diagnosis is an acceptable situation.

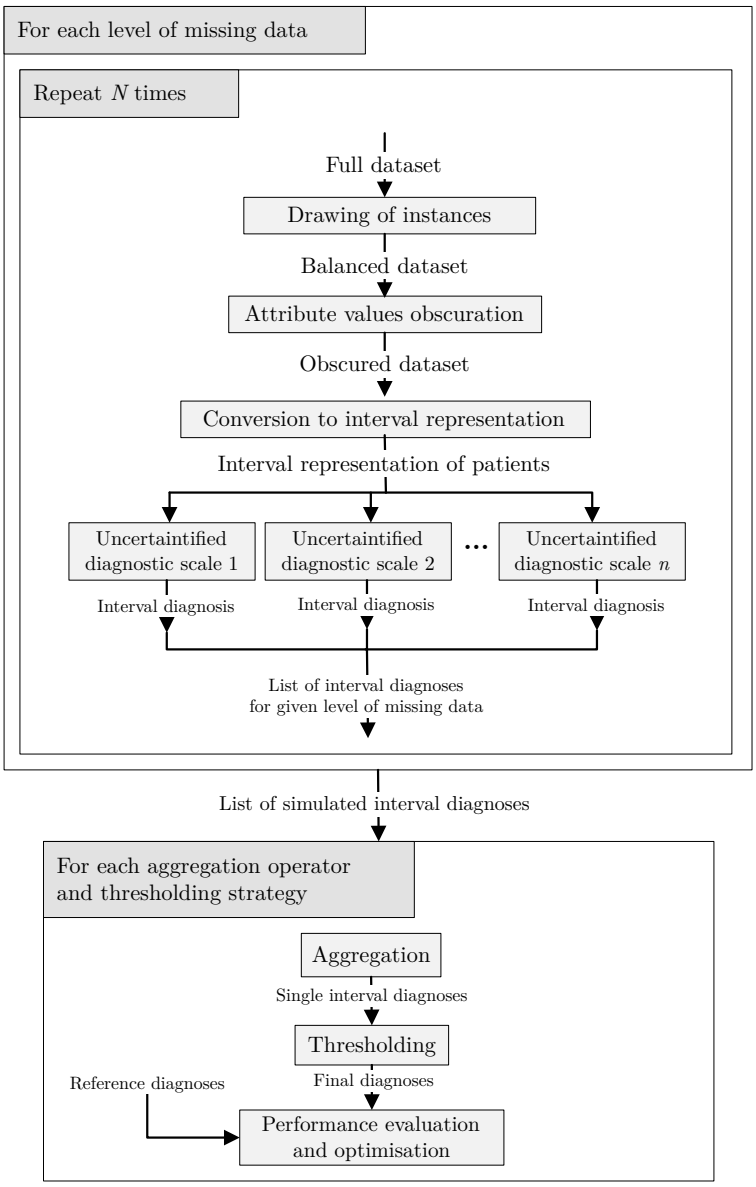

Figure 4.4: Visualisation of the training phase. Data flow is represented by arrows, and boxes represent operations on data.

		Predicted		
		benign	malignant	NA
Actual	benign	0	2.5	1
	malignant	5	0	2

Table 4.2: Cost matrix. The costs were assigned based on expert gynaecologists' opinions.

In general it is difficult to select an appropriate performance measure that unifies certain other measures [42]. Hence, the cost matrix method can be considered in this problem. An advantage of this method is that it has a good interpretation in medical terms.

Table 4.2 presents the costs associated with possible decisions made by a classifier. The correct classification of tumours, i.e. *true positives* and *true negatives*, comes with zero cost. The highest cost is associated with *false negatives*, when a patient has a malignant tumour and the prediction indicates that it is benign. The cost of a *false positive* is two times smaller than that of a *false negative*, since unnecessary surgery is still dangerous for a patient, but there is a much greater chance of recovery. There is also a certain difference in costs when a classifier does not know which class should be assigned. The cost of no decision (`NA`) is lower than that of a *false positive*, since the patient is referred to an experienced gynaecologist who is still able to make a good decision. However, a further misclassification is not ruled out, so in case of no prediction, the cost when the tumour is malignant is two times greater than the cost when it is benign. Here, a theoretical maximal total cost is equal to 567.5 (if all cases are misclassified).

4.5 Technical issues

The statistical evaluation, as well as the implementation of the proposed methodology, were performed using R software, version 3.1.2 [57]. All scripts, documentation and non-sensitive data are available on the GitHub repository[1]. All computations were performed with the use of the Microsoft Azure cloud service.

[1] `https://github.com/ovaexpert/ovarian-tumor-aggregation`

4.6 Results

In the training phase eight groups of aggregation operators and four groups of thresholding strategies were checked and optimised in order to minimise the total cost obtained according to the cost matrix. This resulted in a set of aggregation strategies with optimised parameters, and the best within-group were selected. For each group the top three aggregation strategies are listed in Table 4.3.

Figures 4.5–4.7 summarise the training phase with detailed levels of missing data. Although the original and uncertaintified classifiers are not subject of the optimisation, they are plotted for comparison with the aggregation strategies. Firstly, the original diagnostic models were run on the input data. Note that the original models may still classify patients with missing values, since a particular model may not use the features for which values are missing. As one can see in Figure 4.5, the total cost grows rapidly with an increasing level of missing data. This is caused by the fact that the original models are not able to make a diagnosis if any of the used attributes are not available; thus they fail to predict and produce no diagnosis (`NA`).

Secondly, Figure 4.6 shows how the diagnostic models perform if they are uncertaintified. In this case the cost grows more slowly. This illustrates that even the self-contained process of uncertaintification reduces the impact of missing data on the effectiveness of classification.

Thirdly, Figure 4.7 depicts the costs of diagnosis for aggregation strategies. The diagram shows one arbitrarily chosen aggregation strategy for a given group. The total costs are smaller than in the cases shown in Figure 4.5 and Figure 4.6 for each level of missing data, and their growths are also small. To sum up the training phase, this step allows us to select a group of aggregation strategies that perform better than single diagnostic models for each level of missing data.

No.	Operator parameters	Performance measure with 95% CI Total cost	Dec	Sen	Spe
	Integrals in interval mode given by Formulae (A.8) and (A.9)				
1	Choquet, μ_{AUC}, $\hat{\tau}_{\text{mp},0.025}$	80.0 (±28.8)	92.0 (±4.0)	82.6 (±11.0)	93.0 (±4.6)
2	Choquet, μ_{card}, $\hat{\tau}_{\text{mp},0.025}$	80.0 (±27.8)	92.0 (±4.1)	84.8 (±10.3)	91.3 (±5.3)
3	Sugeno, μ_{card}, $\hat{\tau}_{\text{mp},0.025}$	80.0 (±26.1)	87.4 (±4.9)	90.9 (±8.0)	89.0 (±6.2)
	Integrals in numerical mode given by Formulae (A.3) and (A.4)				
4	Choquet, REP_{mp}, μ_{AUC}, $\tau_{0.025}$	80.0 (±28.8)	92.0 (±4.0)	82.6 (±11.0)	93.0 (±4.6)
5	Choquet, REP_{mp}, μ_{card}, $\tau_{0.025}$	80.0 (±27.8)	92.0 (±4.1)	84.8 (±10.3)	91.3 (±5.3)
6	Sugeno, REP_{min}, μ_{card}, $\tau_{0.0}$	87.5 (±31.8)	100.0 (–)	86.5 (±8.6)	82.9 (±6.9)
	Weighted means in interval mode given by Formula (A.6)				
7	ω_{wid}, $r = 2$, $\hat{\tau}_{\text{mp},0.025}$	75.5 (±26.2)	97.1 (±2.6)	91.8 (±7.2)	84.3 (±6.2)
8	ω_{mp}, $r = 3$, $\hat{\tau}_{\text{mp},0.0}$	77.5 (±28.1)	100.0 (–)	88.5 (±8.7)	84.6 (±6.3)
9	ω_1, $r = 2$, $\hat{\tau}_{\text{mp},0.0}$	79.0 (±27.5)	94.3 (±3.2)	91.7 (±7.3)	84.6 (±6.4)
	Weighted means in numerical mode given by Formula (A.1)				
10	REP_{min}, ω_{ep}, $r = 3$, $\tau_{0.0}$	72.0 (±27.5)	97.1 (±2.6)	90.0 (±8.6)	86.7 (±5.9)
11	REP_{mp}, ω_{ep}, $r = 3$, $\tau_{0.0}$	74.5 (±27.9)	97.1 (±2.6)	90.0 (±8.6)	85.8 (±5.9)
12	REP_{min}, ω_{wid}, $r = 3$, $\tau_{0.025}$	78.0 (±30.0)	94.3 (±3.4)	85.7 (±9.3)	89.7 (±5.5)
	Ordered Weighted Average (OWA) operators in interval mode given by Formula (A.7)				
13	ω_{dec}, π_{mp}, $\hat{\tau}_{\text{mp},0.025}$	70.0 (±29.1)	94.9 (±3.4)	90.2 (±8.3)	87.8 (±5.9)
14	ω_{dec}, π_{min}, $\hat{\tau}_{\text{mp},0.025}$	72.0 (±29.3)	96.6 (±2.8)	90.2 (±8.3)	86.4 (±5.9)
15	ω_{dec}, π_{wm}, $\hat{\tau}_{\text{mp},0.025}$	73.5 (±28.4)	94.9 (±3.1)	90.0 (±8.5)	87.1 (±6.2)
	Ordered Weighted Average (OWA) operators in numerical mode given by Formula (A.2)				
16	REP_{mp}, ω_{dec}, π_{mp}, $\tau_{0.025}$	70.0 (±29.1)	94.9 (±3.4)	90.2 (±8.3)	87.8 (±5.9)
17	REP_{mp}, ω_{dec}, π_{min}, $\tau_{0.025}$	72.0 (±29.3)	96.6 (±2.8)	90.2 (±8.3)	86.4 (±5.9)
18	REP_{mp}, ω_{dec}, π_{wm}, $\tau_{0.025}$	73.5 (±28.4)	94.9 (±3.1)	90.0 (±8.5)	87.1 (±6.2)
	t-operation based operators in interval mode given by Formula (A.10)				
19	s_{max}, $\alpha = 0.25$, $\hat{\tau}_{\text{mp},0.025}$	78.0 (±26.6)	94.3 (±3.4)	91.8 (±7.0)	84.5 (±6.6)
20	t_{min}, $\alpha = 0.25$, $\hat{\tau}_{\text{max},0.025}$	89.5 (±28.5)	94.9 (±3.1)	89.8 (±8.8)	82.1 (±6.9)
21	t_{min}, $\alpha = 1.0$, $\hat{\tau}_{\text{max},0.0}$	100.0 (±35.0)	100.0 (–)	73.1 (±12.5)	90.2 (±5.2)
	t-operation based operators in numerical mode given by Formula (A.5)				
22	REP_{mp}, s_{max}, $\alpha = 0.25$, $\tau_{0.025}$	82.0 (±27.5)	94.9 (±2.8)	89.8 (±8.4)	84.6 (±6.4)
23	REP_{max}, t_{min}, $\alpha = 0.25$, $\tau_{0.025}$	89.5 (±28.5)	94.9 (±3.1)	89.8 (±8.8)	82.1 (±6.9)
24	REP_{min}, s_{prod}, $\alpha = 0.25$, $\tau_{0.025}$	95.0 (±29.7)	93.7 (±3.2)	87.5 (±8.9)	82.8 (±7.2)

Table 4.3: Performance measures for the top three aggregation operators and thresholding strategies within each group. All measures, along with bootstrap percentile 95% confidence intervals, are obtained in the test set. The decisiveness, sensitivity and specificity are in percentage values.

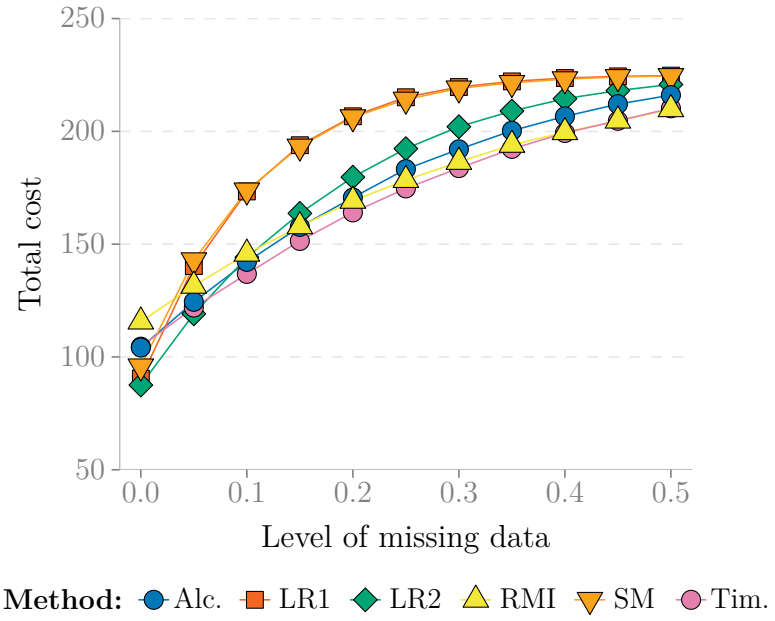

Figure 4.5: Simulation results for the original models

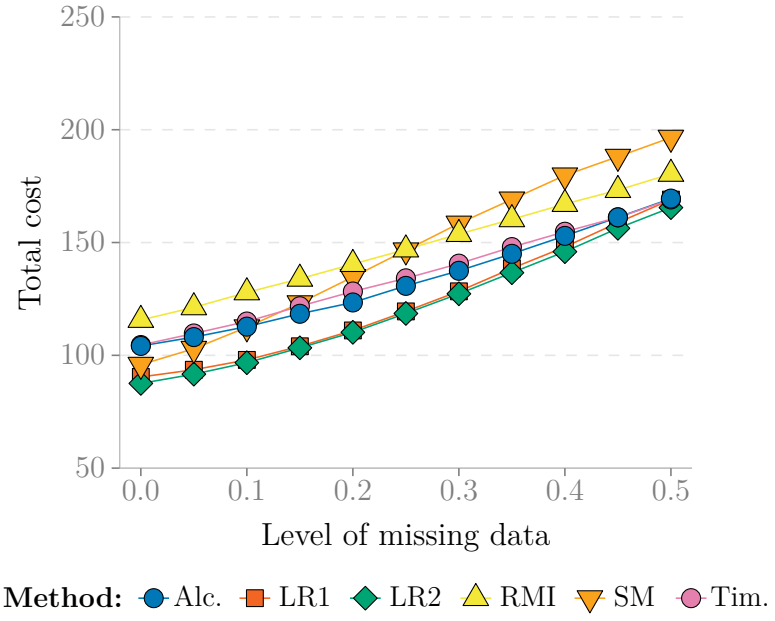

Figure 4.6: Simulation results for uncertaintified models

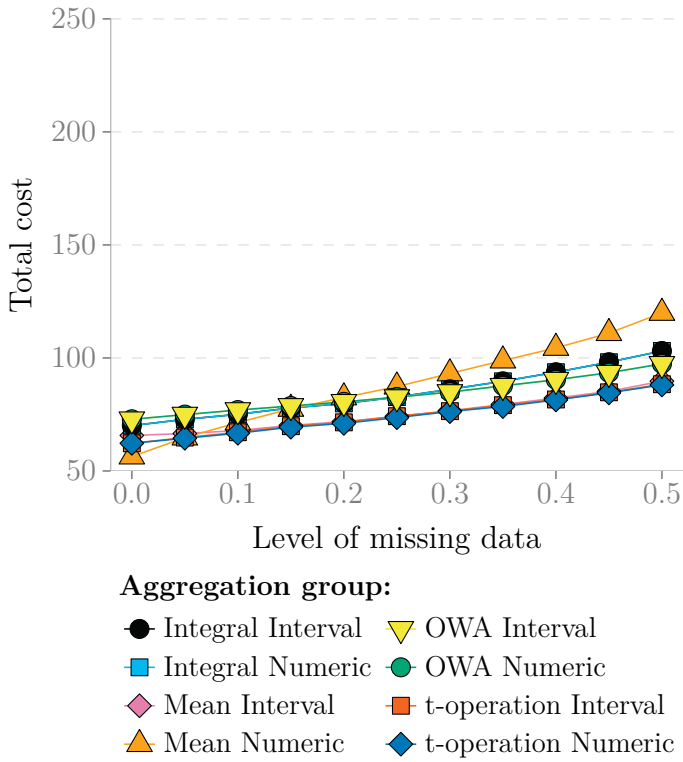

Figure 4.7: Simulation results for aggregation groups. The aggregation groups show strategies with the lowest total cost achieved on the training set.

Next, the results are verified on the test set with real missing values. Figures 4.8–4.10 show the total costs for original models, uncertaintified models, and aggregation strategies, respectively. Again, the cost is highest for the original models and lowest for aggregation strategies. These results confirm the claim that the aggregation strategies are a good tool for supporting diagnosis in the presence of missing data.

An additional exploratory analysis of the results can be made based on Figures 4.11–4.13. Accuracy, sensitivity, specificity and decisiveness are presented for each method. The accuracy, sensitivity and specificity of the original models are quite high, but the decisiveness does not reach an acceptable level. The process of uncertaintification improves the decisiveness. Finally, the aggregation operators produce higher values for the performance measures, while a diagnosis is unavailable for fewer than 10% of the patients. These are very good results, showing that uncertaintification, aggregation and thresholding constitute a promising method of improving the quality of medical diagnosis.

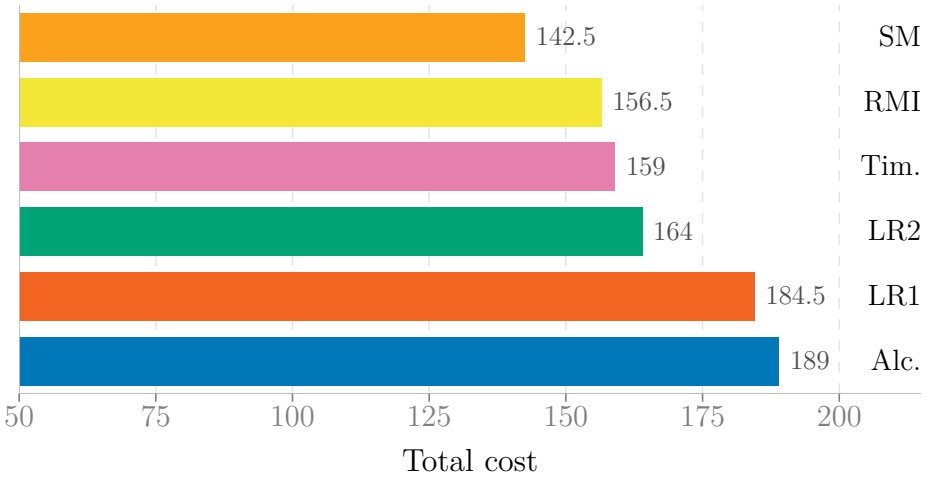

Figure 4.8: Total cost performance on the test set among the original models

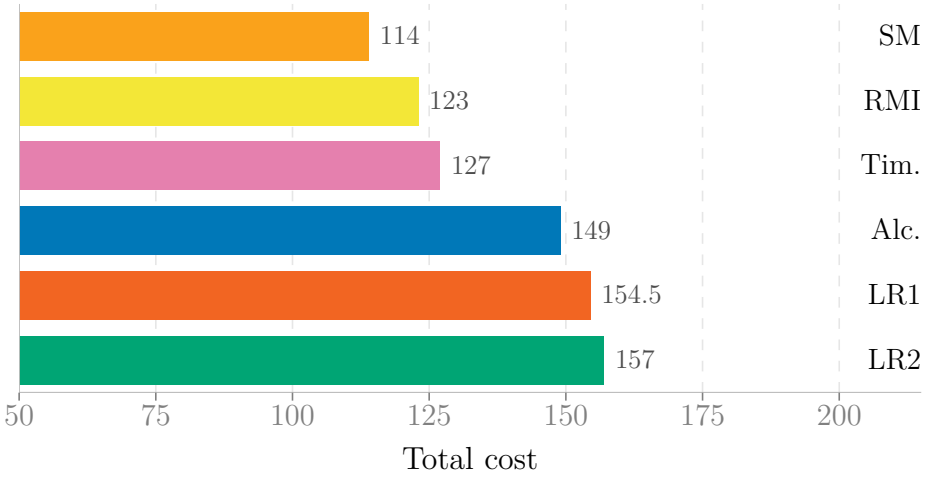

Figure 4.9: Total cost performance on the test set among uncertaintified models

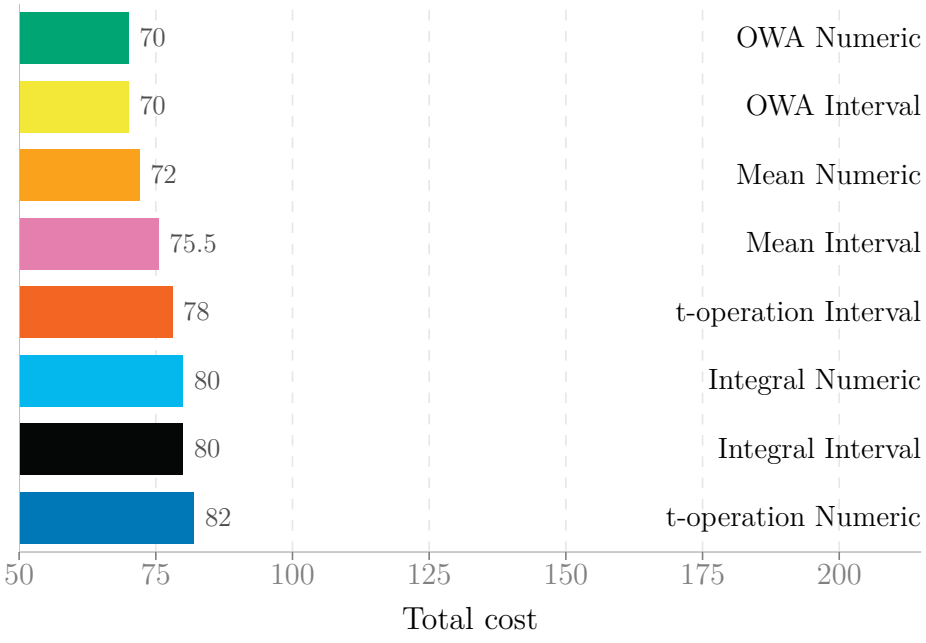

Figure 4.10: Total cost performance on the test set among aggregation groups by the lowest total cost

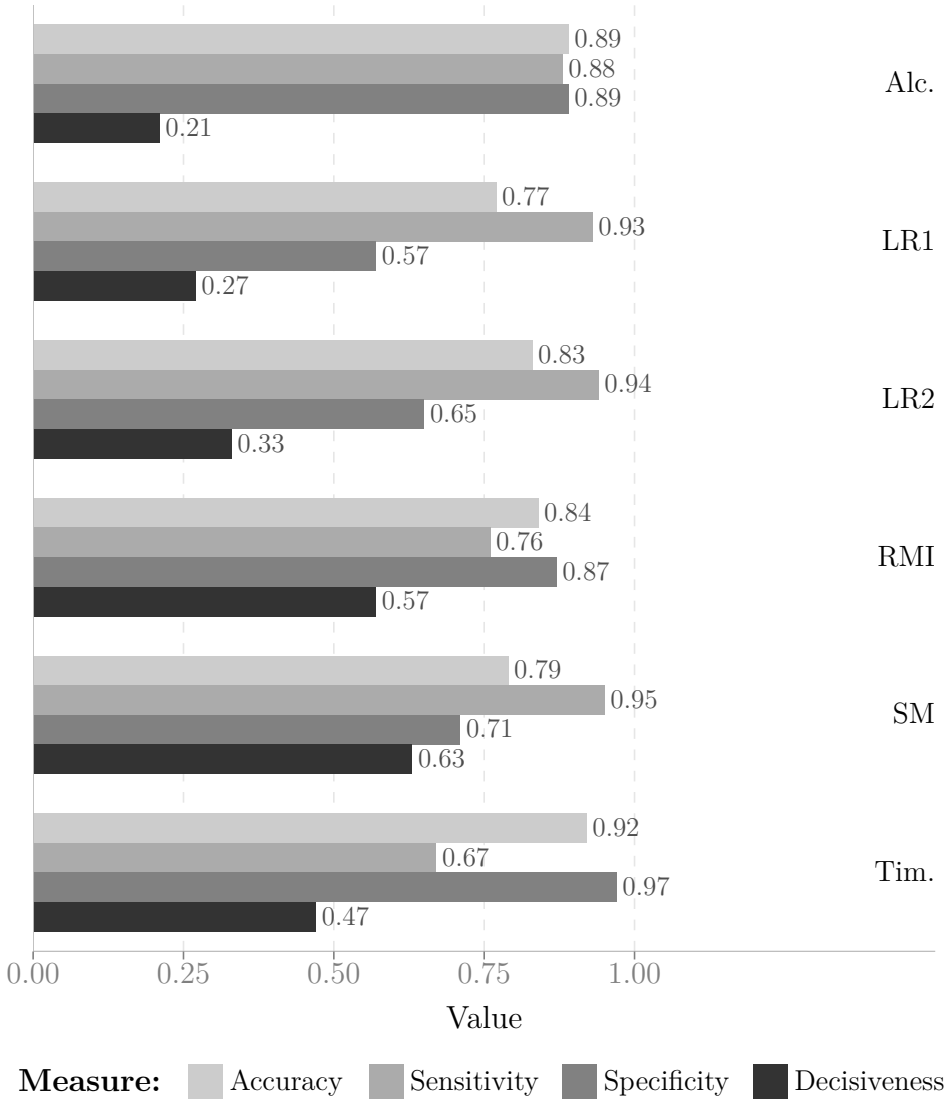

Figure 4.11: Performance measures on the test set among the original models

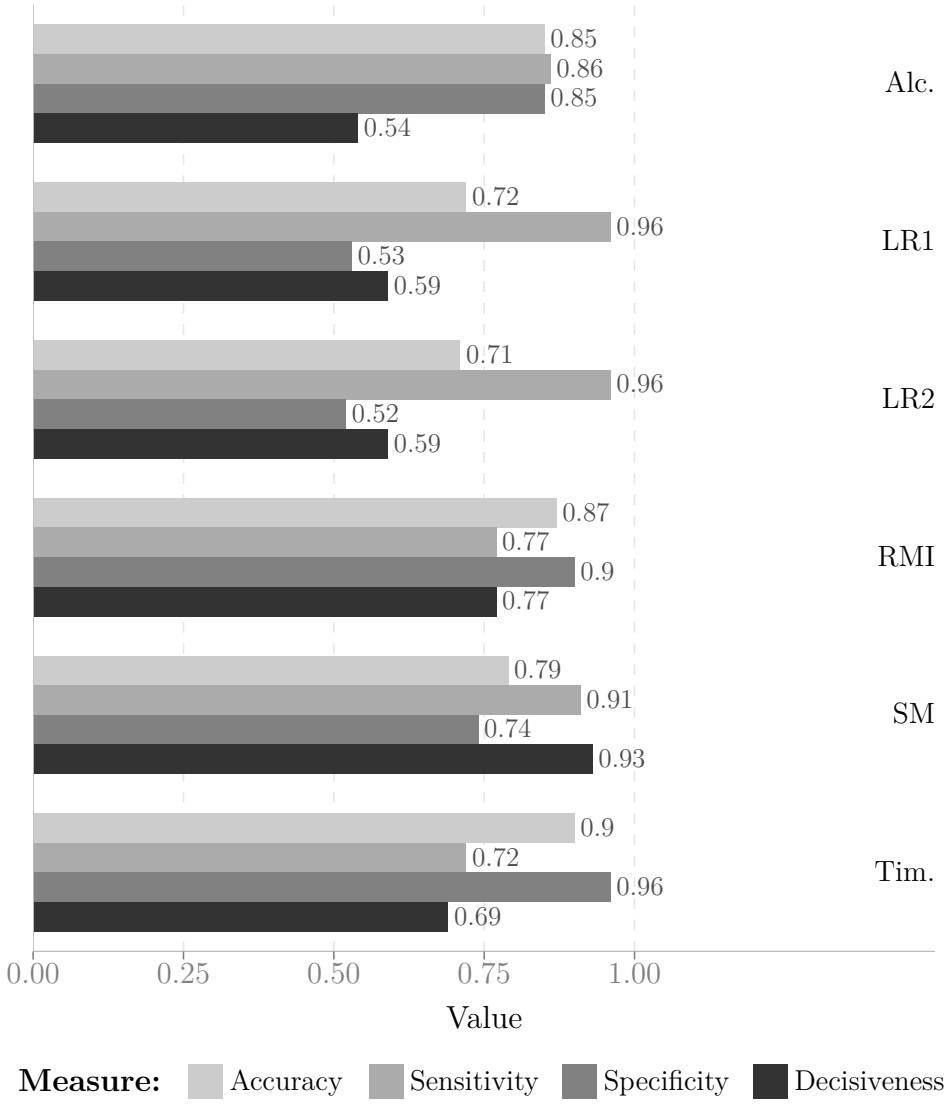

Figure 4.12: Performance measures on the test set among uncertaintified models

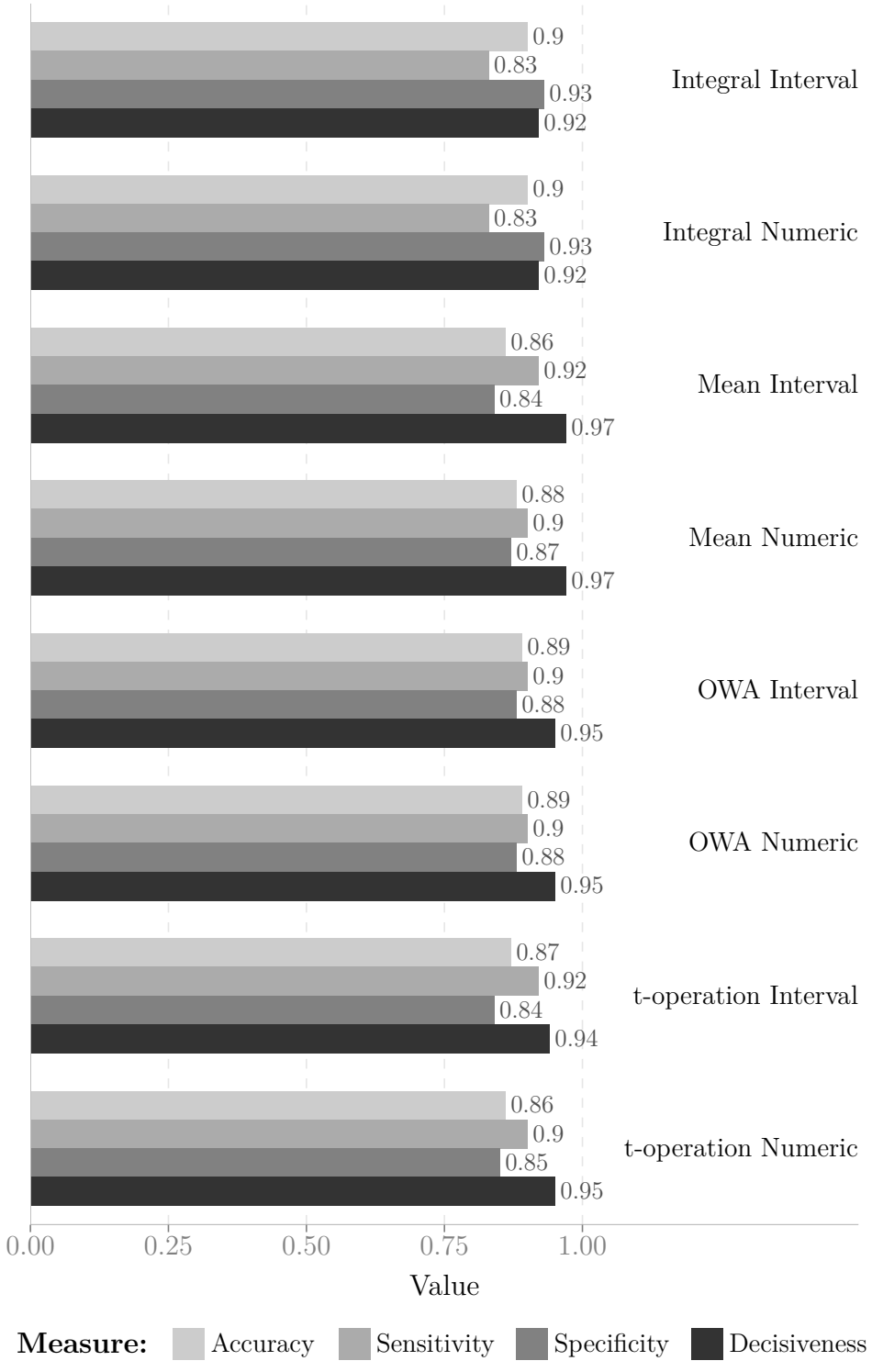

Figure 4.13: Performance measures on the test set among aggregation groups by the lowest total cost

4.7 Discussion and conclusions

The practical aim of the evaluation procedure was to incorporate an aggregation strategy as a new classification method in a real application for supporting ovarian tumour diagnosis. During the research, the OvaExpert system[1] was developed by an interdisciplinary team of scientists from Adam Mickiewicz University in Poznań and Poznan University of Medical Sciences. The system was implemented in order to gather medical data, as well as to support a physician in making classification decisions. In order to use the aggregation strategy as a diagnostic module, it is necessary to choose the best model. To select an aggregation strategy from those returned by the test phase, the following conditions must be satisfied:

- $\textsc{Sen} \geq 90\%$,
- $\textsc{Spe} \geq 80\%$,
- $\textsc{Sen} > \textsc{Spe}$,
- $\textsc{Dec} < 100\%$.

The first two conditions narrow aggregation strategies to those having both high sensitivity and specificity. The third condition reflects the fact that in this medical case sensitivity is more important than specificity. Since these performance measures are correlated, some aggregation operators might trade off sensitivity for specificity – such models should be discarded. Finally, the last condition filters out models that recommend diagnoses without sufficient justification. In such cases no decision, leading to further examinations, is better than a wrong decision.

With this set of conditions, the chosen aggregation strategy is an OWA operator defined by Formula (A.2) with the weighting vector ω_{dec}, REP_{mp} as representative selector, $\tau_{0.025}$ as threshold and $\pi_{\min}$ used to order input values. This aggregation strategy will be further referred to as OEA. A comparison of the total cost of OEA with that of the original diagnostic models is shown in Figures 4.14 and 4.15. OEA is significantly better than all of the other diagnostic models. This was verified with McNemar's test; the results are given in Table 4.4.

[1] http://ovaexpert.pl/en

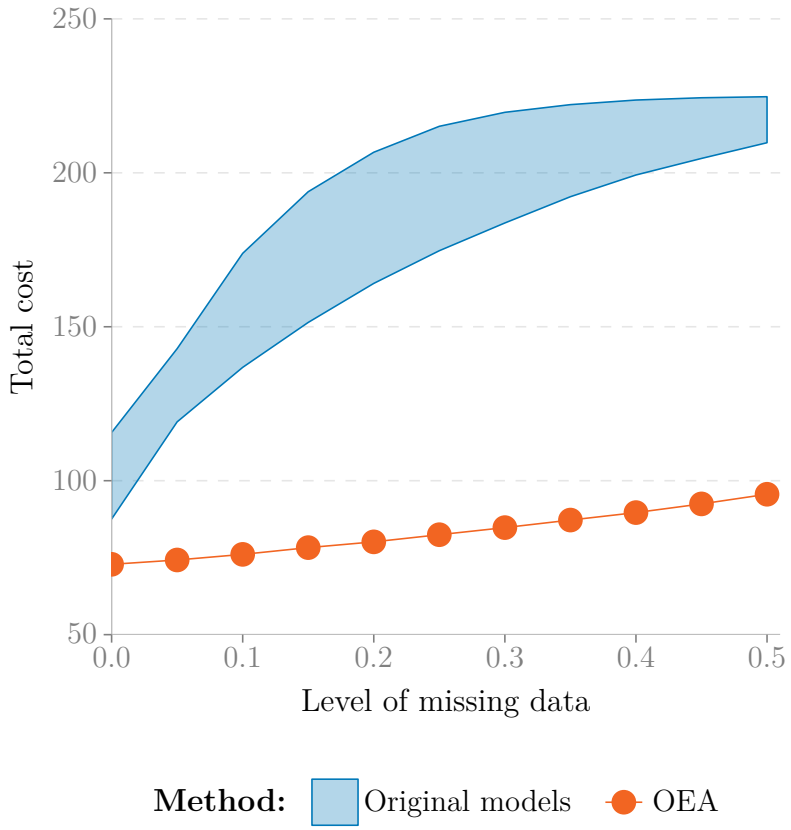

Figure 4.14: Comparison of total costs between the original diagnostic models and the selected aggregation strategy in the training phase. The shaded area indicates lower and upper bounds of the total cost of the original models.

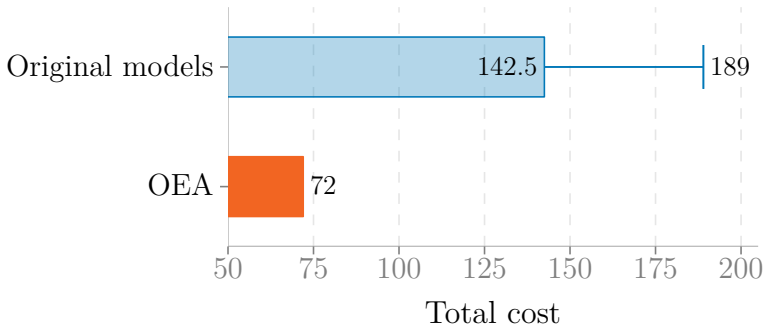

Figure 4.15: Comparison of total costs between the original diagnostic models and the selected aggregation strategy in the test phase. The whiskers on the first bar indicates lower and upper bounds of the total cost of all original models.

		Original model					
		Alc.	LR1	LR2	RMI	SM	Tim.
Uncertaintified model	Alc.	< 0.001	< 0.001	< 0.001	0.834	0.472	0.723
	LR1	< 0.001	< 0.001	< 0.001	0.406	0.080	1.000
	LR2	< 0.001	< 0.001	< 0.001	0.366	0.060	0.935
	RMI	< 0.001	< 0.001	< 0.001	< 0.001	0.001	< 0.001
	SM	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
	Tim.	< 0.001	< 0.001	< 0.001	0.001	0.017	< 0.001
	OEA	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Table 4.4: McNemar's test with Benjamini–Hochberg correction between the original diagnostic models and the uncertaintified models with the selected aggregation strategy. It can be observed that the uncertaintified models significantly outperform the corresponding original models. Moreover, OEA significantly outperforms the original models ($\alpha = 0.05$).

The main conclusion of the experiment is that with the proposed approach one can obtain better performance in the classification of ovarian tumours by the extensive use of known diagnostic models. This is especially evident when diagnosis is based on incomplete data. The aggregation of the diagnostic models exploits the synergy effect and allows one to deal with fairly large quantities of missing data, up to 50%. The selected method, OEA, is able to give proper diagnoses despite missing data. The total cost of 72 is very low compared with the original diagnostic models (142.5–189). In addition, both high sensitivity and specificity are good indicators for applicability in medical practice. The results presented here are a part of the project related to the OvaExpert system. The results obtained for the aggregation strategy will be further generalised when the patient dataset is sufficiently enlarged.

5 Evaluation on UCI datasets

In the previous chapter I showed how the proposed approach can be successfully applied in the problem of supporting ovarian tumour diagnosis. However, there are three issues of concern regarding the presented solution. Firstly, due to the sensitivity of medical data, the described process is only partially reproducible. Secondly, the use of imputation methods was rejected in the project setup, but in general it would be desirable to compare possible solutions. Thirdly, the application considers a particular medical problem, and it might be interesting to check how the proposed approach performs on datasets from different domains. In this chapter I present a fully reproducible application of the proposed approach in various classification problems and compare the results against imputation as an alternative method.

5.1 Subject of evaluation

The study concerned five datasets from the University of California, Irvine (UCI) Machine Learning Repository [58]:

1. *bank-marketing* – the data are related to direct marketing campaigns (phone calls) of a Portuguese banking institution; the classification goal is to predict whether the customer will subscribe a term deposit [59];
2. *census-income* – the goal is to predict whether income exceeds \$50 000 per year, based on United States Census data from 1994 [60];
3. *credit-card* – the aim of the classification is to predict whether customers in Taiwan will have default payments next month [61];

4. *magic* – the goal is to predict whether data from a Cherenkov gamma-ray telescope are gamma (signal) or hadron (background) [62];
5. *wine-quality* – the goal is to assess the quality (good or bad) of red and white wine from the north of Portugal, based on physicochemical tests of wine samples [44].

The aforementioned data have no missing values, have numeric as well as categorical features, and were preprocessed and saved as R objects. All of the preprocessing procedures and generated datasets are publicly available [45].

The five basic classification methods were the following [63]:

1. generalised linear models (*glm*);
2. neural networks (*nnet*);
3. a support vector machines linear model (*svmLinear*);
4. classification trees (*rpart*);
5. the k-nearest neighbours algorithm (*knn*).

The one-rule classifier (*OneR*) [64] was used as a baseline model. All classifiers had a threshold value set to 0.5. The classifiers were built with use of R *caret* library [65].

The aggregation strategies were selected as in Chapter 4, with a preference for those not producing `NA`.

Three imputation methods were chosen for the experiment:

1. median/modal value;
2. random forest [66];
3. multivariate imputation by chained equations (*mice*) [67].

5.2 Assumptions on dataset partitioning

The instances from the datasets were sampled so that the distribution of binary classes was equal (50%/50%). This in particular eliminates the problem of class imbalance and influences on the meaning of performance measures such as sensitivity or specificity.

The split of the dataset D can be less formally expressed in the following form:

$$D = (D^{\text{fs}} + D^{\text{cl}}) \cdot a + D^{\text{ob}}\,,$$

where:

- D^{fs} is a dataset with 150 instances for feature selection;
- D^{cl} is a dataset with 450 instances for classification;
- a is the number of classifiers (here, 5 excluding *OneR*);
- D^{ob} is a dataset with 1000 instances for obscuration.

In this configuration, the datasets have in sum 4000 instances each. This also serves to reflect the situation described in Chapter 4, where a few classifiers were previously constructed independently and then had to classify a new dataset.

5.3 Evaluation procedure

The following steps of the experiment for each dataset are listed in Algorithm 5.1. Firstly, each classifier selects features through random forests (except *cart*, which has its own internal method) and learns on a classification dataset.

Secondly, since the datasets have no missing values, the obscured sub-dataset D^{un} must be generated. In this step 1/3 of instances remain unchanged and the remaining 2/3 are randomly and uniformly obscured. The obscuration process inserts `NA` values into attributes used by the classifiers. This step is performed so that an interval classifier will always have at least one attribute available – this serves to prevent the generation of unit interval predictions. In Chapter 4 the observed obscuration level

Algorithm 5.1: Evaluation procedure for UCI datasets

For each dataset D_i:
  For each classifier g_j:
    Select features g_j on $D_{i,j}^{\text{fs}}$.
    Learn g_j on $D_{i,j}^{\text{cl}}$.
    $\hat{g}_j :=$ uncertaintified g_j.
  $D_i^{\text{un}} :=$ randomly obscured dataset D_i^{ob}.
  For each g_j and $\hat{g}_j$:
    Calculate performance measures of g_j and $\hat{g}_j$ on D_i^{un}.
  Select best imputation method Imp on D_i^{un}.
  Select best aggregation strategy AggStr on D_i^{un}.
  Compare performance of all g_j, $\hat{g}_j$, Imp, AggStr on D_i^{un}.

was limited to 50%. Here the obscuration affects from 1 to $n-1$ of the globally used attributes.

Finally, all classification approaches (classifiers, uncertaintified classifiers, imputation and aggregation strategies) are evaluated on the obscured dataset. All learning steps are carried out with the use of the nested 10-fold cross-validation procedure described in Algorithm 2.3.

5.3.1 Note on aggregation strategies learning

The key issue in choosing aggregation strategies is enlargement of the learning dataset by virtual simulation of missing data. If one has knowledge about missing data patterns, this might have a particularly good impact on future performance. However, usually the patterns cannot be restored. This might be arbitrarily simulated by, for example, a uniform obscuration. By *k-level data* we mean a subset of instances where each instance has exactly k missing values. In Algorithm 5.2 one can see how this step can be implemented. An example visualisation of the algorithm appears in Figure 5.1, where each block refers to k-level data. The algorithm makes extensive use of data that are complete, and hence has good potential for obscuration. Data with many missing values are used least often.

Algorithm 5.2: Simulating missing data patterns for aggregation strategies

```
1 n := # attributes used by the classifiers.
2 For k in [0, n − 1]:
3     Take 0-level data n − k times.
4     If k > 0:
5         For l in [1, k]:
6             Take l-level data 1 time.
7         For l in [0, k − 1]:
8             Change l-level data to k-level data.
```

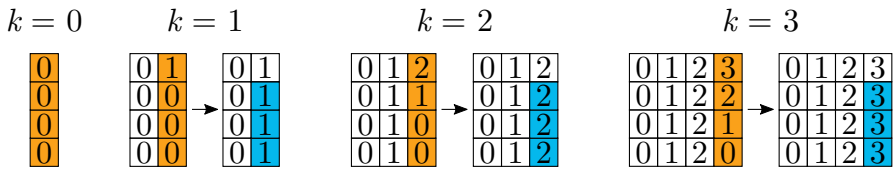

Figure 5.1: Visualisation of Algorithm 5.2 for $n = 4$. $\boxed{k}$ – instances with k missing values, ■ – new drawn instances in the constructed dataset, ■ – instances with changed level of data missingness.

5.4 Criteria of performance evaluation

In Chapter 4 the classification performance was evaluated by means of a cost matrix. Although that approach seems to be acceptable in the particular case, in this experiment a more general metric should be chosen.

For this reason, accuracy was selected as the main performance criterion for all datasets. When performance is estimated on data with missing values, the accuracy per level (ACC_i) is weighted by the decisiveness per level (DEC_i) and the sample size of instances with the given level of missing values ($|D_i|$). This performance metric is simply denoted as AD-SCORE (accuracy-decisiveness score):

$$\text{AD-SCORE} = \sum_{i=0}^{n-1} \text{ACC}_i \text{DEC}_i \frac{|D_i|}{|D|} .$$

5.5 Technical issues

The statistical evaluation, as well as the implementation of the proposed methodology, were performed using Microsoft R Open[1], version 3.3.1 [57]. All scripts, documentation and data are available on the GitHub repository[2]. All computations were performed with the use of computing infrastructure provided by the Faculty of Mathematics and Computer Science at Adam Mickiewicz University in Poznań. Since the experiment requires extensive use of various math libraries, benchmarking was performed in order to take advantage of the Basic Linear Algebra Subprograms libraries [68].

5.6 Results and discussion

This section summarises the results obtained for all datasets. Tables 5.1–5.3 and Figures 5.2–5.6 present the results obtained with original classifiers, interval classifiers, imputation and aggregation strategies on the *bank marketing* dataset. The plots and tables for the remaining datasets appear in Appendix D. The line plots with confidence intervals ($\alpha = 0.05$) were made with local polynomial regression fitting (LOESS) [69]. The results of statistical tests for AD-SCORE ($\alpha = 0.05$) appear in Table 5.4.

The imputation and aggregation strategies significantly outperformed single classifiers on each dataset ($p < 0.05$ for each test). The interval classifiers were significantly better than the original classifiers on only two datasets (*bank-marketing*: $p < 0.001$; *census-income*: $p < 0.035$). Finally, imputation significantly outperformed the aggregation strategies only on the *magic* dataset ($p < 0.001$).

The empirical results lead to interesting observations. It is worthwhile to perform either imputation or aggregation and thresholding in place of single classification. The process of uncertaintification is a step that may improve classification performance. Although aggregation strategies are as good as imputation, the choice of the latter may lead to slightly better performance. More importantly, in applications, the choice between the two should be determined by practical considerations because the justification of the predictions is different.

[1] `https://mran.microsoft.com`
[2] `https://github.com/andre-wojtowicz/agg-vs-imp`

Attribute	Classifier					
	OneR	*glm*	*nnet*	*svmLinear*	*rpart*	*knn*
day	-	-	✓	-	-	✓
education	-	✓	✓	-	-	-
job	-	-	✓	-	-	✓
marital	-	-	✓	✓	-	-
month	-	-	✓	✓	✓	✓
prev. days	-	✓	✓	-	-	✓
prev. days (bin.)	-	-	✓	-	-	✓
prev. outcome	-	✓	✓	✓	-	✓
age	-	-	✓	-	-	-
balance	✓	-	✓	-	-	✓
campaign	-	-	✓	✓	✓	✓
contact	-	✓	✓	✓	✓	✓
housing	-	✓	✓	✓	-	✓
loan	-	-	✓	-	-	✓
previous	-	✓	✓	✓	-	✓

Table 5.1: Predictors used by classifiers in the *bank marketing* dataset. Features in the first group are always available.

Classifier	Acc	Sen	Spe	p-value
OneR	0.524	0.533	0.514	0.045
glm	0.575	0.584	0.566	< 0.001
nnet	0.665	0.717	0.614	< 0.001
svmLinear	0.561	0.698	0.422	0.007
rpart	0.623	0.630	0.617	< 0.001
knn	0.635	0.667	0.604	< 0.001

Table 5.2: Performance of classifiers on the complete *bank marketing* dataset

Model	Group	Acc	Dec	Sen	Spe
glm	Original classifier	0.653	0.453	0.680	0.627
nnet		0.659	0.334	0.713	0.605
svmLinear		0.679	0.396	0.716	0.643
rpart		0.628	0.529	0.342	0.903
knn		0.643	0.350	0.672	0.612
glm	Uncertaintified classifier	0.678	0.625	0.659	0.696
nnet		0.658	0.587	0.733	0.585
svmLinear		0.667	0.712	0.790	0.540
rpart		0.621	0.688	0.329	0.900
knn		0.685	0.680	0.721	0.651
svmLinear & *mice*	Imputation	0.666	1.000	0.691	0.640
weighted mean, interval (A.6)	Aggregation strategy	0.630	1.000	0.664	0.593
ω_{mp}, $r = 0.5$, $\hat{\tau}_{\text{mp},0.0}$					

Table 5.3: Performance of classifiers on the obscured *bank marketing* dataset

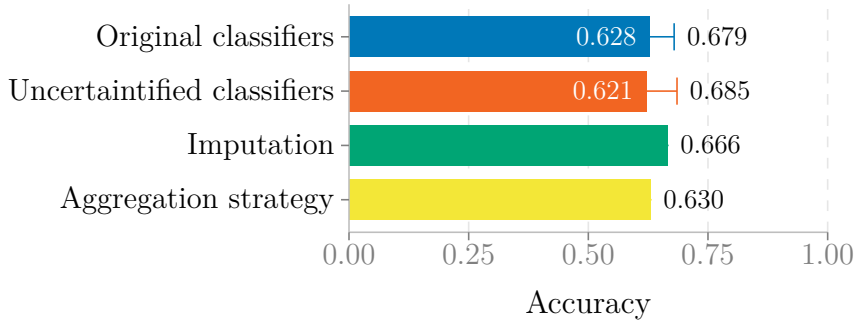

Figure 5.2: Accuracy of prediction models on the obscured *bank marketing* dataset. Whiskers indicate lower and upper bounds of accuracy of classifiers.

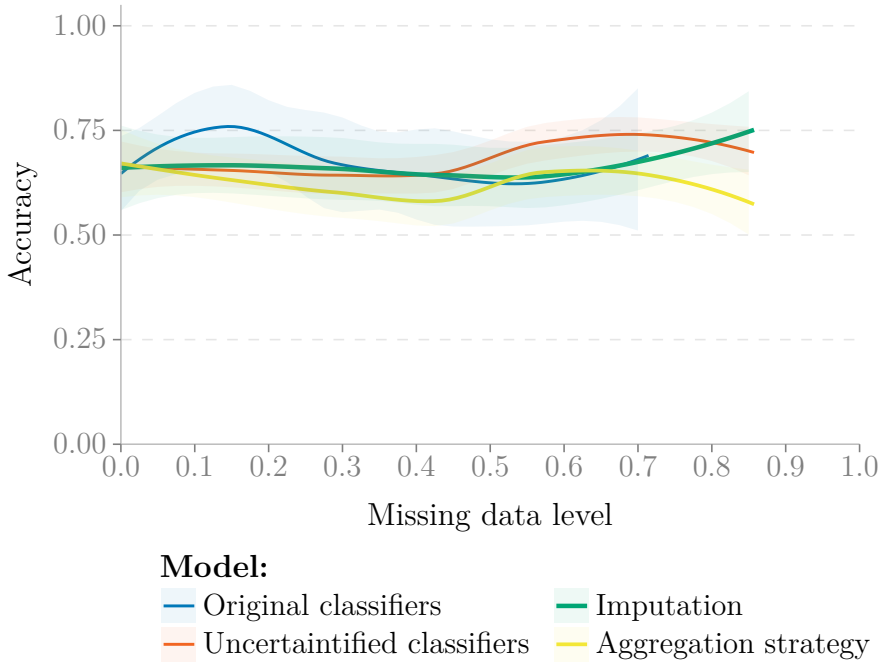

Figure 5.3: Accuracy of prediction models regarding missing data levels on the obscured *bank marketing* dataset. Shaded regions indicate 95% confidence interval bounds.

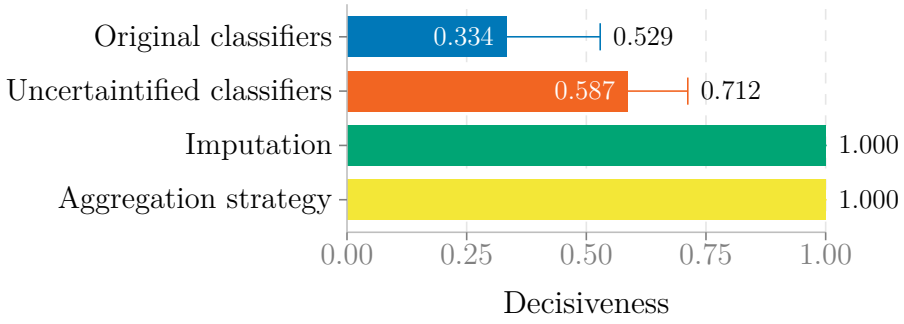

Figure 5.4: Decisiveness of prediction models on obscured *bank marketing* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

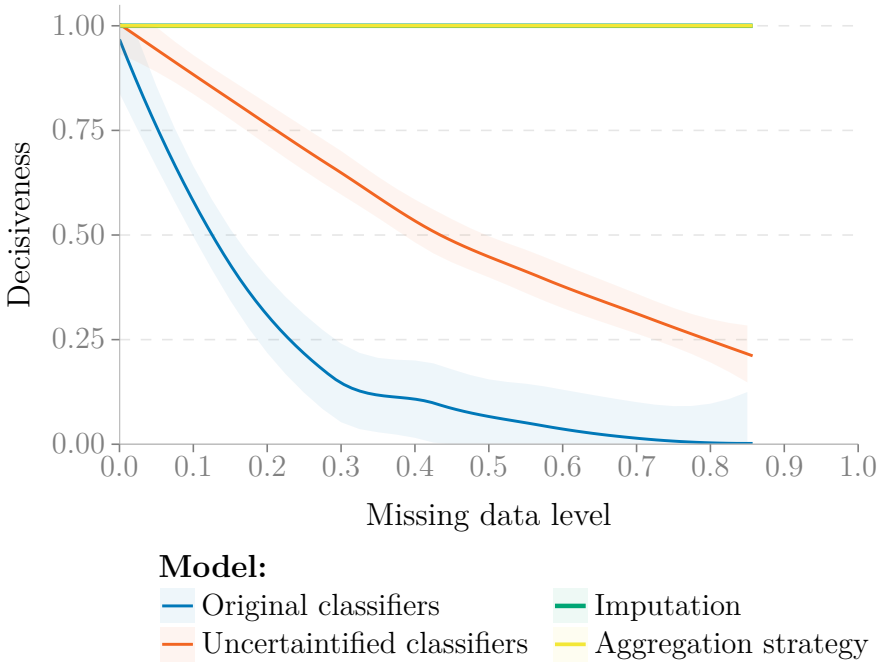

Figure 5.5: Decisiveness of prediction models regarding missing data levels on the obscured *bank marketing* dataset. Shaded regions indicate 95% confidence interval bounds.

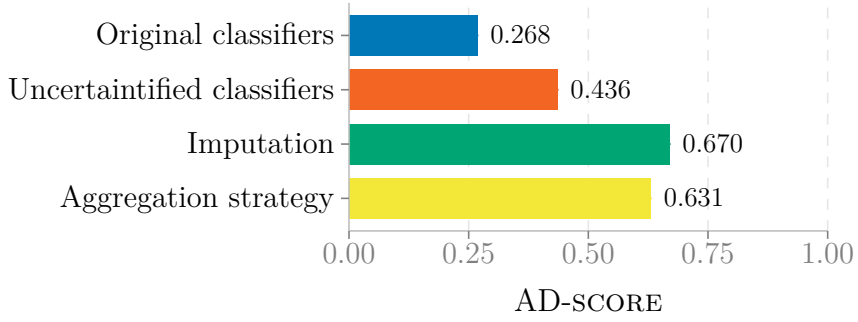

Figure 5.6: AD-score of prediction models on the obscured *bank marketing* dataset

	Original classifiers	Uncertaintified classifiers	Imputation	Aggregation strategy
Original classifiers	-	< 0.001	< 0.001	< 0.001
Uncertaintified classifiers	< 0.001	-	< 0.001	< 0.001
Imputation	< 0.001	< 0.001	-	0.132
Aggregation strategy	< 0.001	< 0.001	0.132	-

Table 5.4: Results of two-sided Student's t-test with Benjamini–Hochberg correction concerning whether by-obscurance-level-weighted means of AD-score differ on the obscured *bank marketing* dataset

An interesting observation can be made as the level of missing data increases. With the exception of the *magic* dataset, the accuracy of the models remains comparable. Figures 5.7 and 5.8 show that in the *magic* dataset the aggregation strategy has imbalanced sensitivity and specificity, which eventually impair the performance. It might be desirable to add a condition in the learning procedure to maintain proper balance between these two factors.

The difference in decisiveness is evident in the case of original and interval classifiers. This may lead to the conclusion that simple uncertaintification will be a sufficient step in some cases.

Lastly, this experiment considered all possible levels of data missingness. However, in practical situations (for example, in that described in Chapter 4) it might be reasonable to restrict prediction to cases with up to, for instance, 50% of missing values. Figure 5.9 shows that on the *wine-quality* dataset this may change the prediction performance for the imputation and aggregation strategies. This restriction may be reasonable, since the less data is present, the wider intervals are produced by the uncertaintified classifiers. In consequence, the aggregation strategies are unable to make good decisions in such situations.

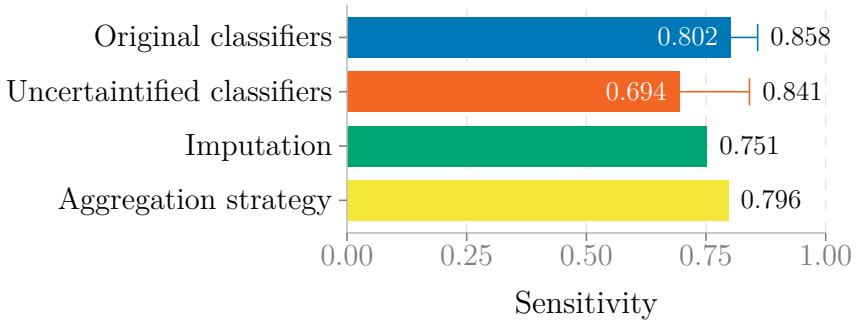

Figure 5.7: Sensitivity of prediction models on the obscured *magic* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

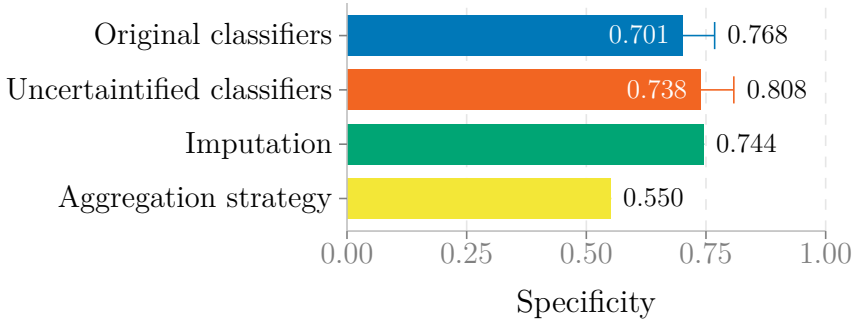

Figure 5.8: Specificity of prediction models on obscured *magic* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

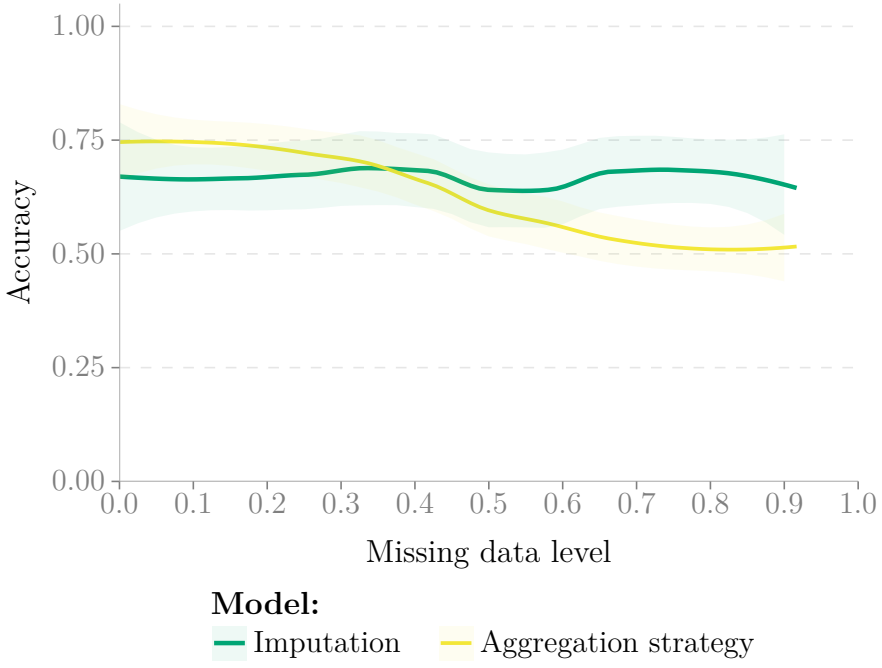

Figure 5.9: Accuracy of imputation and aggregation strategy regarding missing data levels on the obscured *wine quality* dataset. Shaded regions indicate 95% confidence interval bounds.

6 Summary

In this dissertation I have presented an alternative approach to the classification of data with missing values. Firstly, I showed how to transform a classical classifier into an interval version, and how to calculate corresponding interval predictions. Secondly, I presented the possibility of ensemble classification of interval predictions through aggregation operators and thresholding strategies. Thirdly, I described how the proposed approach can be successfully applied in the real problem of supporting ovarian tumour diagnosis. Finally, I showed how the approach behaves in an experiment performed on datasets from real applications in a range of fields.

The experiment with UCI datasets confirmed the possibility of improvement of classification when the input data have missing values. This can be achieved either by a simple uncertaintification step, or by imputation/aggregation. In the experiment, the aggregation strategies did not significantly outperform imputation, giving mostly similar results. However, each applied classification problem has its own configuration and priorities. Hence, when one has to construct applied prediction models, it may be worthwhile to investigate whether aggregation strategies give better results.

The approach with aggregation strategies provides a new way of handling missing data. In the imputation methods we can see how the classification performs when we insert new values. In the proposed method this issue is addressed differently, i.e. we check the performance of predictions when we remove even more data. We have therefore introduced a novel approach to the problem of handling missing data. Moreover, the results obtained in the medical case are the motivation for new research in the field of handling missing data through the framework of the aggregation functions [70].

Appendices

A Aggregation operators

This appendix lists all of the aggregation methods evaluated in our research. There are four groups of operators: r-means, OWA, integrals and t-operations. Each group is represented both in numerical and interval aggregation mode.

A.1 Weight calculation strategies

Many aggregation operators involve assigning appropriate weights to input values. The problem is the same regardless of the mode of aggregation. Thus, we combine the description of different weight calculation strategies into one subsection.

The following weight calculation strategies were implemented in this research:

- constant value:

$$\omega_1([a,b]) = 1\,,$$

- interval length:

$$\omega_{\mathrm{wid}}([a,b]) = 1 - (b-a)\,,$$

- interval endpoint distance from 0.5:

$$\omega_{\text{ep}}([a,b]) = \begin{cases} 0, & \text{if } a \leq 0.5 \leq b \\ 2(a-0.5), & \text{if } a \geq 0.5 \\ 2(0.5-b), & \text{otherwise} \end{cases},$$

- interval midpoint distance from 0.5:

$$\omega_{\text{mp}}([a,b]) = 2 \cdot |0.5 - \frac{a+b}{2}|\,,$$

- lower and upper bounds of interval and interval midpoint (ω_{min}, ω_{max} and ω_{mp}, respectively),
- combined interval midpoint and width

$$\omega_{\text{wm}}([a,b]) = \frac{a+b}{2} \cdot (1-(b-a))\,.$$

A.2 Numerical mode

Aggregation methods that operate in this mode use a single value that represents the whole interval. Such a representative of the interval $\hat{x}$ is denoted by $\text{REP}(\hat{x})$. We evaluated the three most obvious representatives, namely the lower (REP_{min}) and upper (REP_{max}) bound and midpoint of the interval (REP_{mp}). This procedure simplifies the problem to classical non-interval aggregation. For more information about the presented aggregation methods we refer the reader to [51].

A.2.1 Weighted r-means

The weighted mean is probably the most commonly used method of aggregation. r-means generalise this concept by the using r-th power of each argument (for $r = 1$ the r-mean becomes the classical weighted mean). For weighted means, the selection of weights is crucial and determines the final outcome of the aggregation. The general formula for weighted r-mean is the following:

$$\text{AGG}_{\text{mean}}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \sqrt[r]{\frac{\sum_{i=1}^{n} \omega(\hat{x}_i) \cdot \text{REP}(\hat{x}_i)^r}{\sum_{i=1}^{n} \omega(\hat{x}_i)}}\,. \tag{A.1}$$

A.2.2 Ordered weighted average (OWA)

This class of aggregation operators was developed by Yager in 1988 [71]:

$$\textsc{Agg}_{\text{OWA}}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \frac{\sum_{i=1}^{n} \omega_i \cdot \textsc{rep}(\hat{x}_{\pi(i)})}{\sum_{i=1}^{n} \omega_i}\,. \tag{A.2}$$

In contrast to the arithmetic mean, in the *ordered weighted average* the weight vector is constant, while the input variables are ordered with respect to a certain criterion. Our implementation of OWA supports the ordering of input values with respect to any of the weights introduced in Section A.1. Such an ordering obtained from weight ω is denoted by π_ω. The following predefined weight vectors are used in medical evaluation:

- $(0, 0.25, 0.5, 0.5, 0.75, 1)$ – denoted by ω_{inc},
- $(1, 0.75, 0.5, 0.5, 0.25, 0)$ – denoted by ω_{dec},
- $(0.1, 0.5, 1, 1, 0.5, 0.1)$ – denoted by ω_{hill},
- $(1, 0.5, 0.1, 0.1, 0.5, 1)$ – denoted by ω_{pit}.

In the UCI datasets evaluation, the following predefined weight vectors are used:

- $(0, 0.25, 0.5, 0.75, 1)$ – denoted by ω_{inc},
- $(1, 0.75, 0.5, 0.25, 0)$ – denoted by ω_{dec},
- $(0.1, 0.55, 1, 0.55, 0.1)$ – denoted by ω_{hill},
- $(1, 0.55, 0.1, 0.55, 1)$ – denoted by ω_{pit}.

The vectors are afterwards normalised so that their elements sum to 1.

A.2.3 Choquet and Sugeno integrals

These are two classes of aggregation operators defined with the use of a measure μ. Their main advantage is that they are able to model interactions between input variables.

The Choquet integral is given by

$$\textsc{Agg}_{\text{Cho}}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \sum_{i=1}^{n} [\mu(H_i) - \mu(H_{i-1})] \cdot \textsc{rep}(\hat{x}_{\pi(i)}) \tag{A.3}$$

and the Sugeno integral is defined by

$$\mathrm{AGG}_{\mathrm{Sug}}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \max_{i=1 \text{ to } n} \left[\min\left(\mu(H_i), \mathrm{REP}(\hat{x}_{\pi(i)})\right)\right], \qquad (A.4)$$

where $H_i = \{\pi(1), \pi(2), \ldots, \pi(i)\}$, π is a non-decreasing permutation of input variables and μ is a measure. The following measures are implemented in this research:

- set cardinality

$$\mu_{\mathrm{card}}(H) = \frac{|H|}{n},$$

- (in medical evaluation) the additive measure

$$\mu_{\mathrm{AUC}}(\{h_1, h_2, \ldots\}) = \sum_{i=1} \mu(\{h_i\}),$$

  where the measure of a singleton was determined using the area under the ROC curve (AUC) [72] of the original diagnostic models (the greater the AUC, the higher the measure).

A.2.4 Triangular operations

An operation $t : [0,1] \times [0,1] \to [0,1]$ is a t-norm (triangular norm) if t is commutative, associative, non-decreasing and has 1 as neutral element. A similarly defined operation $s : [0,1] \times [0,1] \to [0,1]$ is a t-conorm (triangular conorm) if it has 0 as neutral element. T-norms together with t-conorms we call t-operations (triangular operations), and denote them as Φ[1].

The last class of numerical aggregation operators is based on the triangular operations, namely:

- t-norms (for $\alpha = 1$),
- t-conorms (for $\alpha = 1$),
- soft t-norms (for $\alpha < 1$),
- soft t-conorms (for $\alpha < 1$).

[1] A comprehensive discussion on triangular operations, soft triangular norms and conorms can be found in [73] and [74, Sections 2.4.3 and 4.3.2].

This class of operators is given by the formula

$$\mathrm{AGG}_{\Phi}(\hat{x}_1,\ldots,\hat{x}_n) = \frac{1-\alpha}{n}\sum_{i=1}^{n}\mathrm{REP}(\hat{x}_{\pi(i)}) + \alpha\cdot\Phi(\mathrm{REP}(\hat{x}_1),\ldots,\mathrm{REP}(\hat{x}_n))\,. \tag{A.5}$$

A.3 Interval mode

Interval mode utilises the whole of the interval information. The literature contains two approaches to adapting numerical aggregation strategies to operate on interval data. The first involves the use of interval arithmetic, and the second the application of the original operator to the lower and upper bound separately. Both methods are presented below.

A.3.1 Interval weighted r-means

These aggregation operators are obtained from numerical r-means by the use of interval arithmetic for all calculations. The formula is as follows:

$$\widehat{\mathrm{AGG}}_{\mathrm{mean}}(\hat{x}_1,\hat{x}_2,\ldots,\hat{x}_n) = \sqrt[r]{\frac{\sum_{i=1}^{n}\omega(\hat{x}_i)\times\hat{x}_i{}^{r}}{\sum_{i=1}^{n}\omega(\hat{x}_i)}}\,, \tag{A.6}$$

but now $\sum$ denotes the sum of intervals, and multiplication (division) is replaced by multiplication (division) of an interval by a constant.

A.3.2 Interval OWA

A generalisation of OWA to operate on intervals was proposed by Yager [75], [76]:

$$\begin{aligned}\widehat{\mathrm{AGG}}_{\mathrm{OWA}}(\hat{x}_1,\hat{x}_2,\ldots,\hat{x}_n) = [&\mathrm{AGG}_{\mathrm{OWA}}(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_n),\\ &\mathrm{AGG}_{\mathrm{OWA}}(\bar{x}_1,\bar{x}_2,\ldots,\bar{x}_n)]\,.\end{aligned} \tag{A.7}$$

The main idea is to apply an OWA operator to the lower and upper bounds of the input intervals separately, and to form an interval from the two results.

A.3.3 Interval Choquet and Sugeno integrals

An analogous approach was applied to define the interval Choquet and Sugeno integrals [75]. They are defined by

$$\widehat{\mathrm{AGG}}_{\mathrm{Cho}}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = [\mathrm{AGG}_{\mathrm{Cho}}(\underline{x}_{\pi(1)}, \underline{x}_{\pi(2)}, \ldots, \underline{x}_{\pi(n)}), \mathrm{AGG}_{\mathrm{Cho}}(\bar{x}_{\pi(1)}, \bar{x}_{\pi(2)}, \ldots, \bar{x}_{\pi(n)})] \tag{A.8}$$

and

$$\widehat{\mathrm{AGG}}_{\mathrm{Sug}}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = [\mathrm{AGG}_{\mathrm{Sug}}(\underline{x}_{\pi(1)}, \underline{x}_{\pi(2)}, \ldots, \underline{x}_{\pi(n)}), \mathrm{AGG}_{\mathrm{Sug}}(\bar{x}_{\pi(1)}, \bar{x}_{\pi(2)}, \ldots, \bar{x}_{\pi(n)})] \,. \tag{A.9}$$

A.3.4 Interval triangular norms and conorms

This approach can also be used to obtain interval aggregation operators based on triangular operations:

$$\widehat{\mathrm{AGG}}_{\Phi}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = [\mathrm{AGG}_{\Phi}(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n), \mathrm{AGG}_{\Phi}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)] \,. \tag{A.10}$$

B Thresholding strategies

Thresholding has the aim of converting a numerical or interval decision into a final decision. This appendix lists all implemented and evaluated strategies for both numerical and interval modes.

B.1 Numerical mode

For numerical decisions there is only one class of thresholding strategies, i.e. thresholding with margin $\epsilon \in [-0.5, 0.5]$ given by

$$\tau_\epsilon(a) = \begin{cases} y_1, & \text{if } a > 0.5 + \epsilon \\ y_2, & \text{if } a \leq 0.5 - \epsilon \\ \texttt{NA}, & \text{otherwise} \end{cases} .$$

B.2 Interval mode

For interval mode we evaluated three thresholding strategies. The first approach is to apply a numerical threshold to the interval representative, which results in

$$\hat{\tau}_{\text{REP},\epsilon}([a, b]) = \tau_\epsilon(\text{REP}([a, b])) .$$

The second is the interval version of thresholding with a margin given for each $\epsilon \in [-0.5, 0.5]$ by

$$\hat{\tau}_\epsilon([a, b]) = \begin{cases} y_1, & \text{if } a > 0.5 + \epsilon \\ y_2, & \text{if } b \leq 0.5 - \epsilon \\ \texttt{NA}, & \text{otherwise} \end{cases} .$$

The last approach involves calculation of the common part between intervals. Let $|[a, b]|$ denote the length of interval $[a, b]$. Then this thresholding strategy is given by

$$\hat{\tau}_{\text{cp}}([a,b]) \begin{cases} \texttt{NA}, & \text{if } |[a,b] \cap [0.5-\epsilon, 0.5+\epsilon]| \geq \\ & \quad \max\left(|[a,b] \cap [0.5+\epsilon, 1]|, |[a,b] \cap [0, 0.5-\epsilon]|\right) \\ y_1, & \text{if } |[a,b] \cap [0.5+\epsilon, 1]| > |[a,b] \cap [0, 0.5-\epsilon]| \\ y_2, & \text{otherwise} \end{cases}.$$

C Algorithm complexity analysis

The algorithms presented in Chapter 4 and Chapter 5 rely extensively on simulation data. In such a process, it might be interesting to investigate the algorithms in terms of the number of obscurations to perform on a data subset, i.e. how many times `NA` must be inserted in the data. With this additional knowledge one can make a rough estimation of the required computational resources and time.

Section 4.3 describes the simulation steps in the medical experiment. We can formulate this more generally: with r repetitions, for each k levels of obscurations we randomly draw $n_1 + n_2$ instances and in each instance we obscure exactly k features. The total number of obscurations is then equal to $r(n_1 + n_2)\frac{k(k+1)}{2}$.

A more interesting case is described in Algorithm 5.2. Let us see how the number of obscurations $s(n)$ looks for the first few n's, assuming the data has exactly one instance:

$$
\begin{aligned}
&\text{for } n = 2,\ s(n) = 1\,,\\
&\text{for } n = 3,\ s(n) = 2 + 3\,,\\
&\text{for } n = 4,\ s(n) = 3 + 5 + 6\,,\\
&\text{for } n = 5,\ s(n) = 4 + 7 + 9 + 10\,,\\
&\text{for } n = 6,\ s(n) = 5 + 9 + 12 + 14 + 15\,,\\
&\text{for } n = 7,\ s(n) = 6 + 11 + 15 + 18 + 20 + 21\,,\\
&\qquad \dots
\end{aligned}
$$

One can see that this can be expressed as

$$
\begin{aligned}
s(n) = {} & [n-1] + [2(n-1)-1] + [3(n-1)-3] \\
& + [4(n-1)-6] + [5(n-1)-10] + \dots\,,
\end{aligned}
$$

and finally we can formulate it as

$$\begin{aligned} s(n) &= \sum_{x=1}^{n-1} x(n-1) - \frac{(x-1)x}{2} \\ &= \sum_{x=1}^{n-1} x(n-1) - \frac{1}{2}\sum_{x=1}^{n-1} x^2 + \frac{1}{2}\sum_{x=1}^{n-1} x \\ &= \frac{n(n-1)^2}{2} - \frac{n(2n-1)(n-1)}{12} + \frac{n(n-1)}{4} \\ &= \frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}\,. \end{aligned}$$

D Results for UCI repository datasets

This section contains additional results from the experiments carried out on UCI datasets.

D.1 Census income

Attribute	Classifiers					
	OneR	*glm*	*nnet*	*svmLinear*	*rpart*	*knn*
education	-	✓	✓	✓	-	✓
marital status	✓	✓	✓	✓	✓	✓
occupation	-	-	✓	✓	-	-
race	-	✓	-	✓	-	-
sex	-	✓	✓	✓	✓	-
work class	-	-	✓	✓	-	-
age	-	✓	✓	✓	✓	✓
capital gain	-	✓	✓	✓	-	✓
capital loss	-	-	✓	✓	-	-
final weight	-	-	-	-	-	-
hours per week	-	✓	✓	✓	-	✓
relationship	-	✓	✓	✓	✓	✓

Table D.1: Predictors used by the classifiers in *census income* dataset. Features in the first group are always available.

Classifier	Acc	Sen	Spe	p-value
OneR	0.704	0.660	0.747	< 0.001
glm	0.727	0.703	0.751	< 0.001
nnet	0.762	0.734	0.792	< 0.001
svmLinear	0.782	0.747	0.818	< 0.001
rpart	0.758	0.670	0.847	< 0.001
knn	0.758	0.711	0.805	< 0.001

Table D.2: Performance of classifiers on the complete *census income* dataset

Model	Group	Acc	Dec	Sen	Spe
glm	Original classifier	0.774	0.376	0.734	0.814
nnet		0.766	0.354	0.756	0.775
svmLinear		0.734	0.354	0.665	0.803
rpart		0.728	0.511	0.616	0.835
knn		0.785	0.376	0.761	0.809
glm	Uncertaintified classifier	0.804	0.515	0.741	0.860
nnet		0.794	0.564	0.725	0.858
svmLinear		0.789	0.937	0.739	0.837
rpart		0.758	1.000	0.676	0.840
knn		0.808	0.630	0.792	0.825
glm & random forest	Imputation	0.796	1.000	0.753	0.838
OWA, interval (A.7) ω_{pit}, π_{wid}, $\hat{\tau}_{\text{mp},0.0}$	Aggregation strategy	0.803	1.000	0.747	0.858

Table D.3: Performance of classifiers on the obscured *census income* dataset

	Original classifiers	Uncertaintified classifiers	Imputation	Aggregation strategy
Original classifiers	-	0.035	< 0.001	< 0.001
Uncertaintified classifiers	0.035	-	0.040	0.040
Imputation	< 0.001	0.040	-	0.798
Aggregation strategy	< 0.001	0.040	0.798	-

Table D.4: Results of two-sided Student's t-test with Benjamini–Hochberg correction concerning whether by-obscurance-level-weighted means of AD-score differ on the obscured *census income* dataset

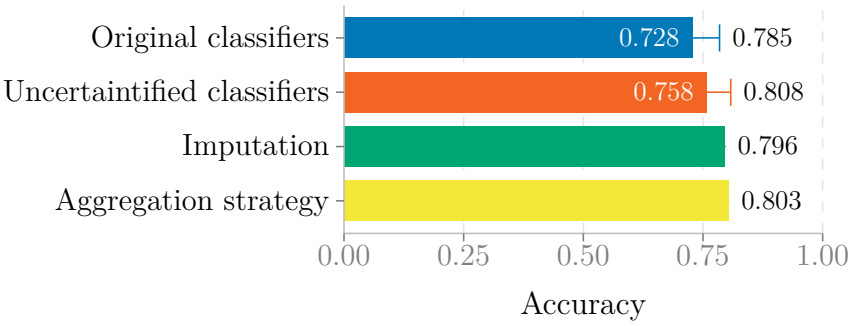

Figure D.1: Accuracy of prediction models on the obscured *census income* dataset. Whiskers indicate lower and upper bounds of accuracy of classifiers.

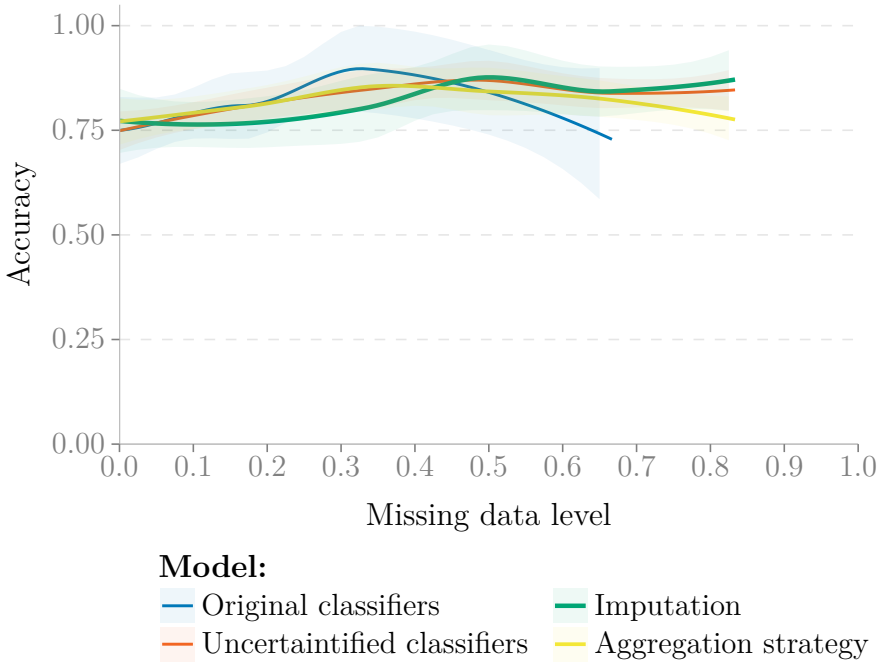

Figure D.2: Accuracy of prediction models regarding missing data levels on the obscured *census income* dataset. Shaded regions indicate 95% confidence interval bounds.

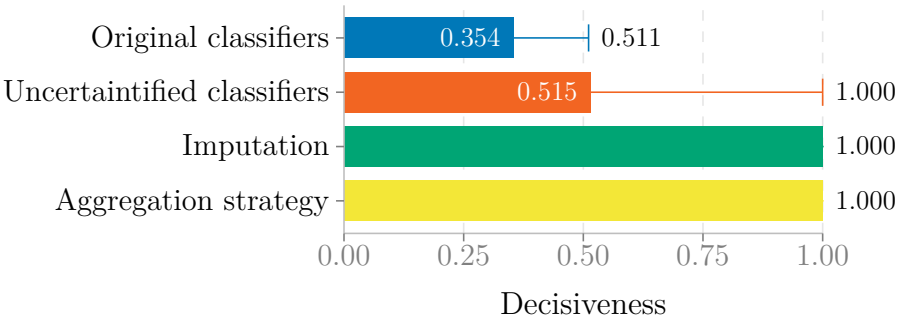

Figure D.3: Decisiveness of prediction models on the obscured *census income* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

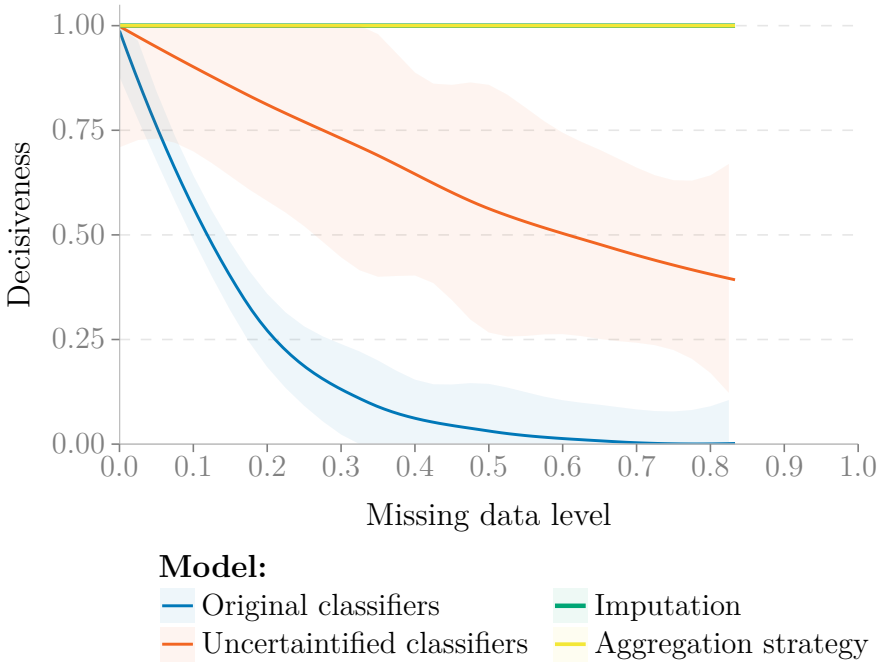

Figure D.4: Decisiveness of prediction models regarding missing data levels on the obscured *census income* dataset. Shaded regions indicate 95% confidence interval bounds.

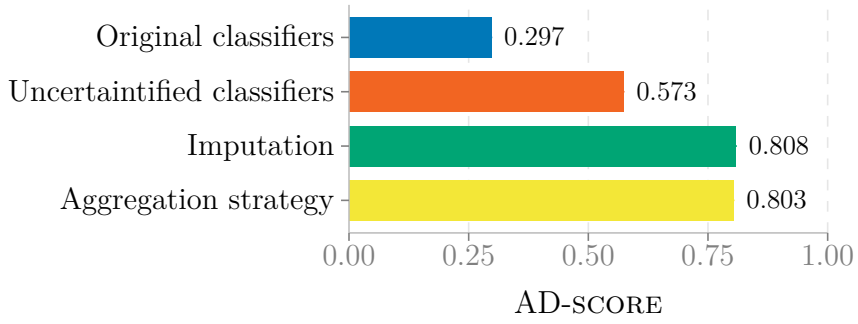

Figure D.5: AD-score of prediction models on the obscured *census income* dataset

D.2 Credit card

Attribute	Classifier					
	OneR	*glm*	*nnet*	*svmLinear*	*rpart*	*knn*
age	-	-	-	-	-	-
bill amount 1	-	-	-	-	-	-
bill amount 2	-	-	-	-	-	-
bill amount 3	-	✓	-	-	-	-
bill amount 4	-	-	-	-	-	-
bill amount 5	-	✓	-	-	-	-
bill amount 6	-	✓	-	-	-	-
education	-	-	-	-	-	-
limit balance	-	-	-	-	✓	-
marriage	-	-	-	-	-	-
pay 0	✓	✓	✓	✓	✓	✓
pay 2	-	✓	-	-	✓	-
pay 3	-	-	-	-	-	-
pay 4	-	-	-	✓	✓	-
pay 5	-	-	-	✓	✓	-
pay 6	-	✓	-	✓	✓	-
pay amount 1	-	-	✓	-	-	-
pay amount 2	-	✓	-	-	-	-
pay amount 3	-	-	-	-	-	-
pay amount 4	-	-	-	-	-	-
pay amount 5	-	✓	-	-	-	-
pay amount 6	-	-	-	-	-	-
sex	-	-	-	-	-	-

Table D.5: Predictors used by classifiers in the *credit card* dataset

Classifier	Acc	Sen	Spe	p-value
OneR	0.547	0.636	0.457	0.030
glm	0.643	0.711	0.575	< 0.001
nnet	0.682	0.858	0.506	< 0.001
svmLinear	0.691	0.885	0.496	< 0.001
rpart	0.632	0.897	0.367	< 0.001
knn	0.702	0.854	0.551	< 0.001

Table D.6: Performance of classifiers on the complete *credit card* dataset

Model	Group	Acc	Dec	Sen	Spe
glm	Original classifier	0.660	0.362	0.735	0.586
nnet		0.662	0.547	0.825	0.496
svmLinear		0.657	0.428	0.749	0.563
rpart		0.642	0.402	0.937	0.335
knn		0.662	0.681	0.833	0.485
glm	Uncertaintified classifier	0.660	0.424	0.724	0.602
nnet		0.662	0.642	0.857	0.448
svmLinear		0.679	0.586	0.675	0.682
rpart		0.625	0.718	0.890	0.351
knn		0.662	0.681	0.833	0.485
glm & *mice*	Imputation	0.620	1.000	0.751	0.489
t-operation, interval (A.10) $s_{\min}$, $\alpha = 0.75$, $\hat{\tau}_{\min,0.0}$	Aggregation strategy	0.622	1.000	0.758	0.486

Table D.7: Performance of classifiers on the obscured *credit card* dataset

	Original classifiers	Uncertaintified classifiers	Imputation	Aggregation strategy
Original classifiers	-	0.171	0.002	0.002
Uncertaintified classifiers	0.171	-	0.002	0.002
Imputation	0.002	0.002	-	0.888
Aggregation strategy	0.002	0.002	0.888	-

Table D.8: Results of two-sided Student's t-test with Benjamini–Hochberg correction concerning whether by-obscurance-level-weighted means of AD-score differ on the obscured *credit card* dataset

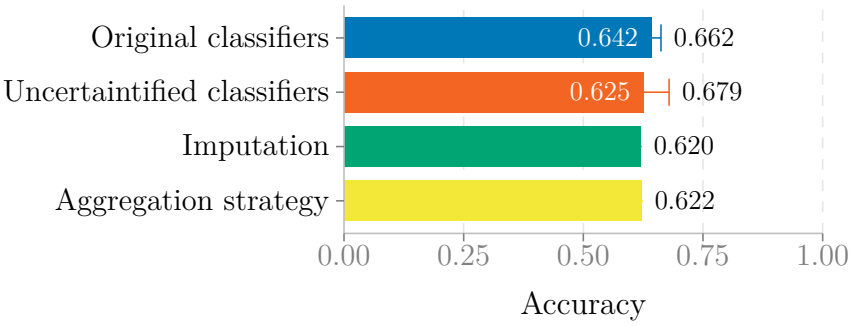

Figure D.6: Accuracy of prediction models on the obscured *credit card* dataset. Whiskers indicate lower and upper bounds of accuracy of classifiers.

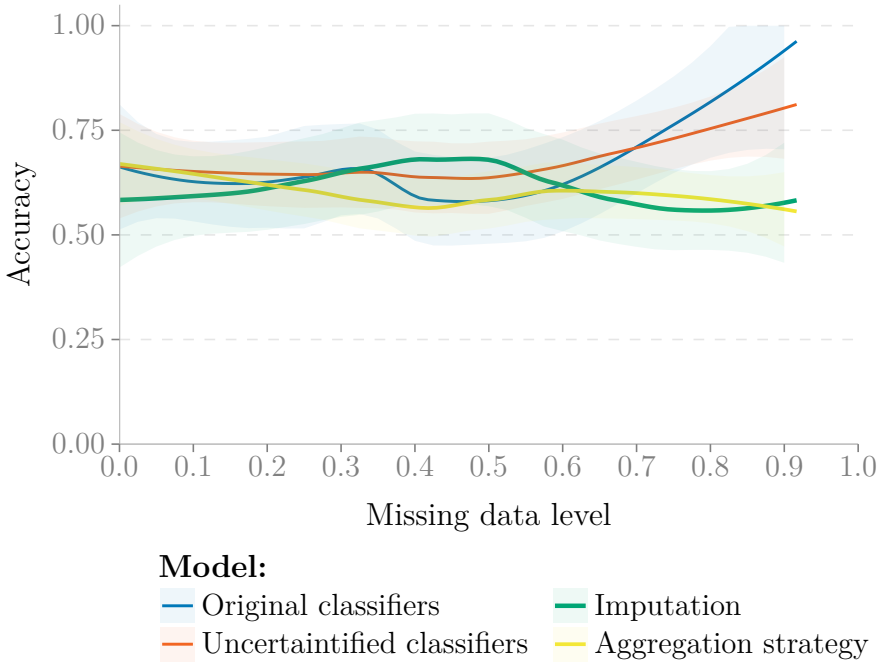

Figure D.7: Accuracy of prediction models regarding missing data levels on the obscured *credit card* dataset. Shaded regions indicate 95% confidence interval bounds.

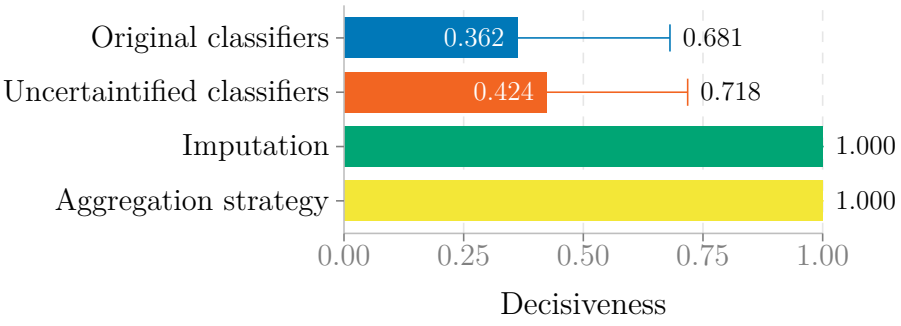

Figure D.8: Decisiveness of prediction models on the obscured *credit card* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

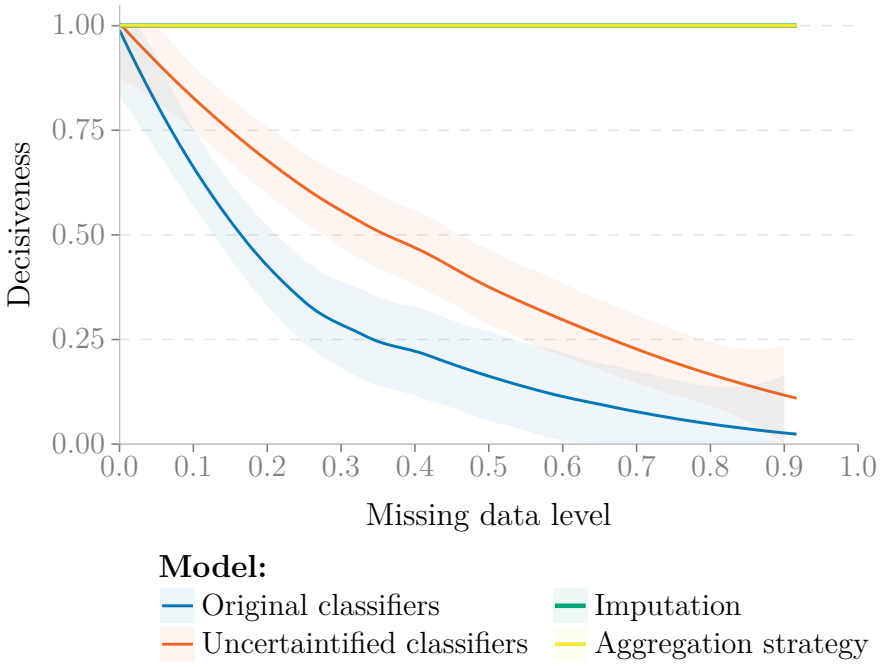

Figure D.9: Decisiveness of prediction models regarding missing data levels on the obscured *credit card* dataset. Shaded regions indicate 95% confidence interval bounds.

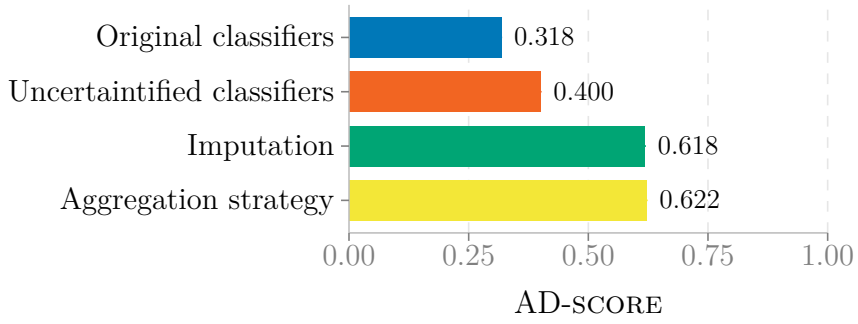

Figure D.10: AD-SCORE of prediction models on the obscured *credit card* dataset

D.3 Magic

Attribute	Classifier					
	OneR	*glm*	*nnet*	*svmLinear*	*rpart*	*knn*
f-alpha	✓	✓	✓	✓	✓	✓
f-asym.	-	✓	-	✓	✓	✓
f-conc.	-	✓	-	✓	✓	✓
f-conc.-1	-	✓	-	✓	✓	✓
f-distance	-	✓	-	-	✓	✓
f-length	-	✓	✓	✓	✓	✓
f-M3-long.	-	✓	-	✓	✓	✓
f-M3-trans.	-	✓	-	-	✓	-
f-size	-	✓	-	✓	✓	✓
f-width	-	✓	-	✓	✓	✓

Table D.9: Predictors used by classifiers in the *magic* dataset

Classifier	ACC	SEN	SPE	p-value
OneR	0.556	0.563	0.550	0.009
glm	0.704	0.755	0.653	< 0.001
nnet	0.778	0.783	0.773	< 0.001
svmLinear	0.773	0.849	0.697	< 0.001
rpart	0.755	0.720	0.790	< 0.001
knn	0.774	0.850	0.698	< 0.001

Table D.10: Performance of classifiers on the complete *magic* dataset

Model	Group	Acc	Dec	Sen	Spe
glm	Original classifier	0.793	0.334	0.856	0.731
nnet		0.785	0.531	0.802	0.768
svmLinear		0.802	0.349	0.858	0.746
rpart		0.751	0.334	0.802	0.701
knn		0.788	0.340	0.842	0.734
glm	Uncertaintified classifier	0.789	0.407	0.841	0.738
nnet		0.787	0.600	0.762	0.808
svmLinear		0.796	0.432	0.832	0.763
rpart		0.735	0.589	0.694	0.772
knn		0.797	0.444	0.821	0.778
svmLinear & *mice*	Imputation	0.748	1.000	0.751	0.744
t-operation, numeric (A.5) $\text{REP}_{\min}$, $s_{\min}$, $\alpha = 1.0$, $\tau_{0.0}$	Aggregation strategy	0.673	1.000	0.796	0.550

Table D.11: Performance of classifiers on the obscured *magic* dataset

	Original classifiers	Uncertaintified classifiers	Imputation	Aggregation strategy
Original classifiers	-	0.068	< 0.001	< 0.001
Uncertaintified classifiers	0.068	-	< 0.001	< 0.001
Imputation	< 0.001	< 0.001	-	< 0.001
Aggregation strategy	< 0.001	< 0.001	< 0.001	-

Table D.12: Results of two-sided Student's t-test with Benjamini–Hochberg correction concerning whether by-obscurance-level-weighted means of AD-score differ on the obscured *magic* dataset

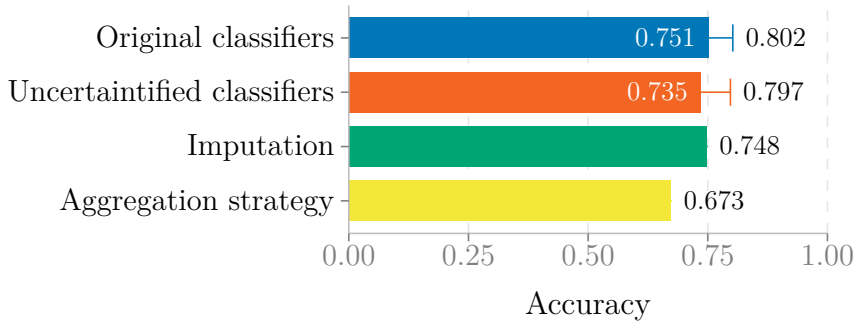

Figure D.11: Accuracy of prediction models on the obscured *magic* dataset. Whiskers indicate lower and upper bounds of accuracy of classifiers.

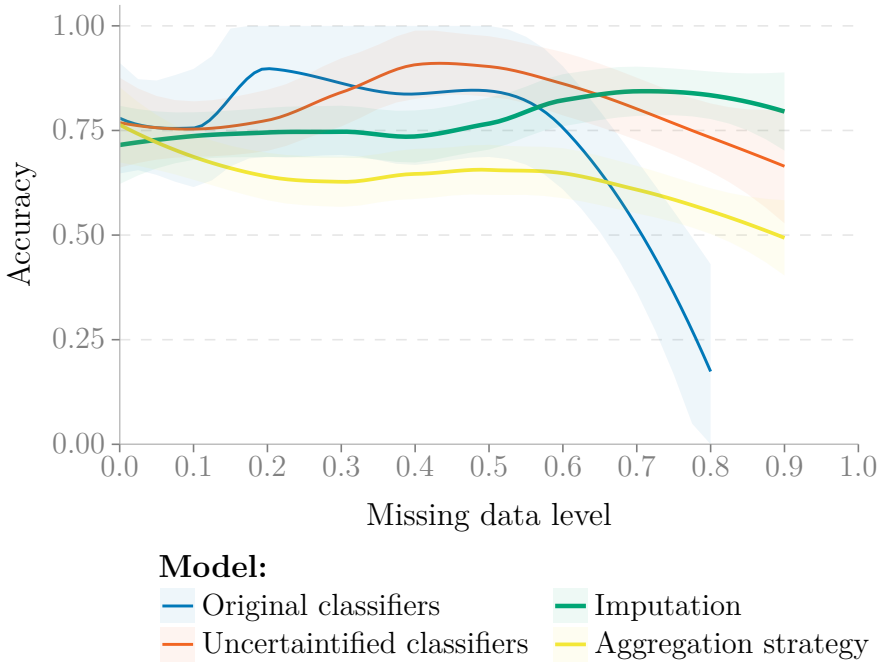

Figure D.12: Accuracy of prediction models regarding missing data levels on the obscured *magic* dataset. Shaded regions indicate 95% confidence interval bounds.

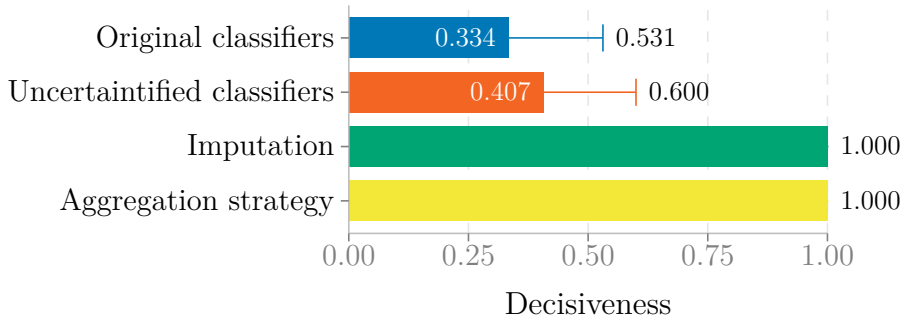

Figure D.13: Decisiveness of prediction models on the obscured *magic* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

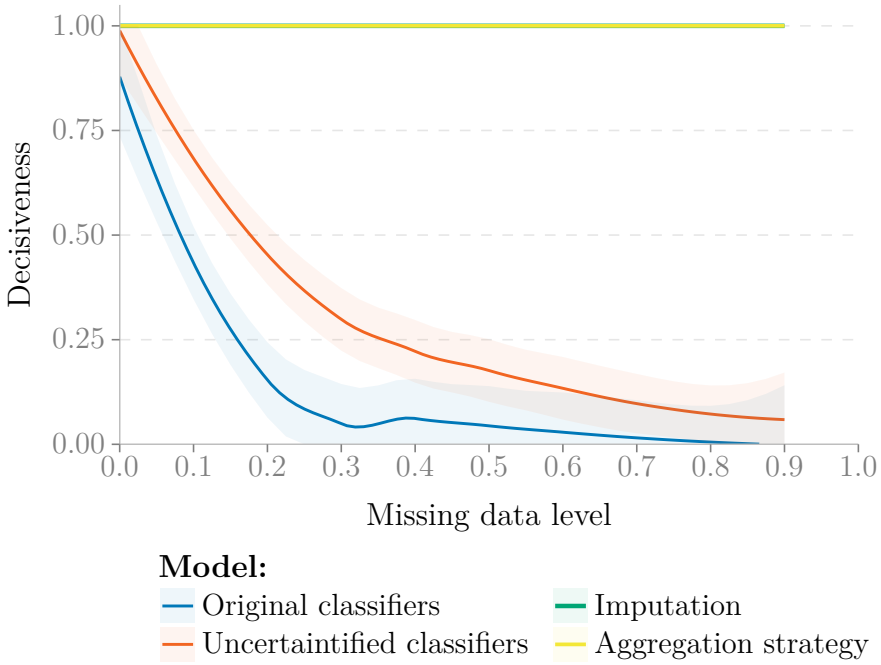

Figure D.14: Decisiveness of prediction models regarding missing data levels on the obscured *magic* dataset. Shaded regions indicate 95% confidence interval bounds.

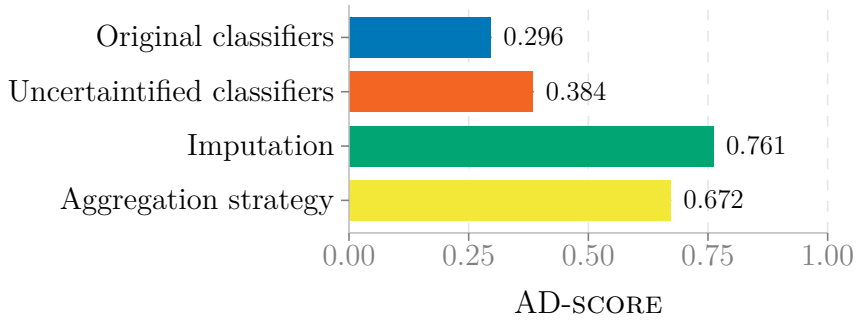

Figure D.15: AD-SCORE of prediction models on the obscured *magic* dataset

D.4 Wine quality

Attribute	Classifier					
	OneR	*glm*	*nnet*	*svmLinear*	*rpart*	*knn*
alcohol	✓	✓	✓	✓	✓	✓
chlorides	-	-	-	✓	✓	-
citric acid	-	✓	✓	✓	-	✓
colour	-	-	-	-	-	-
density	-	✓	✓	✓	✓	✓
fixed acidity	-	-	-	-	✓	✓
free sulphur dioxide	-	-	-	✓	-	-
pH	-	-	-	-	-	-
residual sugar	-	-	-	-	-	-
sulphates	-	-	-	✓	✓	✓
total sulphur dioxide	-	-	-	✓	✓	✓
volatile acidity	-	✓	✓	✓	-	✓

Table D.13: Predictors used by classifiers in the *wine quality* dataset

Classifier	ACC	SEN	SPE	p-value
OneR	0.648	0.735	0.560	< 0.001
glm	0.691	0.697	0.684	< 0.001
nnet	0.734	0.733	0.735	< 0.001
svmLinear	0.711	0.770	0.653	< 0.001
rpart	0.658	0.733	0.583	< 0.001
knn	0.687	0.681	0.694	< 0.001

Table D.14: Performance of classifiers on the complete *wine quality* dataset

Model	Group	Acc	Dec	Sen	Spe
glm	Original classifier	0.727	0.425	0.737	0.718
nnet		0.720	0.425	0.727	0.713
svmLinear		0.753	0.360	0.730	0.775
rpart		0.729	0.380	0.880	0.574
knn		0.724	0.362	0.740	0.707
glm	Uncertaintified classifier	0.737	0.494	0.737	0.737
nnet		0.733	0.491	0.731	0.735
svmLinear		0.758	0.418	0.730	0.785
rpart		0.724	0.651	0.865	0.583
knn		0.734	0.413	0.801	0.663
svmLinear & *mice*	Imputation	0.678	1.000	0.631	0.724
weighted mean, interval (A.6)	Aggregation strategy	0.658	1.000	0.596	0.721
$\omega_{\min}$, $r = 0.5$, $\hat{\tau}_{\text{mp},0.0}$					

Table D.15: Performance of classifiers on the obscured *wine quality* dataset

	Original classifiers	Uncertaintified classifiers	Imputation	Aggregation strategy
Original classifiers	-	0.064	< 0.001	< 0.001
Uncertaintified classifiers	0.064	-	< 0.001	< 0.001
Imputation	< 0.001	< 0.001	-	0.811
Aggregation strategy	< 0.001	< 0.001	0.811	-

Table D.16: Results of two-sided Student's t-test with Benjamini–Hochberg correction concerning whether by-obscurance-level-weighted means of AD-score differ on the obscured *wine quality* dataset

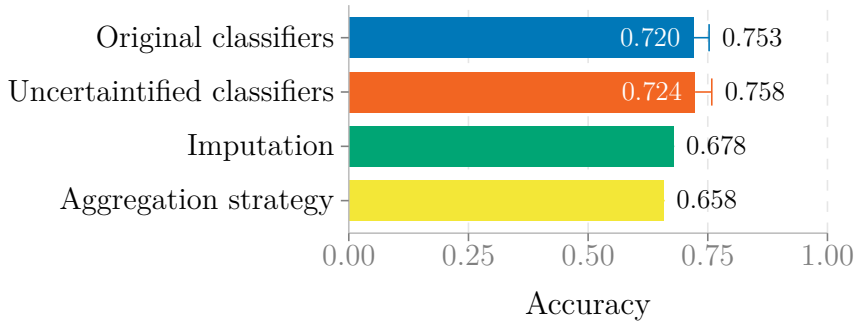

Figure D.16: Accuracy of prediction models on the obscured *wine quality* dataset. Whiskers indicate lower and upper bounds of accuracy of classifiers.

Figure D.17: Accuracy of prediction models regarding missing data levels on the obscured *wine quality* dataset. Shaded regions indicate 95% confidence interval bounds.

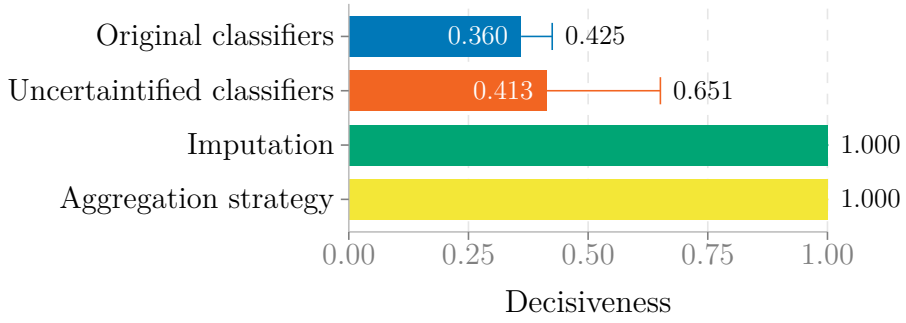

Figure D.18: Decisiveness of prediction models on the obscured *wine quality* dataset. Whiskers indicate lower and upper bounds of decisiveness of classifiers.

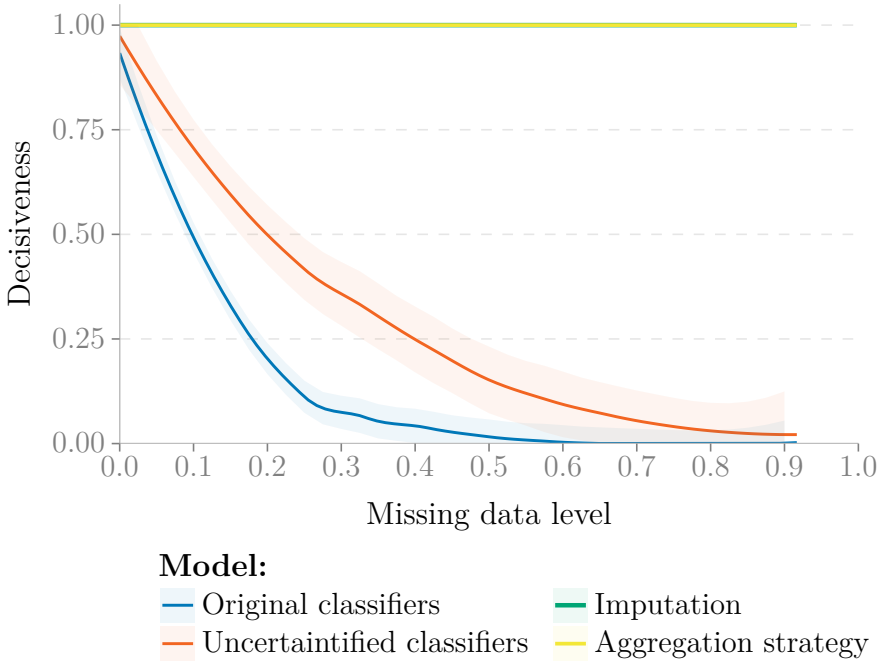

Figure D.19: Decisiveness of prediction models regarding missing data levels on the obscured *wine quality* dataset. Shaded regions indicate 95% confidence interval bounds.

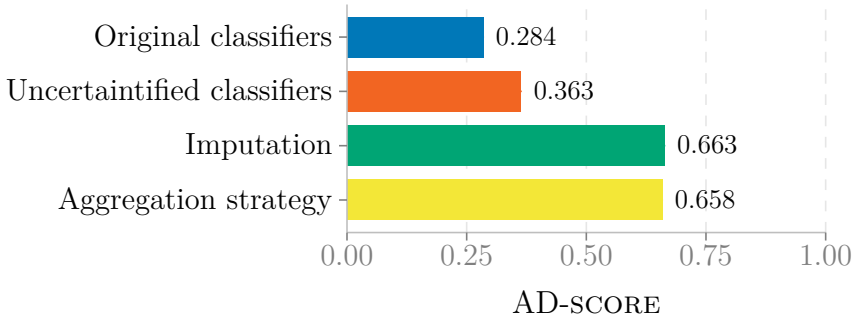

Figure D.20: AD-score of prediction models on the obscured *wine quality* dataset

List of Symbols

Basic math

Notation	Description
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integers
$\mathbb{N}_0$	set of natural numbers with 0
$\varnothing$	empty set
T	transpose of vector or matrix
$\#$	number sign
$\mathcal{I}_{[0,1]}$	set of all unit intervals
$\in_{\mathsf{E}}$	vector embedding
p_i	probability
P	set of probabilities

Datasets

Notation	Description
x_i	attribute (feature) of instance
$\hat{x}_i$	interval attribute
$\underline{x}_i$	lower bound of interval attribute
$\bar{x}_i$	upper bound of interval attribute
X_i	domain of attribute
$\hat{X}_i$	set of all nonempty closed subintervals of X_i
$\mathbf{x}$	instance; input vector for classifier
$\hat{\mathbf{x}}$	interval instance
$\underline{\mathbf{x}}$	lower bound of interval instance
$\bar{\mathbf{x}}$	upper bound of interval instance

Notation	Description
$\underset{\sim}{x}$	lower bound of attribute of interval instance
$\tilde{x}$	upper bound of attribute of interval instance
X	domain of instance
$\hat{X}$	interval domain of instance
y_i	class of an instance
Y	domain of a class
D	dataset
$\|D\|$	number of instances in dataset
D^{train}	training set
D^{test}	test set
`NA`	missing value
Imp	imputation method
D^{fs}	feature selection dataset
D^{cl}	classification dataset
D^{ob}	obscuration dataset
D^{un}	obscured dataset

Classification

Notation	Description
f	scoring function
θ	threshold (cutoff) of classifier
$\hat{f}$	interval scoring function
g	classification model
$\hat{g}$	interval classification model
f^{sco}	scoring function of scoring system
γ_j	scoring points
S_j	scoring points domain
s_j	scoring step function
q_i	point scoring function
Q_i	domain of point scoring function
g^{sco}	scoring system
θ^{sco}	threshold (cutoff) of scoring system
ξ	scoring normalisation function
f^{lgr}	scoring function of logistic regression
$\mathbf{v}$	logistic regression parameter vector

Notation	Description
u_i	logistic regression weight variable
$\mathbf{u}$	logistic regression weight vector
g^{lgr}	logistic regression
θ^{lgr}	threshold (cutoff) of logistic regression
ϕ	logistic regression normalisation function
T	set of nodes of binary tree
E	set of edges of binary tree
(T, E)	binary tree
t_i	node of binary tree
ρ	height of binary tree
f^{tree}	scoring function of classification tree
θ^{tree}	threshold (cutoff) of classification tree
g^{tree}	classification tree
h	ensemble classification model (ensemble classifier)
θ^{ens}	threshold (cutoff) of ensemble classifier
TP	true positive
TN	true negative
FP	false positive
FN	false negative
PERF	generic performance measure
ACC	accuracy
SEN	sensitivity
SPE	specificity
DEC	decisiveness
AGGSTR	aggregation strategy
AD-SCORE	accuracy-decisiveness score

Aggregation operators

Notation	Description
ω	weight strategy
ω_1	weight by constant
ω_{wid}	weight by interval length
ω_{ep}	weight by interval endpoint distance from 0.5
ω_{mp}	weight by interval midpoint distance from 0.5

Notation	Description
$\omega_{\min}$	weight by lower bound of interval
$\omega_{\max}$	weight by upper bound of interval
ω_{mp}	weight by midpoint of interval
ω_{wm}	weight by combined midpoint of interval and its width
ω_{inc}	increasing weights
ω_{dec}	decreasing weights
ω_{hill}	*hill* weights
ω_{pit}	*pit* weights
REP	representative of interval
$\mathrm{REP}_{\min}$	lower bound representative of interval
$\mathrm{REP}_{\max}$	upper bound representative of interval
$\mathrm{REP}_{\mathrm{mp}}$	midpoint representative of interval
AGG	numeric aggregation operator
$\mathrm{AGG}_{\mathrm{mean}}$	numeric weighted r-mean
$\mathrm{AGG}_{\mathrm{OWA}}$	ordered weighted average (OWA)
$\mathrm{AGG}_{\mathrm{Cho}}$	numeric Choquet integral
$\mathrm{AGG}_{\mathrm{Sug}}$	numeric Sugeno integral
π	non-decreasing permutation of input variables
μ	measure
μ_{card}	set cardinality measure
μ_{AUC}	additive measure with measure of singleton determined using the area under receiver operating characteristic curve
Φ	triangular operation
AGG_{Φ}	numeric aggregation operator based on triangular operation
$\widehat{\mathrm{AGG}}$	interval aggregation operator
$\widehat{\mathrm{AGG}}_{\mathrm{mean}}$	interval weighted r-mean
$\widehat{\mathrm{AGG}}_{\mathrm{OWA}}$	interval ordered weighted average (OWA)
$\widehat{\mathrm{AGG}}_{\mathrm{Cho}}$	interval Choquet integral
$\widehat{\mathrm{AGG}}_{\mathrm{Sug}}$	interval Sugeno integral
$\widehat{\mathrm{AGG}}_{\Phi}$	interval aggregation operator based on triangular operation

Thresholding strategies

Notation	Description
τ	numeric thresholding strategy
τ_ϵ	numeric thresholding with margin
$\hat{\tau}$	interval thresholding strategy
$\hat{\tau}_{\text{REP},\epsilon}$	interval thresholding through numerical threshold on interval representative
$\hat{\tau}_\epsilon$	interval thresholding with margin
$\hat{\tau}_{\text{cp}}$	interval thresholding through common part of intervals

List of Algorithms

List of Figures

List of Tables

References

[1] R. L. Siegel, K. D. Miller, *et al.*, Cancer Statistics, 2017, *CA: A Cancer Journal for Clinicians*, vol. 67, no. 1, pp. 7–30, 2017. DOI: 10.3322/caac.21387.

[2] World Health Organization, Department of Information, Evidence and Research. (2016). Mortality database, [Online]. Available: http://www.who.int/healthinfo/statistics/mortality_rawdata/en/index.html (visited on 01/11/2017).

[3] World Health Organization, International Agency for Research on Cancer. (2016). WHO cancer mortality database (IARC), [Online]. Available: http://www-dep.iarc.fr/WHOdb/WHOdb.htm (visited on 01/11/2017).

[4] D. Timmerman, P. Schwärzler, *et al.*, Subjective assessment of adnexal masses with the use of ultrasonography: an analysis of interobserver variability and experience, *Ultrasound in Obstetrics & Gynecology*, vol. 13, no. 1, pp. 11–16, 1999. DOI: 10.1046/j.1469-0705.1999.13010011.x.

[5] P. DePriest, D. Shenson, *et al.*, A morphology index based on sonographic findings in ovarian cancer, *Gynecologic Oncology*, vol. 51, no. 1, pp. 7–11, 1993. DOI: 10.1006/gyno.1993.1238.

[6] J. L. Alcázar, L. T. Mercé, *et al.*, A new scoring system to differentiate benign from malignant adnexal masses, *Obstetrical & Gynecological Survey*, vol. 58, no. 7, pp. 462–463, 2003. DOI: 10.1097/01.ogx.0000074382.15607.65.

[7] D. Szpurek, R. Moszyński, *et al.*, An ultrasonographic morphological index for prediction of ovarian tumor malignancy, *European Journal of Gynaecological Oncology*, vol. 26, no. 1, pp. 51–54, 2005.

[8] D. Timmerman, A. C. Testa, *et al.*, Simple ultrasound-based rules for the diagnosis of ovarian cancer, *Ultrasound in Obstetrics & Gynecology*, vol. 31, no. 6, pp. 681–690, 2008. DOI: 10.1002/uog.5365.

[9] R. Moszyński, Ocena przydatności teorii zbiorów przybliżonych w diagnostyce guzów jajnika. (Polish) [Evaluation of usefulness of rough set theory in diagnostics of ovarian tumours], PhD thesis, Poznan University of Medical Sciences, Poland, 2003.

[10] D. Timmerman, A. C. Testa, *et al.*, Logistic regression model to distinguish between the benign and malignant adnexal mass before surgery: a multicenter study by the International Ovarian Tumor Analysis Group, *Journal of Clinical Oncology*, vol. 23, no. 34, pp. 8794–8801, 2005. DOI: 10.1200/jco.2005.01.7632.

[11] R. G. Moore, D. S. McMeekin, *et al.*, A novel multiple marker bioassay utilizing HE4 and CA125 for the prediction of ovarian cancer in patients with a pelvic mass, *Gynecologic Oncology*, vol. 112, no. 1, pp. 40–46, 2009. DOI: 10.1016/j.ygyno.2008.08.031.

[12] B. Van Calster, K. Van Hoorde, *et al.*, Evaluating the risk of ovarian cancer before surgery using the ADNEX model to differentiate between benign, borderline, early and advanced stage invasive, and secondary metastatic tumours: prospective multicentre diagnostic study, *BMJ – British Medical Journal*, vol. 349, 2014. DOI: 10.1136/bmj.g5920.

[13] A. Tailor, D. Jurkovic, *et al.*, Sonographic prediction of malignancy in adnexal masses using an artificial neural network, *BJOG: An International Journal of Obstetrics & Gynaecology*, vol. 106, no. 1, pp. 21–30, 1999. DOI: 10.1111/j.1471-0528.1999.tb08080.x.

[14] A. Smoleń, A. Czekierdowski, *et al.*, Use of multilayer perceptron artificial neural networks for the prediction of the probability of malignancy in adnexal tumors, *Ginekologia Polska*, vol. 74, no. 9, pp. 855–862, 2003.

[15] B. Van Calster, D. Timmerman, *et al.*, Preoperative diagnosis of ovarian tumors using Bayesian kernel-based methods, *Ultra-*

sound in Obstetrics & Gynecology, vol. 29, no. 5, pp. 496–504, 2007. DOI: 10.1002/uog.3996.

[16] B. Van Calster, D. Timmerman, *et al.*, Using Bayesian neural networks with ARD input selection to detect malignant ovarian masses prior to surgery, *Neural Computing and Applications*, vol. 17, no. 5-6, pp. 489–500, 2008. DOI: 10.1007/s00521-007-0147-1.

[17] E. Madu, V. Stalbovskaya, *et al.*, Preoperative ovarian cancer diagnosis using neuro-fuzzy approach, in *European Conference on Emergent Aspects on Clinical Data Analysis*, 2005.

[18] S. Tingulstad, B. Hagen, *et al.*, Evaluation of a risk of malignancy index based on serum CA125, ultrasound findings and menopausal status in the pre-operative diagnosis of pelvic masses, *BJOG: An International Journal of Obstetrics & Gynaecology*, vol. 103, no. 8, pp. 826–831, 1996. DOI: 10.1111/j.1471-0528.1996.tb09882.x.

[19] F. Amor, H. Vaccaro, *et al.*, Gynecologic Imaging Reporting and Data System A New Proposal for Classifying Adnexal Masses on the Basis of Sonographic Findings, *Journal of Ultrasound in Medicine*, vol. 28, no. 3, pp. 285–291, 2009. DOI: 10.7863/jum.2009.28.3.285.

[20] U. R. Acharya, M. M. R. Krishnan, *et al.*, Ovarian Tumor Characterization Using 3D Ultrasound, in *Ovarian Neoplasm Imaging*, L. Saba, U. R. Acharya, *et al.*, Eds. Boston, MA: Springer, 2013, pp. 399–412. DOI: 10.1007/978-1-4614-8633-6_25.

[21] S. Khazendar, H. Al-Assam, *et al.*, Automated classification of static ultrasound images of ovarian tumours based on decision level fusion, in *2014 6th Computer Science and Electronic Engineering Conference (CEEC)*, New York, NY: IEEE, 2014, pp. 148–153. DOI: 10.1109/CEEC.2014.6958571.

[22] V. Aramendía-Vidaurreta, R. Cabeza, *et al.*, Ultrasound Image Discrimination between Benign and Malignant Adnexal Masses Based on a Neural Network Approach, *Ultrasound in Medicine & Biology*, vol. 42, no. 3, pp. 742–752, 2016. DOI: 10.1016/j.ultrasmedbio.2015.11.014.

[23] M. Stukan, M. Dudziak, *et al.*, Usefulness of diagnostic indices comprising clinical, sonographic, and biomarker data for discriminating benign from malignant ovarian masses, *Journal of Ultrasound in Medicine*, vol. 34, no. 2, pp. 207–217, 2015. DOI: 10.7863/ultra.34.2.207.

[24] P. K. J. Han, W. M. P. Klein, *et al.*, Varieties of Uncertainty in Health Care: a conceptual taxonomy, *Medical Decision Making*, vol. 31, no. 6, pp. 828–838, 2011. DOI: 10.1177/0272989X10393976.

[25] S. Hatch, *Snowball in a Blizzard: A Physician's Notes on Uncertainty in Medicine*. New York, NY: Basic Books, 2016.

[26] B. R. Benacerraf, Ultrasonic diagnosis of ovarian masses: can the playing field be leveled and raised at the same time?, *American Journal of Obstetrics and Gynecology*, vol. 214, no. 4, pp. 419–421, 2016. DOI: 10.1016/j.ajog.2015.12.045.

[27] H. L. Semigran, D. M. Levine, *et al.*, Comparison of physician and computer diagnostic accuracy, *JAMA Internal Medicine*, vol. 176, no. 12, pp. 1860–1861, 2016. DOI: 10.1001/jamainternmed.2016.6001.

[28] A. Wójtowicz, P. Żywica, K. Szarzyński, R. Moszyński, S. Szubert, K. Dyczkowski, A. Stachowiak, D. Szpurek, and M. Wygralak, Dealing with Uncertainty in Ovarian Tumor Diagnosis, in *Modern Approaches in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications*, K. Atanassov, W. Homenda, *et al.*, Eds., Warsaw: SRI PAS/IBS PAN, 2014, pp. 151–158.

[29] R. Moszyński, P. Żywica, A. Wójtowicz, S. Szubert, S. Sajdak, A. Stachowiak, K. Dyczkowski, M. Wygralak, and D. Szpurek, Menopausal status strongly influences the utility of predictive models in differential diagnosis of ovarian tumors: An external validation of selected diagnostic tools, *Ginekologia Polska*, vol. 85, no. 12, pp. 892–899, 2014. DOI: 10.17772/gp/1879.

[30] P. Żywica, A. Wójtowicz, A. Stachowiak, and K. Dyczkowski, Improving medical decisions under incomplete data using interval-valued fuzzy aggregation, in *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and*

the European Society for Fuzzy Logic and Technology, J. M. Alonso, H. Bustince, *et al.*, Eds., Amsterdam: Atlantis Press, 2015, pp. 577–584. DOI: 10.2991/ifsa-eusflat-15.2015.83.

[31] A. Wójtowicz, P. Żywica, A. Stachowiak, and K. Dyczkowski, Interval-valued aggregation as a tool to improve medical diagnosis, in *Proceedings of 8th International Summer School on Aggregation Operators (AGOP 2015)*, J. M. Alonso, H. Bustince, *et al.*, Eds., Katowice: University of Silesia, 2015, pp. 239–244.

[32] A. Wójtowicz, P. Żywica, A. Stachowiak, and K. Dyczkowski, Solving the problem of incomplete data in medical diagnosis via interval modeling, *Applied Soft Computing*, vol. 47, pp. 424–437, 2016. DOI: 10.1016/j.asoc.2016.05.029.

[33] K. Dyczkowski, A. Wójtowicz, P. Żywica, A. Stachowiak, R. Moszyński, and S. Szubert, An intelligent system for computer-aided ovarian tumor diagnosis, in *Intelligent Systems'2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume 2: Tools, Architectures, Systems, Applications*, D. Filev, J. Jabłkowski, *et al.*, Eds. Cham: Springer International Publishing, 2015, pp. 335–343. DOI: 10.1007/978-3-319-11310-4_29.

[34] A. Stachowiak, K. Dyczkowski, A. Wójtowicz, P. Żywica, and M. Wygralak, A Bipolar View on Medical Diagnosis in Ova-Expert System, in *Flexible Query Answering Systems 2015: Proceedings of the 11th International Conference FQAS 2015, Cracow, Poland, October 26-28, 2015*, T. Andreasen, H. Christiansen, *et al.*, Eds. Cham: Springer International Publishing, 2016, pp. 483–492. DOI: 10.1007/978-3-319-26154-6_37.

[35] P. Żywica, K. Dyczkowski, A. Wójtowicz, A. Stachowiak, S. Szubert, and R. Moszyński, Development of a fuzzy-driven system for ovarian tumor diagnosis, *Biocybernetics and Biomedical Engineering*, vol. 36, no. 4, pp. 632–643, 2016. DOI: 10.1016/j.bbe.2016.08.003.

[36] A. Stachowiak, P. Żywica, K. Dyczkowski, and A. Wójtowicz, An Interval-Valued Fuzzy Classifier Based on an Uncertainty-Aware Similarity Measure, in *Intelligent Systems'2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume*

1: Mathematical Foundations, Theory, Analyses, P. Angelov, K. Atanassov, *et al.*, Eds. Cham: Springer International Publishing, 2015, pp. 741–751. DOI: 10.1007/978-3-319-11313-5_65.

[37] I. H. Witten, E. Frank, *et al.*, *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA: Elsevier BV, 2011. DOI: 10.1016/c2009-0-19715-5.

[38] G. James, D. Witten, *et al.*, *An Introduction to Statistical Learning*. New York, NY: Springer, 2013. DOI: 10.1007/978-1-4614-7138-7.

[39] L. Breiman, J. Friedman, *et al.*, *Classification and regression trees*. Boca Raton, FL: Chapman & Hall/CRC Press, 1984.

[40] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley, 2014. DOI: 10.1002/9781118914564.

[41] G. Louppe, Understanding Random Forests: From Theory to Practice, PhD thesis, University of Liege, Belgium, 2014. [Online]. Available: http://hdl.handle.net/2268/170309 (visited on 01/11/2017).

[42] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. New York, NY: Cambridge University Press, 2011. DOI: 10.1017/cbo9780511921803.

[43] T. de Waal, J. Pannekoek, *et al.*, *Handbook of Statistical Data Editing and Imputation*. Hoboken, NJ: Wiley-Blackwell, 2011. DOI: 10.1002/9780470904848.

[44] P. Cortez, A. Cerdeira, *et al.*, Modeling wine preferences by data mining from physicochemical properties, *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009. DOI: 10.1016/j.dss.2009.05.016.

[45] A. Wójtowicz, *UCI Machine Learning datasets for R*, 2017. DOI: 10.5281/zenodo.230896. [Online]. Available: https://github.com/andre-wojtowicz/uci-ml-to-r (visited on 01/05/2017).

[46] Ministry of Family, Labour and Social Policy (Poland). (2016). Pytania i odpowiedzi w kwestionariuszu do profilowania pomocy dla osób bezrobotnych. (Polish) [Questions and answers in the questionnaire for profiling support for the unemployed]. The document was made public by the Panoptykon Foundation under the Polish Act on Access to Public Information, [Online]. Available: https://pliki.panoptykon.org/DIP/Cyfrowy%20Nadz%C3%B3r%20-%20EOG%20I/Przychodz%C4%85ca%20-%20odpowiedzi%20na%20wniosek/MRPiPS_Kwestionariusz%20Informacja%20dla%20Fundacji%20Panoptykon%20o%20liczbie%20punkt%C3%B3w%20i%20algorytmie.pdf (visited on 01/11/2017).

[47] L. Breiman, Random Forests, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.

[48] S. van Buuren, Multiple imputation of discrete and continuous data by fully conditional specification, *Statistical Methods in Medical Research*, vol. 16, no. 3, pp. 219–242, 2007. DOI: 10.1177/0962280206074463.

[49] J. A. Nelder and R. Mead, A Simplex Method for Function Minimization, *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965. DOI: 10.1093/comjnl/7.4.308.

[50] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of IEEE International Conference on Neural Networks 1995*, vol. 4, 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

[51] G. Beliakov, A. Pradera, *et al.*, *Aggregation Functions: A Guide for Practitioners*. Berlin: Springer, 2007. DOI: 10.1007/978-3-540-73721-6.

[52] G. Deschrijver and E. Kerre, Aggregation operators in interval-valued fuzzy and Atanassov's intuitionistic fuzzy set theory, in *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, H. Bustince, F. Herrera, *et al.*, Eds., Berlin: Springer, 2008, pp. 183–203. DOI: 10.1007/978-3-540-73723-0_10.

[53] G. Beliakov, H. Bustince, *et al.*, On averaging operators for Atanassov's intuitionistic fuzzy sets, *Information Sciences*, vol. 181, no. 6, pp. 1116–1124, 2011. DOI: 10.1016/j.ins.2010.11.024.

[54] D. Timmerman, T. H. Bourne, *et al.*, A comparison of methods for preoperative discrimination between malignant and benign adnexal masses: the development of a new logistic regression model, *American Journal of Obstetrics and Gynecology*, vol. 181, no. 1, pp. 57–65, 1999. DOI: 10.1016/S0002-9378(99)70436-9.

[55] I. Jacobs, D. Oram, *et al.*, A risk of malignancy index incorporating CA 125, ultrasound and menopausal status for the accurate preoperative diagnosis of ovarian cancer, *BJOG: An International Journal of Obstetrics & Gynaecology*, vol. 97, no. 10, pp. 922–929, 1990. DOI: 10.1111/j.1471-0528.1990.tb02448.x.

[56] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Boca Raton, FL: Chapman & Hall/CRC Press, 1993.

[57] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, 2016. [Online]. Available: https://www.R-project.org/ (visited on 08/19/2016).

[58] M. Lichman, *UCI Machine Learning Repository*, 2013. [Online]. Available: http://archive.ics.uci.edu/ml (visited on 08/19/2016).

[59] S. Moro, P. Cortez, *et al.*, A data-driven approach to predict the success of bank telemarketing, *Decision Support Systems*, vol. 62, pp. 22–31, 2014. DOI: 10.1016/j.dss.2014.03.001.

[60] R. Kohavi, Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-tree Hybrid, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, Portland, OR: AAAI Press, 1996, pp. 202–207.

[61] I. Yeh and C. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 2473–2480, 2009. DOI: 10.1016/j.eswa.2007.12.020.

[62] R. Bock, A. Chilingarian, *et al.*, Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 516, no. 2–3, pp. 511–528, 2004. DOI: 10.1016/j.nima.2003.08.157.

[63] T. Hastie, R. Tibshirani, *et al.*, *The Elements of Statistical Learning*. New York, NY: Springer, 2009. DOI: 10.1007/978-0-387-84858-7.

[64] R. C. Holte, Very simple classification rules perform well on most commonly used datasets, *Machine Learning*, vol. 11, pp. 63–91, 1993. DOI: 10.1023/A:1022631118932.

[65] M. Kuhn, Building Predictive Models in R Using the caret Package, *Journal of Statistical Software*, vol. 28, no. 1, pp. 1–26, 2008. DOI: 10.18637/jss.v028.i05.

[66] D. J. Stekhoven and P. Bühlmann, MissForest—non-parametric missing value imputation for mixed-type data, *Bioinformatics*, vol. 28, no. 1, p. 112, 2011. DOI: 10.1093/bioinformatics/btr597.

[67] S. van Buuren and K. Groothuis-Oudshoorn, mice: Multivariate Imputation by Chained Equations in R, *Journal of Statistical Software*, vol. 45, no. 1, pp. 1–67, 2011. DOI: 10.18637/jss.v045.i03.

[68] A. Wójtowicz, *Timing results for BLAS (Basic Linear Algebra Subprograms) libraries in R*, 2016. DOI: 10.5281/zenodo.57910. [Online]. Available: https://github.com/andre-wojtowicz/blas-benchmarks (visited on 12/01/2016).

[69] E. G. W. S. Cleveland and W. M. Shyu, Local regression models, in *Statistical Models in S*, J. M. Chambers and T. J. Hastie, Eds., Pacific Grove, CA: Wadsworth & Brooks/Cole, 1992.

[70] G. Beliakov, D. Gómez, *et al.*, Approaches to learning strictly-stable weights for data with missing values, *Fuzzy Sets and Systems*, 2017, in press. DOI: 10.1016/j.fss.2017.02.003.

[71] R. R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988. DOI: 10.1109/21.87068.

[72] A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997. DOI: 10.1016/S0031-3203(96)00142-2.

[73] E. P. Klement, R. Mesiar, *et al.*, *Triangular Norms*. Dordrecht: Springer, 2000. DOI: 10.1007/978-94-015-9540-7.

[74] M. Wygralak, *Intelligent Counting Under Information Imprecision. Applications to Intelligent Systems and Decision Support*. Berlin: Springer, 2013. DOI: 10.1007/978-3-642-34685-9.

[75] R. Yager, OWA aggregation of intuitionistic fuzzy sets, *International Journal of General Systems*, vol. 38, no. 6, pp. 617–641, 2009. DOI: 10.1080/03081070902847689.

[76] R. Mesiar, A. Stupňanová, *et al.*, Generalizations of OWA operators, *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2154–2162, 2015. DOI: 10.1109/tfuzz.2015.2406888.