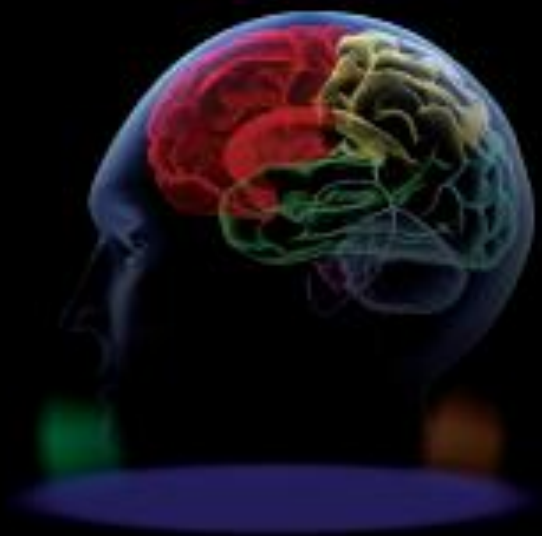
Intelligent lighting : a machine learning perspective

Kota Gopalakrishna, A.

Published: 01/01/2015

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Intelligent Lighting: A Machine Learning Perspective



Aravind Kota Gopalakrishna

Intelligent Lighting: A Machine Learning Perspective

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. C.J. van Duijn, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op dinsdag 21 april 2015 om 16:00 uur

door

Aravind Kota Gopalakrishna

geboren te Bangalore, India

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.E.H.L. Aarts
1e promotor:	prof.dr.A.Liotta
2e promotor:	prof.dr.J.J.Lukkien
copromotor:	dr.T.Ozcelebi
leden:	prof.dr.ir.J.H.Eggen
	prof.dr.L.Liu (University of Derby)
	prof.dr.M.M.M.Pai (Manipal University)
	prof.dr.H.de Ridder (Delft University of Technology)

Intelligent Lighting: A Machine Learning Perspective

Aravind Kota Gopalakrishna

The work in this thesis was carried out under the Smart Context-aware Services (SmaCS) project through Point One grant No. 10012172

A catalogue record is available from the Eindhoven University of Technology Library
ISBN:978-90-386-3821-8

Typeset using LaTeX
Cover design by: Dharmashri Offsets and Screens, Bengaluru, India
Printed by: Gildeprint - www.gildeprint.nl

“The world is ready to give up its secrets, if we only know how to knock, how to give it the necessary blow."

-Swami Vivekananda

Acknowledgements

Four years of work, leading to a book, is obviously not an individual task. This book may not have been written without their help. I acknowledge my parents Gopalakrishna and Sharada in the first place for their constant inspiration and for reminding me to be a good human being before anything else. If this thesis has taken a good shape, I primarily credit it to my three inspiring advisors – Dr.Tanir Ozcelebi, Prof.dr.Johan Lukkien, and Prof.dr.Antonio Liotta – for being my backbone throughout. They gave me absolute freedom; enthusiastically motivated, encouraged and helped me head towards the right direction. I would like to thank Tanir very much for his mind-boggling discussions, patience, English edits and meetings without appointments. I would like to thank Prof.Johan immensely for helping me to improve technical clarity and depth in my works. I would like to thank Prof.Antonio extremely for his encouraging and motivating appreciations, and his huge assistance in shaping my work to be more presentable.

My special thanks to the reviewers – Prof.dr.Lu Liu, Prof.dr.Manohara Pai M. M., Prof.dr.Huib de Ridder and Prof.dr.ir.Berry Eggen – for spending their valuable time in reading my thesis and giving an invaluable feedback and encouraging words.

It was very difficult when moving alone to an unknown country for the first time. I would like to thank my dear friend Sunder and Barbara for their kind hospitality and good care, when I arrived in the Netherlands. Nowadays, working in collaborations is tedious and tiresome. However, I am fortunate and would like to thank Serge Offermans from Industrial Design for being an awesome and humble team partner, and Harm for providing me a pleasant stay in the department. Four years at the Intelligent Lighting Institute offered me a wonderful multi-dimensional view of the *World of Lights*. I would also like to thanks my SmaCS partners for wonderful meetings and hosting.

I thoroughly acknowledge Anjolein for the immense assistance provided in administrative activities, which would have been a Herculean task, otherwise. I would like to thank Sunder and Vinh very much for all the sense and no sense discussions we have had over the years. I also would like to thank Sachin, Tarun, Ali and Gourisankar, for all the fun-filled times. My earnest thanks to Ehsan for morning coffee talks, Ionut for late evening chats and Natasa for being a cool Computer Network teaching assistant colleague for two years. I would like to express my sincere gratitude to all my SAN colleagues for making this PhD journey more beautiful and memorable.

I would like to heartily thank my sister Nivedita and my pet Mothi, whose cheerful, energetic and funny conduct fades away all my tiredness and brings smile on my face always. Finally, I would like to thank everyone who are involved in the success of my work.

Summary

Light is the primary need for individuals to perform most of their daily activities comfortably. At times and places where natural lighting, i.e. daylight and moonlight, is not sufficient or desired (e.g. in a theater), people depend on artificial lighting in their environments to continue such activities. Obviously, enjoying the presence of light is a matter of visual perception, which may differ from person to person. In addition to such functional and aesthetic concerns, research has shown that light has an influence on circadian rhythms of humans and human physiology. Furthermore, light also plays an important role in the effects of an environment on people's mood and well-being, and possibly on their social behavior. Therefore, *good lighting* can improve the quality of our lives, emphasizing the importance of lighting design and control in living and working environments. This is not easy to achieve, as the desired lighting conditions per activity may vary across different users, different types of tasks undertaken and also based on many other things. The *intelligent* lighting aims at providing most suitable light setting to its users based on the observed context such as the activity performed and the time of the day.

In order to understand and evaluate how well the intelligent lighting system can support its users with varying interests, needs and preferences, one need to focus on specific scenarios. In this work, we focus on specific lighting scenarios that take place at a certain part of an office space, dubbed the *breakout* area. This is an area that the office employees may use to have informal meetings or for personal retreat. Our pilot intelligent lighting implementation contains numerous connected devices (sensors and luminaries) that communicate with each other. The challenge here is to design and develop a learning system that learns users' preference through contextual data (the values of relevant features) gathered from the breakout area either implicitly via sensors or explicitly from the users. Later, this knowledge is used to predict suitable lighting condi-

tions for new contexts.

In this direction, we explored supervised machine learning in the context of intelligent lighting. The experiments were performed using rule based prediction algorithms. To evaluate prediction algorithms, we need performance metrics that suits the nature of intelligent lighting. In applications such as intelligent lighting, more than one output may be acceptable for a given input scenario, due to the factors such as variability in human perception. This type of relationship between the input and output makes it difficult to analyze machine learning algorithms using commonly used performance metrics such as Classification Accuracy (CA). CA only measures whether a predicted output is right or not, whereas it is more important to determine whether the predicted output is relevant for the given context or not. In this direction, we introduce a new metric, the *Relevance Score* (RS) that is effective for the class of applications where user perception leads to non-deterministic input-output relationships. RS determines the extent by which a predicted output is relevant to the user's context and behaviors, taking into account the variability and bias that come with human perception factors. The experimental results showed that there is a *one-to-many* relationship between the input features (defines the state of an observed environment) and the output i.e. the lighting condition. This means that for a given input there is more than one acceptable output. In such applications, we find that CA is not an appropriate metric to analyze the performance of the prediction algorithms.

Supervised prediction algorithms are the parametric class of algorithms where the underlying distribution among the variables is assumed. In intelligent lighting, we cannot assume any particular distribution because they may change over time due to human related factors. Therefore, it is necessary to have a learning scheme that can learn continuously with every new incoming sample. To this end, we explore non-parametric class of algorithms such as instance-based learning (k-nearest neighbor) and online learning algorithms in the context of intelligent lighting. The experimental results using the RS metric, we find that even though the prediction algorithms are not tailored for the non-deterministic problems such as intelligent lighting, they perform continuously better as they observe more samples. From our study, we observe that instance-based algorithms provide maximum prediction performance compared to online learning algorithms as they extract the information available from the observed samples completely. This makes them the most preferred approach to realize intelligent lighting.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Life without light is difficult to imagine. Light is a source of life for all living beings on earth either directly or indirectly. Plants need light for photosynthesis i.e. the process of making their own food. Light is also necessary for plant growth, and for both metabolic processes and interaction with their surroundings [111]. Most animals need light to find their food, for navigation purpose and sometimes even for communication [96]. For humans, the most basic need of light is to perform their daily activities with ease. However, the use of light is not limited only to visualize things, but also in medical treatments, communication and beautification. It is estimated that around 0.8% of the world's population is affected by depression caused by inadequate exposure to daylight [61] and especially women [21]. Also, an interesting research shows that the suicide rates across the world are influenced by the duration of sunshine [127].

In this chapter, we discuss the importance of light on human beings and its influence on human health and physiology. We, then, motivate the need for intelligent lighting in indoor environments from the viewpoint of human perception and well-being. Subsequently, we formulate research questions that arise in the process of realizing intelligent lighting and introduce the thesis contributions. Finally, we provide an outline of this thesis.

1.1 Importance of Light on Human Health

Research has shown that light, not only has an impact on functional aspects of human beings but also has various physiological effects. The daylight stimulates the formation of vitamin D and controls biological rhythms [132]. A

biological process known as *circadian* rhythm that repeats at regular intervals has been observed in plants, animals and humans [71]. The circadian rhythm in human beings is responsible for synchronizing the body's internal clock to 24 hours, and light acts as a prime regulator [85]. If circadian rhythms do not synchronize with the workday rhythms, which happens due to an insufficient amount of light, people may suffer from sleep disorders, chronic fatigue and depression [39].

In early 2000s, neuroscientists discovered a third class of light-sensitive eye cells called *intrinsically photosensitive retinal ganglion cells* (ipRGCs) [17]. They make use of light to set the circadian clock. These ipRGCs contain a light-sensitive protein called *melanopsin* that is involved in short-term memory. Melan-opsins are highly sensitive to blue light [62], which is present much more in daylight than in incandescent bulbs. Appropriate light during the day helps the circadian cycle to be in sync, enabling us to be active during the day and sleep better during the night [40]. Moreover, effective light during the day increases the production of the melatonin hormones during the night. Melatonin is referred to as the chemical expression or hormone of darkness i.e. it is the body's own signal for darkness [20]. The increase in melatonin thus promotes sleep. People who get a good night's sleep have better capability to perform better at work during the day.

Light treatments are being used in addressing winter depression [41], a variant of the *Seasonal Affective Disorder* (SAD) [1]. Most people in the extreme northern and southern latitudes, during the winter live in biological darkness during the day, due to an insufficient amount of daylight. Studies have shown that exposure to blue-enriched light when symptoms first appear is enough to reduce SAD scores for the whole winter [62] [41].

Light also plays an important role in affecting our mood, health, well being [42], and also on our social behavior [126]. It helps in serotonin to be produced, a neuro-transmitter responsible for maintaining mood balance, a deficit of serotonin leads to depression [134]. The increase in serotonin results in happiness and well-being of an individual, thereby protecting from mental and physical disorders. More exposure to outdoor lighting and improving indoor lighting may help optimizing everyday social behavior and mood across seasons in people [1]. Several other applications of light on health aspects are described in [62] [71].

1.2 Motivation for Intelligent Lighting

Now that we realize the role of light on human health, let us understand the need for intelligent lighting. By *intelligent*, we mean that the lighting setup in a physical area adapts its behavior to meet the desired objectives such as satisfying users, power-saving or adapting based on environmental conditions like room temperature, weather and time. Several hundred millions of people work in indoor environments, depriving them from their daily doses of natural daylight (for example, office workers, miners, construction workers and night drivers). In these cases, people have excessive sleep complaints and disorders [84], which in turn influence their mental performance resulting in under productivity and increased production cost. People depend on artificial lighting when it is difficult to receive natural lighting (indoor employees) or when natural lighting is not desired (extreme sunny conditions). Studies have shown that the quality of light in offices is directly linked to the productivity of employees, decrease in accidents and increased level of mental performance [41]. Research has also shown stronger effect on the sleep quality and alertness of the office workers than on their mood and pleasure, when they are exposed to light daily of around 1000 lux [66] [120].

Recent studies have shown that dynamic lighting characterized by varying color temperature and illumination provides better satisfaction among the office workers than static lighting [33] [76]. Moreover, people find dynamic lighting (varying intensities over short periods) to be less tiring, interesting and stimulating [22]. This emphasizes the importance of lighting design and control in living and working environments. *Good quality artificial* lighting at the work place that is biologically effective and developed on the basis of scientific findings is essential [16]. This is not easy to achieve, because like in the case of food and clothing, an individual's lighting preference is influenced by several factors. Some of them may include user's circadian cycles [16], alertness and mood, time of the day [119], type of task undertaken and influence of weather. For example, the lighting preferences of two roommates called Tom and Harry for reading can differ dramatically. Tom may always prefer diffused white light from the ceiling, as opposed to Harry, who prefers reading using illumination from a directed desk lamp. Furthermore, their preferences may also depend on social and cultural aspects. For example, Tom, being a kind roommate, may as well prefer to use just the desk lamp at night when Harry is asleep.

Environmental color is also found to have significant impact on various aspects of human life [44]. For example, lighter colors are believed to be more pleasant, friendly and also to make life easier [59]. However, the choice of color

may vary across people, particularly in the way it can stimulate their mood and well-being. By implementing intelligent lighting, appropriate colored lighting may be provided based on the *observed context*. A context here may be referred to as the set of inputs such as activity performed, time of the day and external light influence that motivates a user to select a certain light setting over others. Advances in lighting technologies such as Light Emitting Diode (LED) have added benefits such as high efficiency, lower maintenance, intelligent control, safer technology and fast operation that helps in realizing intelligent lighting effectively [23]. Therefore, achieving a better and healthier life has motivated the development of intelligent lighting that aims at providing ambient lighting based on the observed context.

Intelligent lighting is a very challenging application that is driven by clear objectives. Intelligence can be interpreted in several ways such as adapting the lighting conditions continuously for changing environments or distributing the task of illuminating a certain area among different lighting fixtures. Therefore, the objectives should be clearly defined to identify what *intelligence* is. Once intelligence has been determined, the next challenge is to identify a suitable tool to implement intelligent lighting. Machine learning is usually used to realize intelligent behavior by learning from data or experience and providing relevant predictions [102]. There are several approaches of machine learning that can be adopted based on factors such as the objectives to be achieved and the very nature of data. However, selecting a machine learning model is not straightforward without experimenting, because there is no direct mapping from a specific problem to a machine learning model. Furthermore, it is important to select appropriate performance metrics, depending upon the nature of the application; there might otherwise be a good chance of selecting poor models and algorithms.

1.3 Research Problems

This thesis addresses the problems that arise in the process of implementing intelligent lighting. It focuses on the following research questions:

1. For a concept of intelligent lighting, how can we achieve it, using computational intelligence?
2. How can machine learning be used in realizing intelligent lighting?
3. How is the data collected and what does it say about intelligent lighting?

4. Are the existing performance metrics well suited to analyze whether or not a given machine learning approach is performing well or not?

5. What is the most suitable approach for implementing intelligent lighting?

1.4 Contributions

In this thesis, we contribute the following, while addressing the problems towards implementing intelligent lighting.

1.4.1 Learning frameworks

Intelligent lighting can be described through various perspectives and can be achieved using machine learning methods in several ways. In Chapter 2, we describe our perspective of intelligent lighting that we aim to achieve. Subsequently, we explore some of the existing intelligent lighting plans and implementations. We then discuss different types of learning frameworks such as supervised learning, online learning and reinforcement learning, illustrating how these may be used for intelligent lighting.

1.4.2 The unique nature of intelligent lighting

To make a better design choice about selecting an approach for implementing intelligent lighting, it is necessary to know about the quality of the data collected and the nature of the dataset. In Chapter 3, we describe the pilot setup used for collecting intelligent lighting data. Based on the pilot setup, we then discuss the input features that may have influence on the output light setting selected. Furthermore, we explain how the data was collected and cleaned to obtain a dataset that represents intelligent lighting. In Chapter 4, we briefly give an overview of the supervised prediction (learning) models that we use for experiments on the intelligent lighting dataset. We analyze the nature of the dataset through the performance of prediction algorithms using two performance metrics. We find that intelligent lighting has an interesting *non-deterministic one-to-many* relationship between the input and output that is uncommon in machine learning applications.

1.4.3 Machine learning for intelligent lighting

Once we know that the intelligent lighting dataset has a non-deterministic one-to-many relationship between the input and output, it is not straightforward to select a machine learning method that best suits intelligent lighting. To this end, in Chapter 4, we apply supervised learning models on the intelligent lighting dataset. In Chapter 7, we apply instance-based learning algorithm and in Chapter 8 we apply online learning algorithms to investigate which machine learning approach is best suited for applications such as intelligent lighting. This is done by comparing and analyzing the results from the experiments.

1.4.4 A performance metric for intelligent lighting

An efficient intelligent lighting system can be realized if the analyzed performance captures the nature of the dataset well. The most commonly used performance metric, i.e. *Classification Accuracy* (CA), is not appropriate for intelligent lighting, because a given input may have many possible acceptable outputs. Therefore, in Chapter 5, we devise a new performance metric, dubbed *Relevance Score* (RS), that is more appropriate for applications such as intelligent lighting. A key difference between CA and RS is that the RS may not reject a predicted light output entirely when it does not match the actual light output as desired by a user. The RS is computed using the probability information calculated from the entire dataset. Furthermore, we demonstrate the significance of RS through experiments on several public-domain datasets.

1.4.5 The missing-data problem

Missing data in a dataset occurs due to reasons such as malfunctioning or failure of sensors, improper implementation and insufficient interfaces. In our intelligent lighting application, the missing-data problem occurs when users do not use smart phone as an interface to communicate with the system. Missing data in intelligent lighting is an interesting problem because of the non-deterministic input-output relationships. This means that each input has a unique probability distribution of output class labels. Therefore, it is difficult to determine the input or a part of the input, just by knowing the output or by making use of the other part of the input. In such cases, deterministic missing-data treatment methods are not helpful. To this end, in Chapter 6, we explore several probability based approaches to address the missing-data problem. It is also important to know whether the missing-data treatment is effective or not (sometimes, the

improvement may occur by chance). Hence, we perform significance analysis to study the impact of missing-data treatment on the performance of prediction algorithms using a statistical test.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 discusses the main objective that we aim to achieve on implementing intelligent lighting. In this direction, the chapter also gives an overview of various possible machine learning approaches such as supervised learning and online learning. Chapter 3 describes the pilot implementation of intelligent lighting and discusses the data collection mechanism that is later used for our machine learning study. Chapter 4 studies the nature of the data collected from the breakout area, i.e. the relationship between the input and output through experiments and various metrics. Chapter 5 introduces a performance metric dubbed *Relevance Score* that is suitable for applications such as intelligent lighting that have a one-to-many input-output relationship. Chapter 6 discusses the missing-data problem in intelligent lighting and, subsequently, explores various probability based approaches to solve the problem. Chapter 7 studies an instance based learning algorithm *(k nearest neighbor)* with different values of k for intelligent lighting. Chapter 8 explores some of the state-of-the-art online learning models for implementing intelligent lighting. Chapter 9 concludes this thesis and proposes the directions towards enhancing various aspects of the current intelligent lighting implementation.

CHAPTER 2

Learning Frameworks for Intelligent Lighting

2.1 Introduction

What is *Intelligence*? Researchers have defined *intelligence* in several ways, such as the ability to deal with cognitive complexity [57] and goal-directed adaptive behavior [101]. Intelligence is also defined in terms of one's capacity for logic, abstract thought, understanding, self-awareness, communication, learning, emotional knowledge, memory, planning, creativity and problem solving [130]. A more detailed description of intelligence has been provided in [122]. Over the years, researchers have been trying to mimic some of these aspects of intelligence using machines to improve the quality of human life. Examples are many, ranging from heart sound analysis for symptom detection [112], fraud detection [50], face recognition [15] and autonomous driving [2].

Intelligent lighting is one such application where the objective is to learn and provide most suitable lighting conditions in an environment. Precisely, we define an intelligent lighting system as follows.

An intelligent lighting is a lighting infrastructure that autonomously learns and predicts a suitable lighting condition for every context in a physical environment to achieve a certain objective.

In intelligent lighting, learning happens through a feedback mechanism either explicitly, via dedicated interfaces such as smart phones, or implicitly by

Table 2.1: Evolution of Lighting Control

	Type of Lighting	User Intervention	Behavior	Example
1.	Traditional	Maximum	Users have full control over lights	On/Off Switch
2.	Autonomous	Nil	Users not needed for controlling lights	Sensor controlled
3.	Adaptive	Minimum	User preferences can be stored	Personalized lighting
4.	Intelligent	Minimum	Lighting systems learn user preferences	Breakout 404 [103]

monitoring through sensors or observing manual changes. The context, here, is defined as a state of the environment on which the *suitable* lighting depends. It can be determined by a *feature* or a *set of features* that are relevant to achieve the objective of intelligent lighting.

To have a better understanding of *intelligent* lighting, let us look into the history of lighting control. Table 2.1 shows the evolution of lighting control. Initially, lighting systems could be controlled only using switches and dimmers. With the rapid development of sensors and semiconductor technologies, lighting systems could be controlled using sensors without the need for switches. This has made it possible to give more emphasis on user satisfaction and saving energy by controlling the intensity [12] and illumination pattern of lights [118]. Revolution of memory devices enabled the storing of users' light preferences [79] or scenario-specific light settings [86] on the lighting systems, to provide customized/adaptive lighting. The development of data mining techniques and machine learning algorithms have made learning possible from the data or through interactions. Intelligent lighting is an improvement of adaptive lighting where a predefined light setting is not stored for a context but is learned through the system's interaction with the environment. This means that a lighting condition need not be preset for a given context and it may be adapted over a period of time based on the responses received from the environment.

In this chapter, we describe the objective of intelligent lighting. Next, we explain different machine learning approaches that can be used to realize this objective. Finally, we discuss the approaches that we use for implementing intelligent lighting in this thesis.

2.2 Objective

Intelligent lighting applications may have different objectives such as adapting lighting conditions until the users are satisfied, based on weather conditions or to optimize energy consumption. Our objective of intelligent lighting is to learn and predict a suitable lighting condition based on contextual information such as the user identity, activity and time of the day in an environment. For example, assume Mary is bored working at her desk for a long duration. She prefers to refresh herself by working on a creative design in the intelligent lighting room at 4.00 PM. She enters the intelligent lighting room with her smart phone and selects the type of activity e.g. *working alone*. Now, the intelligent lighting system knows that Mary would like to *work alone* at 4.00 PM and hence it predicts a *warm bright* light. Even though Mary is satisfied with the predicted lighting condition, she changes it to *cool bright* through available interfaces. The intelligent system learns Mary's preferences for that context. Mary revisits the intelligent lighting room after a week in a similar context. The system knows that Mary likes bright light in that context and hence provides either a *warm bright* or *cool bright* light. This scenario provides an idea about the nature of intelligence we seek to realize, where the system learns and predicts the most suitable lighting condition for a given context and improves over passage of time. From this example, we can also observe that the users have a key role in making the intelligent lighting system to perform well. By this we mean that the system learns quite well from a user who is consistent in terms of selecting output lighting conditions for a given context, whereas the learning becomes difficult when the user frequently prefers a different lighting condition under the same context.

2.3 Related Work

In this section, we discuss some of the existing works on intelligent lighting. Most of these lighting systems fall into the *adaptive* lighting category, although they are referred to by their authors as *intelligent*. The lighting conditions either adapt to changing contexts in a pre-defined manner or by adjusting their behavior to satisfy a certain objective function.

Miki et al. [93] proposed a lighting system to provide sufficient illumination at a desired location. The system is completely distributed and is in the form of a network that consists of fluorescent lights, controllers for each light and illumination sensors. The system uses a distributed optimization algorithm based on

the stochastic hillclimbing method [72] to control the lights. The autonomous lighting system is given the goal of minimizing the amount of power consumed using pre-determined value sets at each illuminance sensor. Each illuminance sensor detects and communicates the current illumination and target illumination to the network. Each lighting fixture controls luminance based on the illumination control algorithm using the information on the amount of power, current and target illumination of each sensor. The cycle repeats until the goal and constraints are satisfied. Here, each intelligent lighting fixture operates on its own, without acquiring information from other intelligent lighting fixtures or illumination sensors. This problem was removed later in the extended work using visible light technology [92].

Similar work has been proposed by Bhardwaj et al. [18] to provide various light control strategies based on user preferences. This work focuses on providing adaptive light emitting diode (LED) lighting rather than using conventional florescent and incandescent lights. The system is centralized, whereby the illumination sensors provide information about the current illumination levels and the light actuators provide information about the current light output levels for each fixture. Here, the adaptiveness is in distributing the task of providing a certain level of light illumination among the LED fixtures based on the designed illumination model. Also, for an activity, user preference for light illumination is assumed to be known. For many reasons, LEDs may be added or removed to/from a particular area. The role of intelligent lighting is to identify the change and actuate the light fixtures accordingly to provide the required illumination.

Singhvi et al. [118] proposed lighting systems that aimed at balancing user satisfaction and operational costs. The system is based on optimizing a utility function that combines the satisfaction of users and the expended energy. This function depends on the location of users and the corresponding light fixtures that are associated in illuminating the space. The implementation approach is based on a decision theoretic formulation of the control task. This work also extends to optimally exploit external light sources for additional energy savings, a process called daylight harvesting. The system implementation is evaluated through a proof-of-concept test bed using MICA2 sensor nodes and dimmable lamps.

Magielse et al. [86] presented an adaptive lighting environment in an office meeting room scenario where multiple luminaries adapt themselves to emit light based on the user's activity and context. This work adopts an interdisciplinary approach to achieve adaptive lighting environment in the context of a room in an office space. This approach integrates the development of user

interfaces and interaction, the design of a network architecture, as well as the design of activity recognition models to provide adaptive lighting environment. The hardware comprises of light tiles on the ceiling of the room, sensors deployed to detect presence, pressure etc. Based on the activity identified using the sensor data gathered at a central station, one of the pre-defined scenarios is selected and the light environment is provided accordingly.

Park et al. [106] presented a lighting control system for entertainment and media production. The system is named *Illuminator* and uses a multi-modal and high fidelity light sensor module. The Illuminator is a tool-set to characterize the illumination profile of a deployed set of fixed position lights. It generate desired lighting effects for moving targets such as actors and scenic elements. The user constraints are expressed in a formal language that represents the requirements for lighting effects and includes the desired light intensities in the field and/or a high-level description of lighting conditions. The Illuminator computes optimal light settings at run-time to achieve a user-specified light profile, using an optimization framework based on a genetic algorithm. The experiments demonstrated that the Illuminator can handle various high-level user requirements and generate an optimal light actuation profile.

Lighting design is very important in game environments for directing attention, establishing good action visibility, evoking emotions, setting atmosphere, and providing depth [43]. Lighting design in game environments has been static and manual where the designers set up the positions, angles, and colors for each light in a scene. The static lighting design does not serve desired effects as the game environments are dynamic and change unpredictably in real time due to user interaction. El-Nasr [43] presented an intelligent lighting system that automatically sets and adjusts scene lighting in real time. The system is named Expressive Lighting Engine (ELE) and allows to achieve effects such as evoking emotions, directing visual focus, and providing visibility and depth. ELE operates as a separate system that interacts with game/graphics engines through a standard interface. ELE uses constraint non-linear optimization that optimizes several parameters such as visibility, depth and tension to select the best lighting configuration and achieve the desired visual design goals given their priorities and the current situation. ELE helps in accelerating the development process by eliminating the time-consuming process of static lighting design.

In general, we observe that the main objective of these intelligent lighting implementations is to provide necessary illumination at a desired location based on some utility function. The primary task here is to distribute the task of providing necessary illumination among different light sources. Users' preferences are either assumed or obtained from users at runtime. However, all these imple-

mentations use sensor-based technology to sense the level of illumination over an area and use different methods, objective functions, models and different kind of light sources to achieve the mentioned objective. Moreover, these works focused mainly on achieving certain level of illumination while there was no focus on creating the ambient environment using different colored lightings. In the last two implementations, intelligent lighting is used for a different kind of applications, in media production and gaming environments. In this thesis, we aim at learning the users' preferences by observing the environment, interacting with the users or through explicit user's feedbacks. We focus on using machine learning models to learn user preferences and provide the suitable colored lighting conditions for an observed context.

2.4 Machine Learning

Machine Learning (ML) is the science of developing systems that can learn from data and act accordingly without programming the behavior of the application or specifying thresholds explicitly. The most widely accepted definition from Mitchell [94] is as follows.

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

In intelligent lighting, ML is used to determine the relationship between the context and the output lighting conditions. The input-output relationship is then used to predict suitable lighting conditions for new inputs. Applying Mitchell's definition to intelligent lighting, we have,

- Task T: predicting suitable lighting condition for a given context,
- Performance Measure P: quality of the predicted lighting conditions,
- Experience E: a sequence of observed contexts and the user response.

In this section, we discuss different ML approaches to that may be used in the context of intelligent lighting.

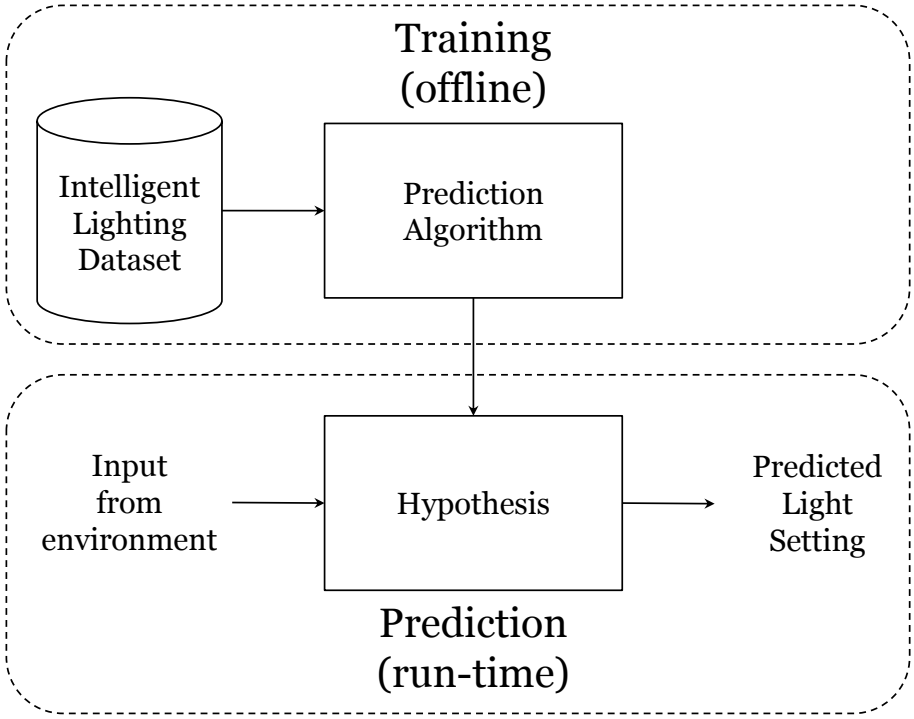


Figure 2.1: Supervised learning framework for intelligent lighting

2.4.1 Supervised learning

Supervised learning is the ML task of *identifying the relationship between input and output* from labeled training data [31]. In supervised learning, a prediction algorithm h is trained using a dataset that consist of a set of training examples. Each sample is a pair consisting of values for input features (context) and the corresponding class (output). During training, h analyzes the samples in the dataset and generates hypotheses. A hypothesis represents the relationship between the input and output that can take the form of functions, rules or trees [131]. The hypothesis that provides best prediction performance is then used to make predictions for future inputs.

Figure 2.1 shows the framework for supervised learning for intelligent lighting. An intelligent lighting dataset consists of samples i.e. input and output pairs collected from a pilot implementation. The input/context is a vector of

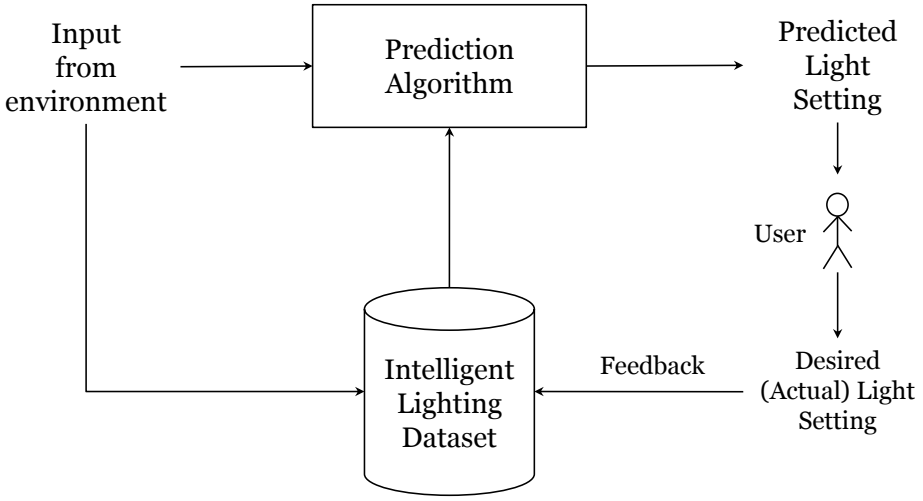


Figure 2.2: Instance-based learning framework for intelligent lighting

features that may influence users' preference for particular lighting conditions. The output is the preferred lighting condition by the user for a given context. A supervised prediction algorithm h, is trained on the dataset to generate a hypothesis, which is used to predict a suitable lighting condition for new incoming inputs.

2.4.2 Instance based learning

In this form of learning, the input-output relationship is not deduced when the training samples are provided whereas it is deduced when a new input arrives that needs to be predicted. In other words, instance based learning algorithms do not generate any useful representations from the observed inputs [4]. To assign a class label for a new instance, its relationship to the samples that were already encountered is determined. The main advantage of this kind of learning is that the target function is estimated every time for a new instance to be predicted rather than estimating once [94]. This makes instance-based learning useful when the input-output relationship changes over time.

Figure 2.2 gives the framework for instance-based learning for intelligent lighting. For example, let us consider the popular instance-based algorithm, *k-Nearest Neighbor* (kNN). In kNN when a new input arrives, it computes the dis-

tance of the input to all other observed inputs in the intelligent lighting dataset. For a selected value of k, k samples that are the closest in distance to the input are selected. A lighting condition that appeared maximum number of times among the k selected samples is selected for prediction. A user may select a lighting condition that suits them, if the user is not satisfied with the predicted lighting condition. The selected lighting condition along with the input is stored in the dataset so that the predictions can be improved for new inputs.

2.4.3 Online learning

Online learning is a ML approach that learns from one sample at a time. In online learning, the input arrives as a sequence of samples in contrast to the supervised learning where the input is fed as a batch [5]. As an input arrives, the online learning model should predict an output. Immediately after the prediction is made, the actual output is made available to the model. This information can be used as a feedback to update the prediction hypothesis used by the model. A key advantage of online learning over supervised learning is that the model adapts itself with every incoming new sample. Unlike instance-based learning, online learning models are relevant in applications where huge amounts of data needs to be stored for training purposes.

Figure 2.3 shows the framework for online learning for intelligent lighting. The input from a pilot implementation in the form of a feature vector is fed to the online prediction algorithm. The prediction algorithm predicts a suitable lighting condition for that input. A user may select a lighting condition that better suits them, if the user is not satisfied with the predicted lighting condition. The lighting condition preferred by the user (actual output) is then revealed to the prediction algorithm in the form of feedback. The prediction algorithm will use this feedback to update its hypothesis, in case if the predicted lighting condition is not the same as the user preferred lighting condition.

2.4.4 Reinforcement learning

In supervised learning, learning happens through examples provided by a knowledgeable external supervisor. However, it is not sufficient for learning from interactions. Reinforcement learning is a ML task concerned with how an *agent* should take *actions* in an *environment* so as to maximize some notion of cumulative *reward* [123]. An *agent* is an autonomous entity that monitors and acts upon an environment [116]. In reinforcement learning, inputs (observations

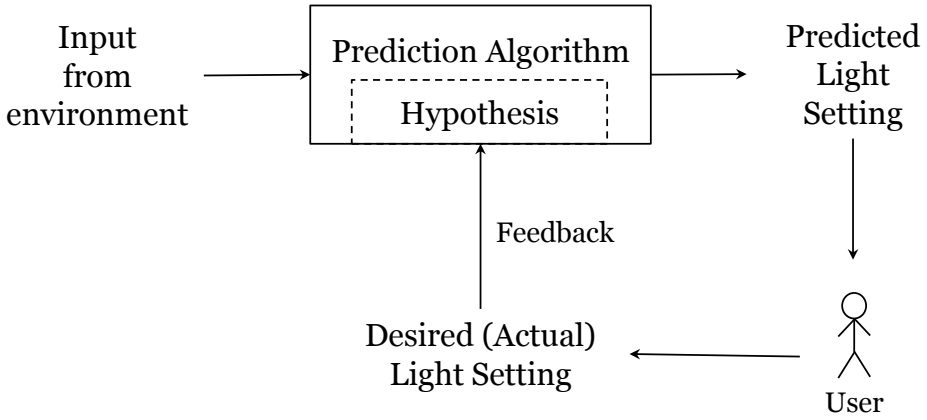


Figure 2.3: Online learning framework for intelligent lighting

from the environment) are in the form of states and the learning entity acts as a decision making agent [73]. It takes actions based on the state of the environment and in response receives a feedback in the form of reward/penalty. In reinforcement learning, there is no training phase to teach the agent about what actions to take or the most preferred output for an observed state. After a set of trial and error runs, it should learn the best policy, which is the sequence of actions that maximize the total reward [73]. Hence, it is also called *learning with a critic*.

Figure 2.4 shows the framework for reinforcement learning for intelligent lighting. The state of environment is provided by means of inputs. The decision making agent then prompts an action in the form of a lighting condition. The user in turn gives a feedback (reward) to the agent such as like or dislike. The agent will use the feedback to improve its state-action mappings and then predicts a new lighting condition. The cycle continues until a suitable lighting condition (with maximum reward) is selected. In the long run, the reinforcement agent tries to achieve maximum reward from limited interactions with the users.

2.5 Discussion

To summarize, a supervised learning algorithm is trained using a number of training samples that in turn generate a hypothesis to predict future inputs.

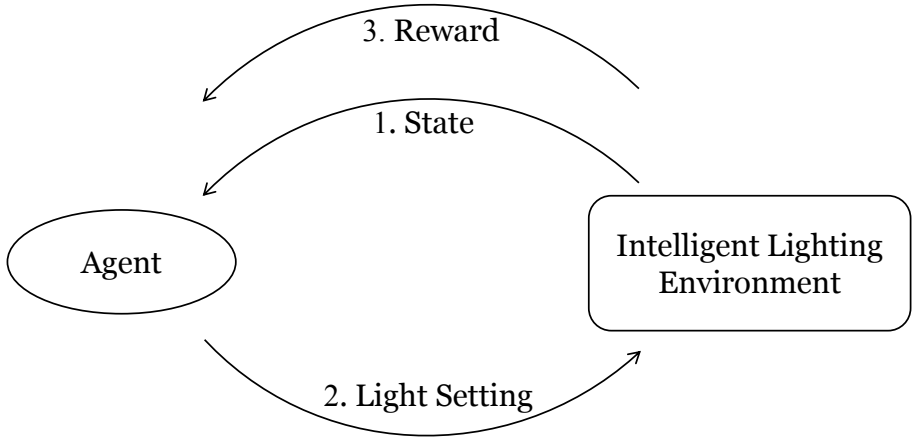


Figure 2.4: Reinforcement learning framework for intelligent lighting

An instance-based learning algorithm waits until a new input arrives and then predicts an output based on its relationship with other observed samples. The instance-based learning algorithms do not generate any hypothesis to have a meaningful representation of the input and output relationship. Unlike supervised learning, an online learning algorithm learns from every incoming sample by modifying the hypothesis continuously. Unlike instance-based learning, an online learning algorithm does not store any sample for future reference, whereas the summary of the observed samples are stored in hypothesis.

In this thesis, we consider applying supervised learning for implementing intelligent lighting in Chapter 4, instance-based learning in Chapter 7 and online learning in Chapter 8. We do not consider reinforcement learning for implementing intelligent lighting. The reasons are as follows. Reinforcement learning uses explore and exploit policy where the agent explores all possible outputs until it reaches an optimal solution that maximizes the objective function. However, our objective of intelligent lighting is that the users must be able to select their preferred lighting condition when they are not satisfied with the predicted light condition. In reinforcement learning, the users are required to respond to every predicted lighting condition until they are satisfied. In the extreme case, because of the exploring aspect of reinforcement learning, there may be a long cycle of predicting lighting conditions and user responding. This situation occurs quite often, which is not preferred. In reinforcement learning applications, an action changes the state of the system and the state determines the action

that needs to be taken. In our case of intelligent lighting, an action i.e. predicting a lighting condition does not change the state of the system, i.e. the context. Formally, in reinforcement lighting, $action = f(state)$ and $state = f(action)$, whereas in intelligent lighting, $action = f(state)$ and $state \neq f(action)$.

CHAPTER 3

Intelligent Lighting Setup and Dataset

3.1 Introduction

Useful knowledge such as the features influencing the users' lighting preference and relationship among these features, helps us to have a better understanding about intelligent lighting. This would enable to make better design choices when implementing intelligent lighting and has motivated us to perform a pilot study to gain insights on the nature of intelligent lighting. Dean et al. [34] presented a detailed outline on how to perform a study or experiment, that is briefed in order as follows.

- Define the objective of the study
- Identify the features (variables that influence the study)
- Specify the experimental procedures
- Run the pilot or perform the experiment
- Perform data analysis on the collected data
- Review and revise the design choices

In our case, the objective of intelligent lighting is to provide a suitable lighting condition for the users, based on the observed context. Here, the observed

context is a vector of feature values that may influence the users' preferences for a particular lighting condition. Essentially, these features enable us to identify what data should be collected in order to perform the data analysis. The features to be selected depend primarily on the architecture of the physical area where the intelligent lighting will be installed.

Once the relevant features have been identified, it is important to design the data collection mechanism to gather data. The mechanism must be devised carefully to reduce the error induced during the measurement process. Any algorithm developed based on data is only as good as the data used. Collecting and managing data can be done in various ways. Data can be collected through observations, experiments, interviews and surveys. In intelligent lighting, we seek to study the input-output relationship, which in turn will be helpful to select a suitable machine learning approach.

Data collection through observations is most suitable for data analysis. According to Young [133], *"Observation may be defined as systematic viewing, coupled with consideration of seen phenomenon. The purpose and aim of observation is to discover significant mutual relations between spontaneously occurring events"*. It is easier to perform data analysis to discover the input-output relationship, if the data is organized in the form of samples consisting of input-output pairs. The input entries are the values of the features that are considered to influence the users' choices, given a certain lighting condition. The output entry is the actual lighting condition desired by the user. The output space may also be multidimensional that indicate characteristics of lighting conditions such as RGB (Red-Green-Blue) composition and intensity or HSB (Hue-Saturation-Brightness) composition. The values for the dataset entries can be gathered through observations from the physical area implicitly, by means of sensors that are deployed in the area, and explicitly, by direct input from the users through well-defined user interfaces.

In this chapter, we describe the layout of our pilot implementation for intelligent lighting system i.e. the *breakout* area. Furthermore, we discuss the network and physical architecture for intelligent lighting. Based on the layout, we list the input features selected that help to predict a suitable lighting condition. Next, we provide the description of the dataset such as the user-sample distribution and output distribution. Finally, the data collection mechanism is explained.

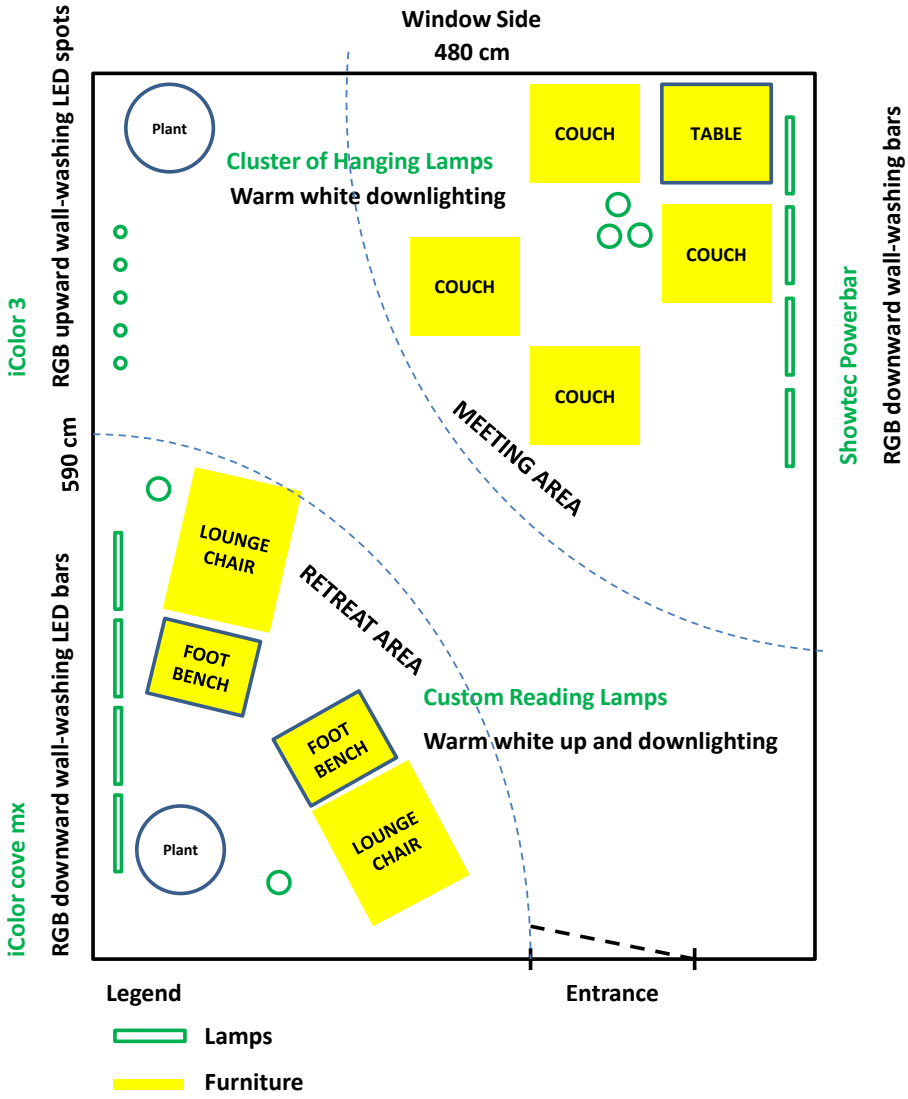
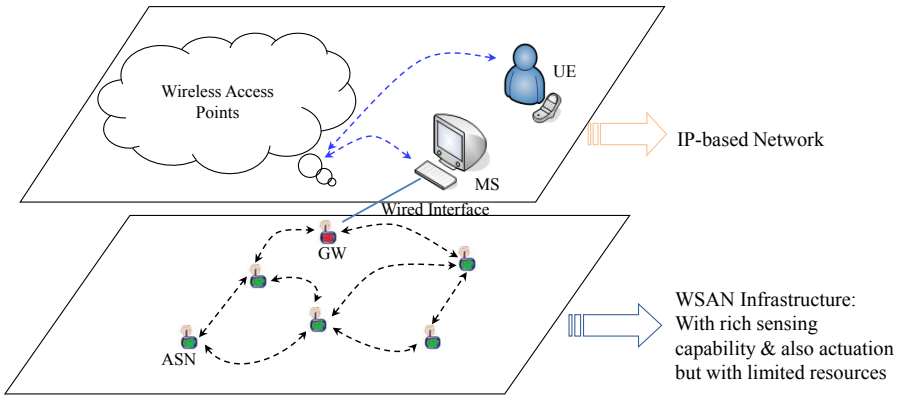


Figure 3.1: Layout of the breakout area



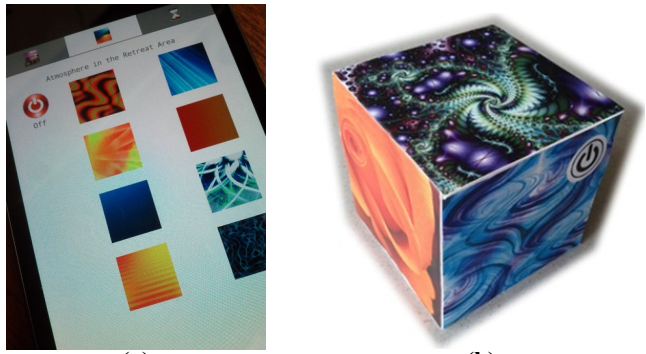
GW – Gateway Node
ASN – Actuator/Sensor Node
MS – Management System
UE – User Equipment
WSAN – Wireless Sensor and Actuator Network

Figure 3.2: Network architecture for intelligent lighting

3.2 Pilot Setup

We need a pilot implementation to understand how well intelligent lighting can support its users with varying interests, needs and preferences. In this section, we discuss the pilot implementation of the intelligent lighting system from where the data have been collected. We focus on specific lighting conditions that take place at a certain part of an office space, dubbed the *breakout* area. This is an area that the office employees may use either to have informal meetings or for personal retreat. The pilot provides a dynamic lighting platform that can be used in many forms such as for task lighting, for atmospheric lighting or for informative lighting. We name the dataset collected from the breakout area as *Breakout* dataset.

Figure 3.1 shows the layout of the breakout area. Opposite to the entrance is a wall with windows and blinds, which allows for controlling the external light influence. The area is divided into two spaces dedicated to different purposes, the *meeting* area for informal meetings and the *retreat* area for personal retreat and relaxation. However, the users of the breakout area are not restricted to use a specific area for a specific activity. For example, a user may choose to use either the meeting area or the retreat area for relaxation. The lighting system



(a) **(b)**

Figure 3.3: Interfaces to communicate with the intelligent lighting system
a. Smart phone showing 8-preset output lighting conditions,
b. Scene selection dice with preset lighting conditions on each side
(Image courtesy: Offermans et al. [104] [105])

in the breakout area contains two types of lights, colored wall-wash lights for creating an atmosphere and white *down-lights* [104] for illuminating areas of user tasks. The sensor system for monitoring the breakout area contains Passive Infra Red (PIR) sensors for monitoring movements, sound pressure sensors for monitoring sound volume intensity and light sensors for measuring external light influence.

3.2.1 Network Architecture

Fig. 3.2 shows the network architecture of the intelligent lighting system. The network architecture gives the hierarchical view of the communication infrastructure wherein the hierarchy is based on the resource capabilities of the components in the breakout area, such as communication, computation and memory as well as the standard of communication used. This perspective provides the underlying communication interactions among the components to accomplish the objective of providing suitable lighting conditions. The wireless sensor and actuator nodes (WSAN) in the breakout area are connected to each other wirelessly, forming the base of the network infrastructure. The lower layer in Fig. 3.2 depicts the Wireless Sensor and Actuator Network (WSAN) infrastructure. One or more sensors (e.g. motion sensors, sound sensors) are attached

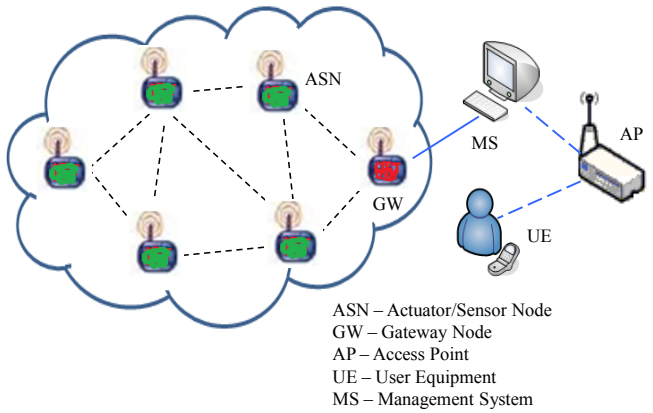


Figure 3.4: Physical architecture for intelligent lighting

to each sensor node to continuously monitor the presence and activity of the user. The actuator nodes (AN) are connected to the light sources which provide varied lighting conditions to the users based on their preferences. These nodes communicate using the IEEE 802.15.4 standard and are limited in communication, computation and memory capabilities. We used the custom designed nodes based on the Arduino platform [9]. These nodes contain an Atmel 328 microprocessor [10] for processing and an xBee Series 2.5 Wireless radio [67] for communication. The nodes allow to connect 6 analog sensors and 6 Pulse Width Modulation (PWM) controlled lamps [103]. Furthermore, they can function as a Digital Multiplex (DMX) controller that can be used to control commercially available lighting sources [103]. The users may control the lighting conditions in the breakout area through user equipments (UE) such as smart phones as shown in Fig. 3.3a or through a dedicated interface such as the scene selection cube as shown in Fig. 3.3b. While, the scene selection cube does not affect privacy for an user, it limits the output space to six lighting conditions. Also, in such cases, it would be more difficult to predict customized lighting condition for that user. The UEs use the IEEE 802.11 standard for communication. Therefore, the Internet Protocol (IP) based network forms the top layer which contains the UEs using the services of the area via an access point (AP). The interoperability issue of communication between the UEs and the WSANs is handled by using gateway (GW) nodes. A GW is a ASN but without any sensor or actuators attached to it. The GW is connected to management system (MS) using a wired interface. The MS communicates the desired lighting condition

(a)

(b)

Figure 3.5: Pictures of the breakout area where the intelligent system lighting is installed; a. Meeting area, b. Retreat area
(Image courtesy: Offermans et al. [104])

by the user to the GW node which in turn communicates this information to the ASN infrastructure.

Table 3.1: List of input features considered, that influences users' preference for lighting conditions

	Feature	Type of the feature	Possible Values
1.	User-Identity (UID)	Categorical	U1, U2, U3, U4, ...
2.	Type of Activity (ToA)	Categorical	Active_Group, Active_Alone, Relax_Group, Relax_Alone
3.	Area of Activity (AoA)	Categorical	Meeting, Retreat
4.	Intensity of Activity (IoA) in the other subarea	Categorical	$0, 1, 2, \ldots, 10$
5.	Time of the Day (ToD)	Numeric	$\in [0, 24)$, e.g. 10.5 for 10:30am
6.	External Light Influence (ExLI)	Categorical	VeryHigh, High, Low, VeryLow

3.2.2 Physical Architecture

Fig. 3.4 shows the physical architecture for intelligent lighting. Fig. 3.5 shows the photographs from the breakout area, where the intelligent lighting system is installed. The WSANs, GW, AP, UE and management system (MS) form the basic infrastructure of the intelligent lighting system. MS is a multipurpose system whose functionalities are to acquire the contextual data, predict suitable lighting conditions, monitoring and management of the system. It is a high capacity system that can be used for administrative purpose such as providing authenticated access for the users to use intelligent lighting. The users can use the service of intelligent lighting through UEs. The communication to the WSAN infrastructure is then established through the AP. The sensor nodes gather contextual data and it communicated via GW to the MS. The MS is a device such as a laptop or personal computer (with 2GHz processor, 2GB RAM, 30GB hard disk, 32-bit operating system), which is used to make predictions based on the observed context. It is also used to store the relevant information from the breakout area such as the sensor data, context and the usersŠ preferences. In our study, we collected around 4MB of data from the breakout area. The predicted lighting conditions or the usersŠ preferences are communicated to the actuator nodes via the GW node. The actuator nodes then actuate the relevant light sources to provide the predicted lighting condition.

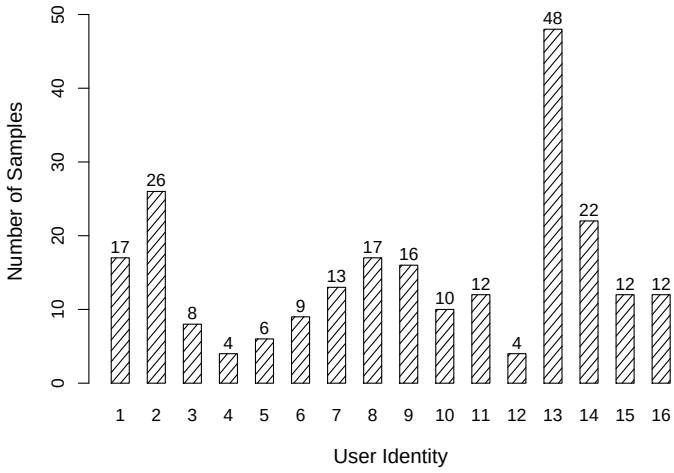


Figure 3.6: The distribution of samples for the users in the *Breakout* dataset with 236 samples

3.3 Breakout Dataset

The Breakout dataset for intelligent lighting consists of 236 samples collected as discussed in Section 3.4. The dataset does not have any missing data. We select six input features that may influence a user's choice in selecting one of the pre-defined light settings for a given context as summarized in Table 3.1. These features are selected based on the layout of the breakout area. For example, the feature AoA is selected because of the two different spaces. Also, selecting the feature ExLI makes no sense, if the breakout area is located indoor without windows. Figure 3.6 shows the number of users and the numbers of data samples collected per user. We consider eight output lighting conditions to support users' activities as shown in Fig. 3.7. The class distribution of these eight lighting conditions over the 236 samples is presented in Fig. 3.8. An example input-output pair in the Breakout dataset is as follows.

$$\underbrace{\texttt{U3 MEETING MA 4 10.50 High}}_{Input}\ \underbrace{\texttt{CB}}_{Output}$$

The sample denotes that a user with id U3 (UID=U3) chooses to use the break-

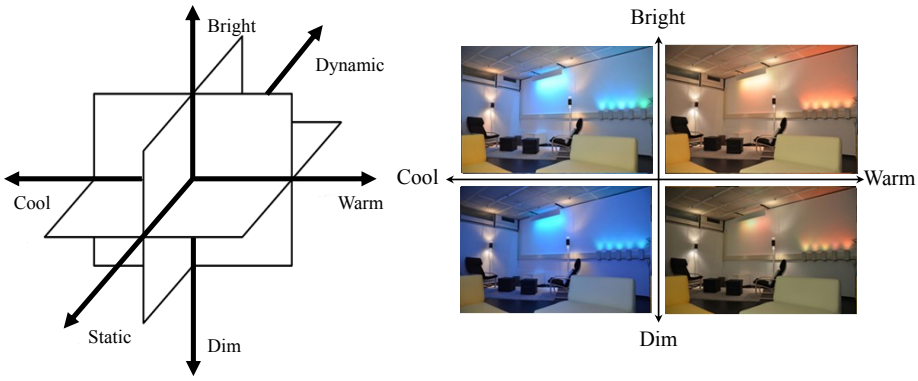


Figure 3.7: Possible output lighting conditions
a. 3-dimensional representation of output showing 8 different possible output lighting conditions,
b. 4 output lighting conditions from the pilot setup (static and dynamic lighting conditions can only be experienced in the breakout area)
(Image courtesy: Offermans et al. [104])

out area for a meeting (ToA=Meeting) at half past ten (ToD=10.5) at the meeting area (AoA=MA) when the external light influence is high (ExLI=High) and some level of activity happening in the retreat area (IoU=4). With this context, the user G prefers cool and bright (CB) output lighting condition. The complete Breakout dataset is presented in the APPENDIX.

3.4 Data Collection

Among the mentioned features in Table 3.1, AoA, ToD, IoA and ExLI are gathered implicitly from the breakout area through sensor monitoring. Both the sound and PIR sensor values are read every 50 milliseconds and the average of 20 readings is stored in a queue of size 60. This makes the queue to contain the data for one minute. The average value of the queue is used as a measurement of motion/loudness. This approach reduces false readings created when people pause during conversation or when people sit still for a moment. However, the light sensors in the breakout area do not use queue to store the values, which means that the change is immediately noticed. The features UID and ToA are

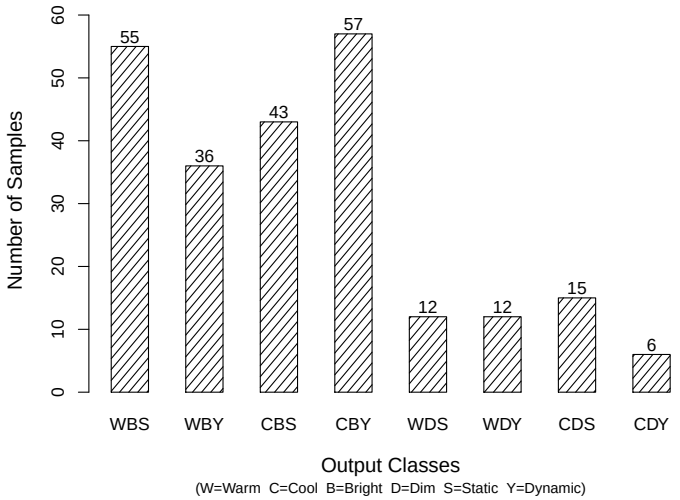


Figure 3.8: The distribution of output classes in the *Breakout* dataset

acquired explicitly from the users via the breakout application installed on their smart phones. If a user does not use a smart phone as the interface to communicate with the intelligent lighting system, the missing data problem occurs, which we discuss in detail in Chapter 6.

The data samples for the breakout dataset are collected using two methods. In the first method, we created various contexts in the breakout area with different ExLI, IoU values, in which the participants are asked to select a lighting condition that they prefer, considering the activities listed in Table 3.1. In the second method, the participants are allowed to use the breakout area on-demand for six weeks. During this six-week period, all interactions of the users with the system (i.e. activities and selected light settings) as well as the sensor readings are logged.

In order to learn users' preferences of light settings in a particular context, collected data samples should contain the values for features UID and ToA. On the other hand, when there are no users in the breakout area, then the data samples collected contain no entries for these features, as they are collected explicitly from the users. The feature values for AoA and the selected output lighting condition are also empty. These samples do not form the context as

most of the feature values in a sample are empty; thereby need to be removed from the dataset. Moreover, when users are not satisfied with the predicted lighting conditions, they explore other lighting conditions that suit them. This data is not necessary to understand users' preferences for a context. Therefore, data cleaning is performed on the Breakout dataset in two steps, to remove unnecessary samples. Firstly, those samples that do not contain feature values for UID, ToA and AoA are filtered out. Secondly, those samples that belong to users' free explorations of different light settings for a particular context are removed.

3.5 Conclusion

Intelligent lighting can be realized efficiently by having useful insights such as the relationship among features. This has motivated us to gather data through a pilot implementation and perform data analysis. In this direction, we have presented the pilot setup of our intelligent lighting. Subsequently, we discussed the network and physical architecture of our implementation. Furthermore, we described the dataset collected dubbed *Breakout* dataset and explained the data collection mechanism. We conclude that the layout of the physical area where the intelligent lighting system should be installed, is very important to identify the right features for data collection.

CHAPTER 4

Statistical Inference for Intelligent Lighting

4.1 Introduction

According to Diamond [36], *"The purpose of an experiment is to better understand the real world, not to understand the experimental data"*. Mere data collection does not help in making good design choices for implementing an application. The design process for implementing intelligent applications benefits from insights about the data collected and a deep understanding of the relations among its variables. Data analysis refers to the process of acquiring information from the collected data and deriving conclusions that help making design choices for implementing an application. Like diagnosing the problem of patients is important for doctors before providing treatment, data analysis is necessary for researchers to acquire meaningful information about the nature of the application to be implemented.

Many intelligent applications involve collecting data from different sources, organizing them and then performing data analysis. Data analysis helps to understand the relationship between variables in the data, evaluate and compare basic design configurations, determining and selecting design parameters and algorithms that may impact the application's performance [95]. Data analysis depends on the type of the data collected, either quantitatively or qualitatively [113]. Quantitative data can be analyzed using statistical operations such as frequency distributions, central tendency (using mean, median and mode), correlation and regression. Analyzing qualitative data involves examining the

data collected through surveys, interviews and observations. Machine learning is a data analysis technique that can be used to discover knowledge from the data for predictive purposes. To design intelligent lighting, it is necessary to understand the significance of each input feature and how the input features and output light settings are related. This helps to either add more features or remove insignificant features to improve the efficiency of intelligent lighting.

In this chapter, we describe the prediction algorithms used to analyze the Breakout dataset. We briefly introduce the performance metrics used to evaluate the prediction algorithms. We apply these prediction algorithms to investigate the nature of intelligent lighting and present the insights gained, which enables to make better design choices in implementing intelligent lighting. Furthermore, we analyze the significance of input features in determining the output lighting conditions.

4.2 Prediction Algorithms

Prediction algorithms can be categorized based on how they operate on the labeled data (data with correct outputs) to provide predictions. Some categories are supervised, instance-based and online prediction algorithms. In this chapter, we apply supervised prediction algorithms for intelligent lighting. Supervised prediction algorithms can be broadly classified into categories based on how they represent the relationship between input and output. These relationships may be expressed in the form of models (Neural Networks [71], Support Vector Machines [26]), rules (DecisionTable [77]) or trees (C4.5 [110]). In this thesis, we use rule-based prediction algorithms for experiments on intelligent lighting. The reasons are as follows. Rules are the clearest, most explored and best understood form of knowledge representation [49]. Generally, analytic methods are not guaranteed to provide good results with fewer samples than rule-based methods [83]. Also, the rule-based algorithms, even with very simple rules, have achieved high accuracy with many datasets [64]. In contrast to tree-based models, rule-based models allow overlapping rules i.e. more than one input can be covered using one rule that results in smaller rule set [49].

4.2.1 Conjunctive Rule

The ConjunctiveRule algorithm employs a single conjunctive rule learner that can predict, for any type of output class labels, such as numeric or categorical [131]. Conjunctive rules are learned by finding all the common features

shared by an output class label. This algorithm generates a single rule which is a combination of antecedents (IF part) and a consequent (class value). The information gain (categorical class) or variance reduction (numerical class) are computed for each antecedent [25]. An antecedent is subsequently selected by pruning the generated rules using *Reduced Error Pruning (REP)*. In the pruning phase, weighted average of the accuracy rates is used for classification while the weighted average of the mean-squared errors is used for regression, on the pruning data. This process may leave uncovered input instances, which are then assigned default class labels. The uncovered test inputs are predicted using the default class labels of the uncovered training inputs. The conjunctive rule states that all the specified features must be present to belong to a particular output class.

4.2.2 DecisionTable

The DecisionTable algorithm generates a simple decision table majority classifier [77]. This algorithm uses the *wrapper* method to evaluate different subsets of features to include in the table. The wrapper method searches for an optimal feature subset for a given prediction algorithm and the dataset [78]. The quality of a feature subset is measured by developing and evaluating a prediction algorithm. The feature subsets may be generated using *BestFirst* search, *Exhaustive* search, or other search techniques. Thus, a *decision table* is generated with the same number of features as in the original dataset. The algorithm reduces the problem of over-fitting by eliminating the features that have less significance and thus generates a condensed decision table. The output class value of a new data instance is predicted by matching the instance values in the decision table.

4.2.3 JRip

JRip [24] employs a propositional rule learner, *Repeated Incremental Pruning to Produce Error Reduction (RIPPER)*. It is based on association rules with *REP*. The algorithm comprises three phases. Initially, the building phase involves two subphases: the grow phase and the prune phase create a rule-set on the training data by splitting the data into growing set and pruning set. In the grow phase, the algorithm processes the growing set to generate new rule set by adding rules repeatedly to an empty set. In the prune phase, the rule generated from grow phase is incrementally pruned on the pruning set. Secondly, global optimization techniques such as post-pass massages are used to process the rule-set to reduce

the size and improve its performance on the training data. Finally, the problem of over-fitting is reduced by processing the rule-set with a combination of cross-validation and minimum-description length techniques.

4.2.4 Nearest Neighbor with generalization

Nearest Neighbor with generalization is commonly known as *NNge* [87]. NNge is a variant of the popular instance-based learning algorithm, k-Nearest neighbor (k-NN) [94]. NNge makes use of non-nested *generalized exemplars*. Exemplar refers to an instance that has already been used for classification. Generalized exemplars are rectangular regions of input space called *hyperrectangles* [131]. These hyperrectangles can be viewed as *if-then* rules. Instance-based learning does not produce any knowledgeable representations, whereas NNge produces a set of rules that can be compared with other rule-based prediction models. NNge employs a learning strategy where search for hypothesis proceeds from specific to general rather than general to specific.

4.2.5 PART

The PART algorithm is a separate-and-conquer rule learner [48]. The algorithm combines the power of decision trees with the RIPPER rule learning, to generate rules. Initially the algorithm repeatedly generates partial decision trees using heuristics such as pruning. A set of rules from these decision trees are extracted. These rules are ordered and are known as *decision lists*. The prediction for a new data instance is done by comparing the data instance with each rule in order. The class value of the first matching rule is assigned to the instance or the default class value is applied if no rule is matched.

4.2.6 Ridor

The Ridor algorithm implements *RIpple-DOwn Rule* learner [51]. Firstly, the algorithm generates a default rule, and then learns exceptions that do not comply with the default rule. Each exception from the default rule has many more exceptions (say sub-exceptions) that takes the form of a tree. The exceptions are learnt using incremental reduced-error pruning that includes iterating to find the best sub-exceptions for each exception. Thus the exceptions are a set of rules that finds a class value other than the class value in the default rule. The prediction for a new instance is done by choosing the best exception with minimum error rate.

4.3 Performance Metrics

It is important to identify appropriate performance metrics to evaluate the performance of prediction algorithms, which will in turn help investigating whether the desired objectives are achieved. In this section, we briefly discuss two metrics; *Classification Accuracy* (CA) and *Relevance Score* (RS) [55] that are used to evaluate the performance of prediction algorithms scrutinized herein the context of intelligent lighting. For a given context, CA measures whether the predicted output is right or not. Formally, if $x \in X$ is the observed context where X is the n-dimensional context space, y_P is the predicted output and y_A is the actual output, then $CA = f(y_P, y_A)$. This means that CA is an objective metric that does not depend on the context whereas only on the predicted and actual output. Intelligent lighting is an application that involves factors such as human perception and cognition. The influence of these factors makes users to select different output lighting conditions for a same given context, thereby making the problem *non-deterministic*. This means that there may be more than one acceptable lighting condition for a given context. Therefore, it is more appropriate to measure the degree of relevance of a predicted lighting condition than measuring its accuracy. Moreover, each observed context may have different distribution of output class labels. For example, a context x_1 may have the Warm-Bright-Static (WBS) lighting condition selected 7 times and the Cool-Bright-Static (CBS) lighting condition selected 3 times in the past, whereas the context x_2 may have WBS selected 1 time and CBS selected 9 times. This means that to evaluate a predicted output lighting condition, it is necessary to know about the contexts and also the frequency of outputs that are selected for those contexts.

In this direction, we have developed a performance metric dubbed *RS*. Unlike CA, *RS measures the degree of relevance of a predicted output for a given context with respect to the actual output* [55]. When there is a mismatch between predicted and actual outputs, RS will account for the level of inconsistency of non-deterministic multiple-output cases. The RS metric will determine the relevance of a predicted output based on the number of times these outputs have occurred before in a similar context. Therefore, the RS metric does not deal with the learning aspects of existing prediction algorithms. Instead, RS allows evaluating predictions from a different perspective where the CA metric fails. For example, if there is a mismatch between the predicted and actual outputs for a given context, the CA scores 0, whereas RS may score 70% indicating that the predicted output is still very relevant for that context. Formally, $RS = g(y_P, y_A, x, D)$ where D is the dataset under investigation. This means

that RS is a subjective metric that depends on the context and the information gathered from the dataset to evaluate a predicted output.

We have further developed RS into two variants, Relevance Score - Case-by-Case (RS_{CC}) and Relevance Score - General (RS_{Gen}). RS_{Gen} is computed by discarding *abstract* features from a dataset. We define *abstract* feature as the one that is either difficult or impossible to model the complete behavior. For example, the user-identity (UID) is an abstract feature in the Breakout dataset. This is because UID only provides information about who the user is, and not about some of the important qualities of that user such as user mood or behavior. By ignoring the abstract feature the dataset becomes as if it is collected from a single user. The details of computing RS and the significance of the RS metric is described in Chapter 5.

4.4 Analysis of the Breakout Dataset

The experiments are performed on the Breakout dataset using *WEKA* (Waikato Environment for Knowledge Analysis) [60]. WEKA is a collection of state-of-the-art machine learning algorithms and data preprocessing tools for data mining tasks. WEKA version 3.6.6 implements nine rule-based prediction algorithms. We do not select ZeroR and OneR algorithms as their prediction performance measured using Classification Accuracy (CA) was lower compared to other algorithms. Also, the CA performance of the DecisionTable Naïve-Bayes algorithm is similar to that of DT and hence it is not considered. We perform the experiments to investigate; 1) the prediction performance and 2) the significance of the input features considered. The performance of the prediction algorithms on the Breakout dataset is computed using 10-fold cross-validation.

4.4.1 Analysis of the Performance of Prediction Algorithms

With 8-Outputs

Figure 4.1a presents the prediction performance of the six rule-based prediction algorithms, considering CA and RS as metrics. It can be seen that CA values are very low for all the considered algorithms, compared to RS values. This is because the CA metric measures how accurate the prediction is for a given context, i.e. the predicted output is compared to the actual output. If the predicted and actual outputs do not match, then the CA metric scores a zero. Since users are not consistent in selecting a particular lighting condition for a given context,

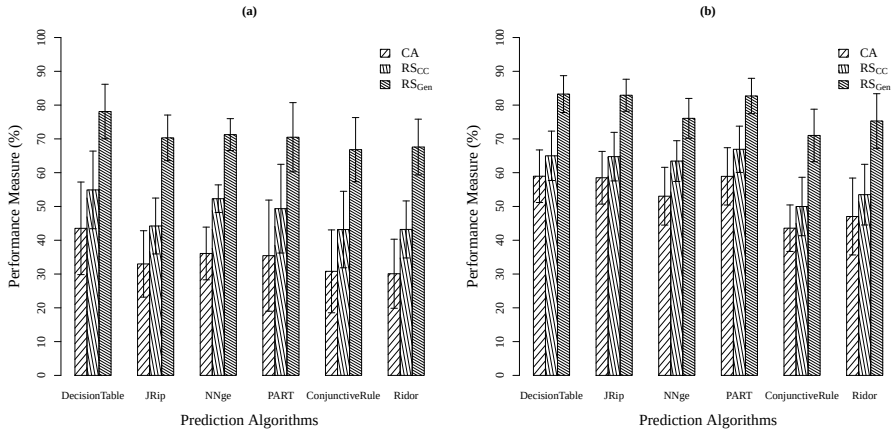


Figure 4.1: Performance of the six rule-based prediction algorithms
a. With 8-output lighting conditions,
b. With 4-output lighting conditions.

the average CA for a lighting application is typically low. The inconsistency comes from the fact that it is very difficult (indeed impossible) to consider the full set of input features (context) that determine a user's preference for a lighting condition. Furthermore, some contextual information, such as user mood, cannot be monitored easily. Instead, a learning algorithm takes only a part of all relevant input features into account. Since, multiple lighting conditions can satisfy a user in a given context, the nature of the Breakout Dataset, i.e. the input-output relationship is *one-to-many*. The RS metric measures how relevant the predicted output is, for a given context based on the information computed from the dataset [55]. The RS metric does not often score a zero when there is a mismatch between the predicted and actual outputs and thus shows higher performance.

With 4-Outputs

The output lighting conditions are designed in terms of intensity (Bright and Dim), warmth (Warm and Cool) and dynamics (Static and Dynamic). The study conducted by Offermans et al. [104] showed that the output lighting conditions were representative in terms of intensity and warmth, but not in terms of dynamics. This means that the users could not differentiate sufficiently between

Table 4.1: Standard deviation (SD) of the performance for the six rule-based prediction algorithms for 8-output lighting conditions. The best values are highlighted in **bold** and the worst values are highlighted in *italics*

Prediction Algorithm	SD (CA)	SD (RS_{CC})	SD (RS_{Gen})
DecisionTable	13.7	11.49	8.07
JRip	9.83	8.27	6.75
NNge	**7.8**	**4.09**	**4.72**
PART	*16.45*	*13.13*	*10.24*
ConjunctiveRule	12.26	11.32	9.49
Ridor	10.22	8.44	8.25

static and dynamic settings. Therefore, in this experiment, static and dynamic lighting conditions are combined into 4-output classes. Figure 4.1b presents the performance values of the six rule-based prediction algorithms considering CA and RS as metrics. It can be seen that since the output space is reduced, the CA values improve as compared to the results of the 8-output dataset. However, the RS values do not improve much. This means that when the outputs are merged i.e. when the output space is reduced the prediction algorithms can predict more accurately. However, the predicted lighting conditions are slightly more relevant than in the case of 8-output lighting conditions.

Dataset Dependency of Prediction Algorithms

Table 4.1 shows the standard deviation of the performance for the six prediction algorithms computed using 10-fold cross-validation. The values show that there is a high degree of inconsistency in the prediction performance for both the metrics considered. This means that the performance of the prediction algorithms that use supervised learning approach varies significantly with different training and test sets.

From this study, we find that intelligent lighting is an application where more than one output may be acceptable for a given context. The experimental results showed that evaluating supervised learning algorithms using CA as the performance metric is inappropriate considering the nature of the intelligent lighting. Also, experiments with 4-output conditions showed that reducing one-to-many input-output relationships enable prediction algorithms to predict

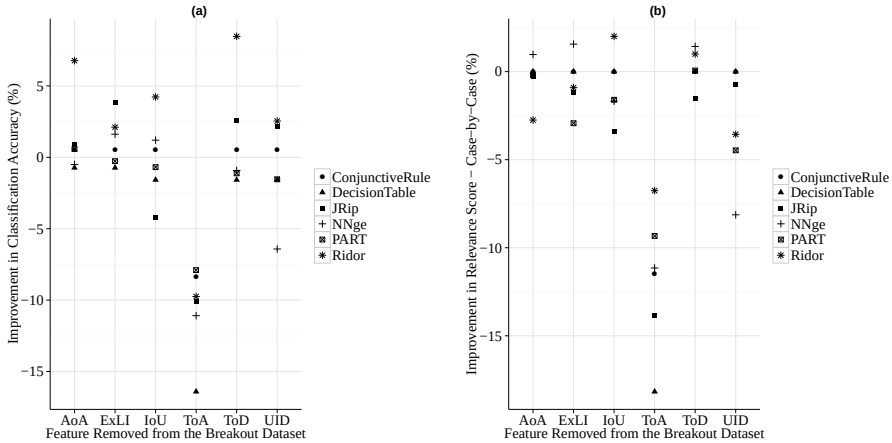


Figure 4.2: Improvement in the prediction performance for six rule-based prediction algorithms for the 5-feature dataset with 8-output lighting conditions
a. With *Classification Accuracy* as metric,
b. With *Relevance Score - Case-by-Case* as metric.

more accurately.

4.4.2 Significance of Input Features

In this experiment, we seek to study the influence of the considered input features on the performance of the prediction algorithms. We discard the values of each feature in every trial and then perform the experiment on the dataset with 5-features. Subsequently, we determine the *improvement* in prediction performance measured using CA and RS_{CC} as metrics. The *improvement* is the difference measured between the performances of a prediction algorithm using the 6-feature and 5-feature dataset. A decrease in prediction performance means a negative improvement.

Figure 4.2a shows the influence of the selected input features on the performance of prediction algorithms, using CA as the metric. It is evident that the level of influence depends on the selected prediction algorithm. If the improvement is negative, then we say that the removed feature has an impact on the algorithm's performance. For the DT algorithm, the improvement is negative for every input feature removed. This means that the DT model makes use of

every feature to predict a lighting condition whereas for the ConjunctiveRule and Ridor algorithms, the *Type of Activity* (ToA) is the only feature needed. The NNge model depends on both *ToA* and *User identity* (UID) to predict a lighting condition. Also, the Ridor model shows better CA performance when features other than ToA were discarded. From the obtained results, we can say that it is difficult to ignore any of the features as their influence is not the same across the prediction algorithms. However, the CA performance reduces drastically when the *ToA* feature is removed for all the prediction algorithms. This shows that the feature *ToA* is very significant in selecting a lighting condition.

With the RS_{CC} metric, the procedure to study the influence of input features remains the same as in case of the CA. However, the RS_{CC} values are computed using the 6-feature dataset. The training and prediction phase is done using the 5-feature dataset whereas evaluating the predicted output makes use of the 6-feature dataset. This is because since the RS_{CC} is a function of the input, evaluating with five features will produce result that is relevant only for the 5-feature input. The impact of a feature can only be noticed if all the six features are considered as input while evaluating a predicted output. Figure 4.2b shows the influence of input features on the performance of prediction algorithms. The DT and ConjunctiveRule algorithms depend only on the *ToA* feature to predict a lighting condition. This means that for the considered dataset, other input features can be ignored to give relevant predictions in case of the DT and the ConjunctiveRule prediction models. For the JRip model, all the input features are necessary to provide relevant predictions and the ToA feature has maximum influence. The NNge algorithm depends mainly on both *ToA* and *User identity* (UID) to provide a relevant lighting condition. Here also, as in the case of CA, we conclude that the ToA is the most influential feature that helps in predicting the most relevant and accurate lighting conditions.

4.5 Conclusion

Data analysis of an intelligent lighting application leads to insights into the input-output relationship as well as suitability of different performance metrics and performance limitations. In designing intelligent applications, such insights help in deciding upon the learning models and how to improve feature space. By means of statistical analysis of the Breakout dataset collected from a pilot implementation, we were able to infer that the Breakout dataset has a *one-to-many* input-output relationship, unlike many available real-world datasets. This means that more than one output may be suitable for a given context. The ex-

periments were performed using six rule-based prediction algorithms and two performance evaluation metrics: CA and RS. We find that the CA is not an appropriate metric for applications such as intelligent lighting having *one-to-many* input-output relationships. In the next chapter, we explain the performance metric, *RS*, which we developed for evaluating machine learning algorithms in the context of intelligent lighting that has one-to-many input-output relationship.

CHAPTER 5

Relevance as a Metric for Machine Learning Algorithms

5.1 Introduction

In Chapter 4, we have seen that the input-output relationship in intelligent lighting is non-deterministic and one-to-many. This kind of variable or inconsistent behavior is rather common in intelligent lighting, whereby human perception plays an important role. Making predictions in a perceptual context involves a range of non-linear, non-deterministic factors, including psychophysics (e.g. the human audio-visual system has limited accuracy) [129]; psychology (e.g. biases and mood affect perception of reality) [122, 90]; cost and lifestyle (e.g. expectations grow substantially with economic implications) [52]; and all sorts of other cognitive elements [8]. The key problem here is to evaluate the performance of such a system in an appropriate way, when there may be more than one output that is satisfactory for a given context. Also, the most satisfactory output may change non-linearly, unpredictably and over the time [91]. The commonly used metrics such as classification accuracy (CA) can only measure whether the predicted light condition is accurate or not, but does not help appreciating whether a light condition is relevant to the overall context. Hence, a new performance metric that suits perceptual machine learning is required, which motivates our work.

Conventionally, classification problems are broadly categorized into *multi-class classification* and *multi-label classification* problems. Multi-class classification algorithms [6] are for those categories of problems where, for a given input

instance, there is exactly one correct output class. Examples of multi-class classification include image recognition and diagnosis [131]. The commonly used performance evaluation metrics for multi-class classification problems are *CA*, *precision* and *recall*. Precision and recall are generally used in binary classification problems where there are only two output classes (positive and negative). For a set of samples, CA is the proportion of predicted cases that are accurately predicted. Precision is the proportion of predicted positive cases that are true positives. Conversely, recall is the proportion of true positive cases that are correctly predicted as positive [109]. Multi-label classification algorithms [125] address the category of problems where more that one output class must be selected for each input instance. Examples of multi-label classification include text [88] and music categorization [124]. The commonly used performance evaluation metrics for multi-label classification problems are *hamming-loss*, *precision* and *recall* [53]. Hamming-loss is the fraction of the wrong labels and the total number of labels.

There exists another class of applications such as the intelligent lighting where the aim is to provide a suitable output based on an observed context. We call these applications, *non-deterministic multiple output* problems. In these applications, the ML algorithm has to select a unique output class for a given input instance (as in multi-class classification), but the output for a given input instance may fall into multiple acceptable output classes (as in multi-label classification). Thus, although multiple output options may be acceptable, we are forced to select only one, and we want this to be the most consistent with and relevant to the context. An interesting property of such an application is that the relationship between input and output is not deterministic and also changeable over time (due to human perception inconsistency), i.e. there is no single acceptable output for a given context. These types of problems arise when, 1) it is not possible to identify all the features that are relevant to determine the output or 2) it is very challenging to model the complete behavior of some features such as user mood or variability in user perception. In such cases, even though the ML algorithms that are used to solve multi-class classification problems suit the need, the evaluation metrics such as accuracy, precision and recall are not suitable. The use of CA as a metric leads to a score of zero for an inaccurate prediction. This metric does not consider the non-deterministic nature of the problem or the changes due to human-perception variability.

It is a challenge to evaluate and compare the performance of different ML algorithms when the observed context is not entirely representative of the actual context of an application. In the literature, different evaluation metrics assess different characteristics of ML algorithms [121]. A bad choice of performance

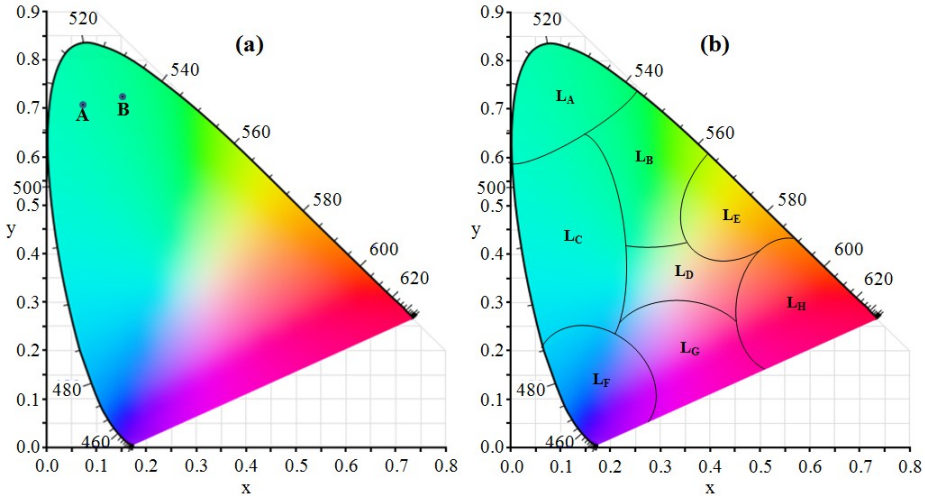


Figure 5.1: CIE (International Commission on Illumination) 1931 color space
a. Spatially distinct points A and B on the color space that is perceived to be same,
b. An example of categories of light conditions where two points inside a category is assumed to be same

metrics would produce results that are meaningless in evaluating performance against the design goal of an application [69] [68]. This has motivated the need for a new evaluation metric that penalizes the prediction algorithms only to the extent that predicted outputs are inconsistent.

To this end, we detail the performance metric dubbed *Relevance Score (RS)* introduced in Chapter 4 to evaluate the performance of ML algorithms for the class of applications tackling non-deterministic multiple output problems. We first discuss the drawbacks of the CA metric for multiple output problems using a practical intelligent lighting example. We then explain our RS performance metric. Finally, we illustrate the RS metric by means of examples and provide an experimental evaluation to demonstrate the significance of RS.

Table 5.1: Representational comparison of two ML performance metrics for an intelligent lighting application considering a series of five predictions.

Actual Output	Predicted Output	CA Metric	Alternative Metric
L_A	L_A	100	Absolutely relevant (100)
L_B	L_C	0	Relevant (80)
L_C	L_A	0	Absolutely irrelevant (0)
L_A	L_A	100	Absolutely relevant (100)
L_B	L_A	0	Moderately relevant (60)
	Average	200/5 = 40%	340/5 = 68%

5.2 Classification Accuracy as a performance metric

Here, we discuss the drawbacks of CA as a metric for evaluating machine learning algorithms, using the intelligent lighting scenario as an example. Consider an intelligent lighting application, where the goal is to predict and provide a desired light condition to users for every observed context. Fig. 5.1a shows the CIE 1931 color space from which the intelligent lighting system can create an ambient lighting condition. Current generation of lighting systems can provide millions of saturated colors [83]. This means that if a machine learning model is used to predict a suitable light condition, there will be a choice of millions output class labels. However, this is not practical because it is very unlikely that a human eye can differentiate between similar light conditions. For example, Fig. 5.1a shows two spatially separated points, A and B, on the color space that appear to be same. A solution can be to group similar colors into separate categories as shown in Fig. 5.1b, where a million class labels are reduced to eight categories.

Let us assume that a supervised prediction algorithm, h is trained using ten data samples collected from a pilot implementation. Let the output light condition selected for a given fixed context, x be light condition L_A five times, light condition L_B three times and light condition L_C two times. This means that for x the user has selected L_A, L_B, L_C and L_D with probabilities 0.5, 0.3,

0.2 and 0.0, respectively, i.e. a sample *x* can be categorized into multiple output classes. Assume that, at run-time, *h* selects L_B, whereas the user actually desires L_C. Here, *h* is not right in selecting the correct output, but it is not entirely wrong either, as the user has not been consistent in selecting the desired lighting condition for the observed context. If CA is used as a metric to evaluate the performance of *h*, the selection made would be assessed as completely wrong, which is not desirable. CA would not make a separation between selecting L_A or L_D. From the application point-of-view, the CA metric is not very representative as it is desirable to measure how relevant the lighting condition is for a given observed context, rather than assessing how accurate it is.

From the example above it is evident that CA is not an appropriate metric for similar applications as it fails to capture the relevance of predicted output when there is a mismatch between the predicted and actual outputs. An example of an *Alternative* metric that computes the relevance of the predicted output for an intelligent lighting application is shown in Table 5.1. It can be seen that the Alternative Metric seems to be more appropriate than CA.

In this direction, we propose a performance metric named *Relevance Score*, which is more suitable than the commonly used CA metric for the third class of applications having non-deterministic multiple outputs as mentioned in Section 5.1. In particular, we seek to find a performance function that maps an input and output pair (x, y) to a real number $RS : (x, y) \in (X, Y) \rightarrow [0, 100]$ rather than $\{0, 100\}$ as in CA. The RS metric provides a score between the range 0 and 100, whereas for CA, it is either 0 or 100.

5.3 The Relevance Score Metric

In this section, we present the RS metric to evaluate prediction algorithms used for non-deterministic multiple output problems. Furthermore, we describe a mechanism to compute two variants of RS, dubbed *Relevance Score - Case-by-Case* (RS_{CC}) and *Relevance Score - General* (RS_{Gen}). When we use the term RS, it refers to both RS_{CC} and RS_{Gen}. RS is computed using the error score *(ErrScore)* for every instance. ErrScore is a real number indicating the degree of mismatch between actual and predicted outputs.

For l test set samples, the CA metric is computed as an average of individual accuracies as in equation 5.1. Similarly, the RS on a test set with l samples is an average of individual relevance scores as in equation 5.2.

$$CA = \frac{1}{l}\sum_{i=1}^{l} CA_i \tag{5.1}$$

$$RS = \frac{1}{l}\sum_{i=1}^{l} RS_i \tag{5.2}$$

5.3.1 Relevance Score - Case-by-Case

We define RS_{CC} for a given predicted output, y_P by a generic prediction algorithm. We do this in a number of steps, by first defining concepts of distance between outputs and $ErrScore$.

The RS_{CC} is computed in two phases; 1. computing posterior probabilities, and 2. evaluating the prediction. The process of computing RS is shown in Fig. 5.2.

Computing Posterior Probabilities

Let $X = X_1 \times X_2 \times X_3 \times \ldots \times X_n$ denote an n-dimensional input feature space where the i^{th} sample $x_i \in X$ is given by $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Let y denote the output class label, where $y \in Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(K)}\}$ and K is the number of possible outcomes. For a given dataset with m samples, the posterior probabilities of the output class labels $y \in Y$ are computed over all instances x in the dataset. Formally, conditional probabilities $P(y^{(k)}|x_i)$ for $i = 1, 2, 3, \ldots, m$ and $k = 1, 2, \ldots, K$ are computed. These probabilities are needed to calculate *ErrScore*. For a context x, we denote the predicted output by y_P, the actual output by y_A, the probability of occurrence of the predicted output by $P(y_P)$, the probability of occurrence of the actual output by $P(y_A)$ and the probability of the most frequently selected output by $P(y_H)$. *ErrScore* is a function of these variables. Formally, $ErrScore = f(y_A, y_P, P(y_A), P(y_P), P(y_H))$. For the given example in Section 5.2, $y_P = L_B$, $y_A = L_C$, $y_H = L_A$. Hence, $P(y_A) = 0.2$, $P(y_P) = 0.3$ and $P(y_H) = 0.5$.

Inorder to define RS, the feature values in the dataset need to be categorical. This is because, features that are continuous can take infinitely many different values. In an extreme case, every input in the dataset may be different and hence has a unique output, resulting in deterministic one-to-one input-output relationship. This is not entirely true as the input points on n-dimensional space

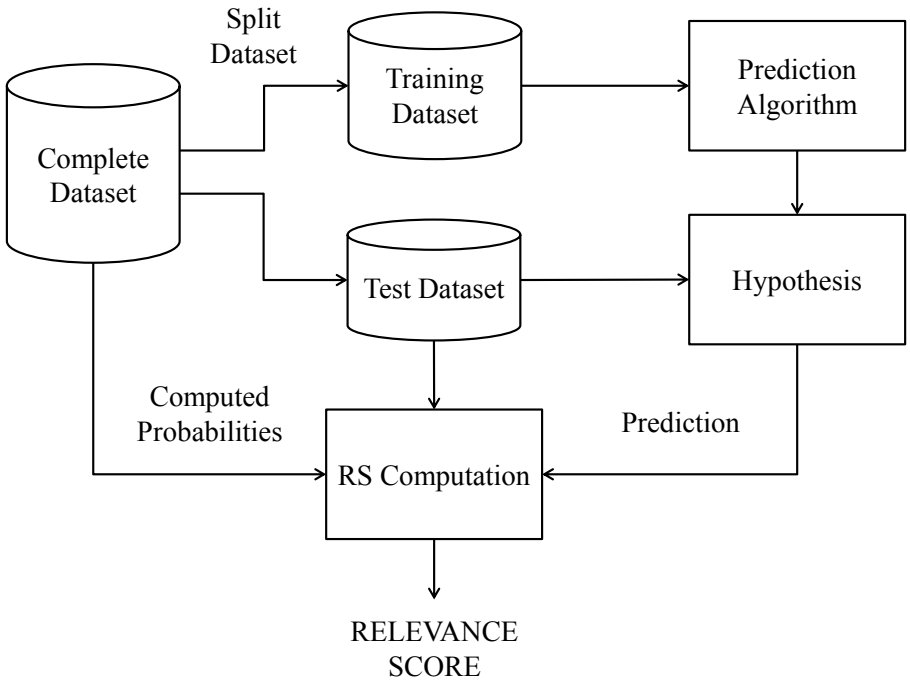


Figure 5.2: Experimental procedure to compute *Relevance Score*

may be close to each other, denoting that the inputs are similar but with different outputs as shown in Fig 5.1a. This problem can be avoided, if continuous features are categorized. For the CA metric, the predicted and actual outputs are compared and hence there is no dependence on input variables.

Evaluating the Prediction

In this phase, the predicted outcomes on the test data are evaluated. For a given context x, let d denote probabilistic distance between two outputs in general. We define d as the difference between the probabilities of two different outputs. The d value for different outputs are given by,

$$d_{HP} = |P(y_H) - P(y_P)| \tag{5.3}$$

$$d_{PA} = |P(y_P) - P(y_A)| \tag{5.4}$$

Table 5.2: Possibilities based on the predicted output, actual output, relation between their probabilities and the corresponding decreasing order of relevance of the predicted output in qualitative terms

Case	Outcome	Probability Condition	Qualitative Relevance
1	$y_P = y_A$	-	Absolutely Relevant
2	$y_P \neq y_A$	$P(y_H) = P(y_P) = P(y_A)$	Very Relevant
3	$y_P \neq y_A$	$P(y_H) = P(y_P); P(y_P) > P(y_A)$	Relevant
4	$y_P \neq y_A$	$P(y_H) > P(y_P) > P(y_A)$	Moderately Relevant
5	$y_P \neq y_A$	$P(y_H) > P(y_A) > P(y_P)$	Slightly Irrelevant
6	$y_P \neq y_A$	$P(y_H) = P(y_A); P(y_A) > P(y_P)$	Absolutely Irrelevant

$$d_{HA} = |P(y_H) - P(y_A)| \tag{5.5}$$

For a fixed context x, there are several possibilities based on the predicted and the actual outcomes and their computed probabilities. The cases may be evaluated qualitatively in terms of relevance of the predicted outputs. The decreasing order of relevance we consider for different cases is summarized in Table 5.2. The ordering is based on the consequence of individual cases as explained next. *ErrScore* is thus a score obtained by quantifying these cases.

Case 1: $y_P = y_A$

In this case, the predicted output and the actual output are equal. Thus, there is no error in the predicted output i.e., $ErrScore = 0$.

Case 2: $y_P \neq y_A; P(y_H) = P(y_P) = P(y_A)$

In this case, the predicted output and the actual output are not equal, whereas the probability parameters are equal. This means that the y_H, y_P and y_A have occurred equally frequently, i.e., any of these three outcomes is equally good (statistically) for that context. Therefore, the $ErrScore$ in this case is also zero.

Case 3: $y_P \neq y_A; P(y_H) = P(y_P) > P(y_A)$

In this case, as shown in Fig. 5.3a, the probabilities of the predicted output and the most frequently selected output are equal, i.e., we have $d_{PH} = 0$. This means that the prediction algorithm has selected the most frequently selected output but it has not been able to capture the change in output. The error is small and is equal to $\beta \cdot d_{PA}$ where β is a positive real constant, whose value depends on the application.

Case 4: $y_P \neq y_A; P(y_H) > P(y_P) > P(y_A)$

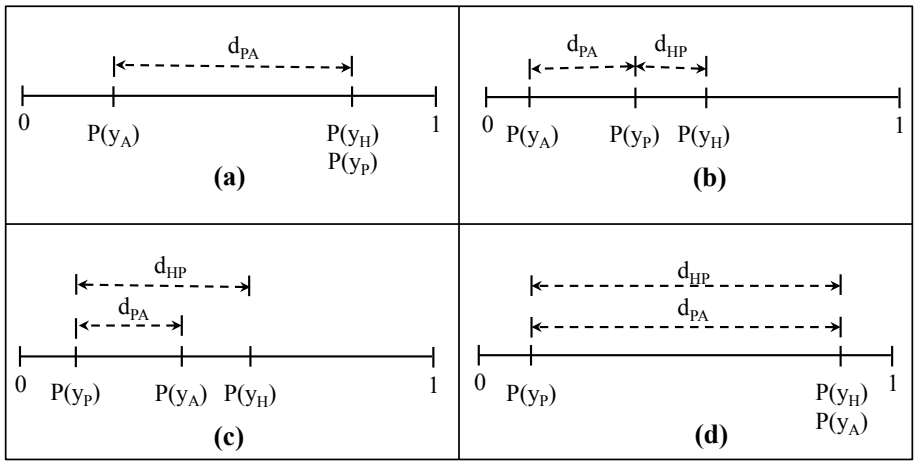


Figure 5.3: Possibilities that arise when predicted and actual outcomes are not same;
a. $y_P \neq y_A$ and $P(y_H) = P(y_P); P(y_P) > P(y_A)$
b. $y_P \neq y_A$ and $P(y_H) > P(y_P) > P(y_A)$
c. $y_P \neq y_A$ and $P(y_H) > P(y_A) > P(y_P)$
d. $y_P \neq y_A$ and $P(y_H) = P(y_A); P(y_A) > P(y_P)$

In this case, as shown in Fig. 5.3b, the probabilities of predicted, actual and most frequently selected outputs are not equal. The probability of the predicted output lies in between that of the most frequently selected output and the actual output. This means that the prediction algorithm has predicted an output that is not most frequently selected, but it is more likely than the actual output. Hence, the error is equal to $(\alpha \cdot d_{HP} + \beta \cdot d_{PA})$ where α denotes the positive real constant.

Case 5: $y_P \neq y_A; P(y_H) > P(y_A) > P(y_P)$

In this case, as shown in Fig. 5.3c, the probability of the predicted output is less than that of the most frequently selected output and of the actual output. This means that the prediction algorithm has predicted an output that is not most frequently selected, and is less probable than the actual output. Hence, d_{HP} is higher than that in Case 4 and the error is equal to $(\alpha \cdot d_{HP} + \beta \cdot d_{PA})$ as in Case 4.

Case 6: $y_P \neq y_A; P(y_H) = P(y_A) > P(y_P)$

In this case, as shown in Fig. 5.3d, the probability of the actual output and that of the most frequently selected output are equal, but the prediction algo-

rithm predicts a different output. The prediction algorithm was not sufficiently good to select the same output, the error is much higher than that of the previous cases. Since $d_{HP} = d_{PA}$, the error is equal to $(\alpha + \beta) \cdot d_{HP}$.

Combining above equations and normalizing over $(\alpha + \beta)$, we have the following $ErrScore$ computation,

$$ErrScore = \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta}. \tag{5.6}$$

The *RS* value for a context x_i is thus computed as a function of $ErrScore$ as

$$RS_i = (1 - ErrScore_i) \times 100, \tag{5.7}$$

$$RS_i = (1 - \frac{\alpha(d_{HP(i)}) + \beta(d_{PA(i)})}{\alpha + \beta}) \times 100. \tag{5.8}$$

5.3.2 Relevance Score - General

RS_{Gen} is a variant of RS, computed by discarding the values of features that are identified as *abstract*. An abstract feature is a feature that is too difficult to model such as user mood. The presence of an abstract feature is one of the reasons for having one-to-many relationships between input and output. By removing the values of the abstract feature (c), the input dimension is reduced by one (X_{-c}). The posterior probabilities $P(y^{(k)}|x_i)$ for $i = 1, 2, 3, \ldots, m$ and $k = 1, 2, \ldots, K$ are then computed. The remaining process to compute RS_{Gen} is same as that of the RS_{CC}. The motivation to compute RS_{Gen} is to study the performance of prediction algorithms for a given context without abstract features.

5.4 The Relevance Score by Example

In this section, we illustrate the benefits of the RS metric through two separate cases, i.e. the *Wine* dataset from the UCI ML repository [11] and an intelligent lighting example [56].

5.4.1 Experiments with Wine (White) Dataset

The *Wine* dataset [27] is used to determine the quality of a wine sample based on the measurements from objective tests such as pH values, density and sulphates. The Wine dataset is of two types; red with 1599 samples and white

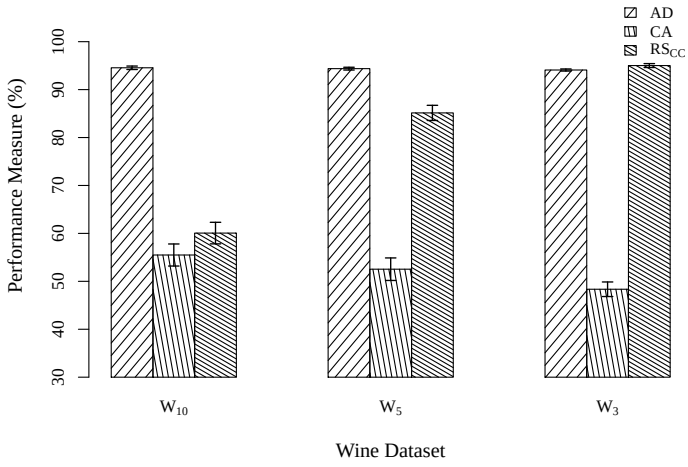


Figure 5.4: Performance of the PART prediction algorithm on the *Wine* dataset with categorized input space. The data is characterized by W_{10}: determinism, W_5: medium uncertainty and W_3: high uncertainty

with 4898 samples. The inputs include measurements from objective tests that are of numerical data type and the output is based on sensory data (median of at least 3 evaluations made by wine experts). The output class labels are numeric and ordered with the wine quality ranging between 0 (very bad) and 10 (excellent). Therefore, the problem addressed using the Wine dataset can be viewed as classification or regression. For our experiments, we consider the Wine (white) dataset as a classification task.

We consider three metrics; Absolute Difference (AD), CA and RS$_{CC}$ to evaluate the performance of prediction algorithms. AD is a metric that computes the difference between the predicted and actual output. The key difference between CA and AD is that CA measures whether the predicted output is right or wrong, whereas AD depends on the actual error. Therefore, for the Wine dataset, the performance measured using AD represents the ground truth against which the performance using CA and RS$_{CC}$ can be compared. The performance values obtained using the AD metric have been normalized to 100 i.e., [0, 100] using the expression $(10 - Error) \times 100$ whereby $Error = |y_A - y_P|$, so that the results

Table 5.3: Standard deviation of the prediction performance measured using the *Wine* dataset for the PART algorithm

Dataset	AD	CA	RS_{CC}
W_{10}	0.36	2.30	2.25
W_5	0.29	2.35	1.58
W_3	0.22	1.52	0.41

can be directly compared with those of the CA and RS_{CC} metrics, respectively.

In order to apply RS_{CC}, we need the input features to be of categorical data type whereas the input features in the Wine dataset is of numerical data type. Thus, for our experiments, we consider three cases. In each case we divide each input feature into 10, 5 and 3 equal categories. We denote the corresponding datasets by W_{10}, W_5 and W_3 respectively. At this point, the output class labels are unaltered. We perform the experiments with the PART prediction algorithm [48] as it provided the maximum CA performance compared to other rule-based supervised learning algorithms. The results are obtained using 10-fold cross-validation. Fig. 5.4 shows the performance of the PART algorithm on the Wine dataset and three performance metrics.

The CA performance of W_{10} is low (around 55%) as the considered input features are not sufficient to determine the output more accurately. Some of the features such as the grape types and wine selling prices are not considered due to privacy and logistical issues [27]. However, the performance measured using the AD metric is around 95%, which means that (when $y_P \neq y_A$) for the given contexts, the prediction algorithm is predicting outputs that are not totally wrong. For example, if $y_P = 3$ and $y_A = 4$ the CA metric scores a zero whereas the AD metric scores 90%. The performance measured using the RS_{CC} metric is approximately 60% i.e. similar to CA. This is because not many samples in W_{10} have one-to-many input-output relationship and the input space is least overlapped i.e. it is difficult to find multiple instances of an observed input. This case shows that when a dataset has deterministic input-output relationships (no or very little uncertainty), the RS_{CC} metric measures similar to the CA metric.

In case of W_5, we increase the overlapping of the input space, i.e. we introduce more uncertainty between the input and output by reducing the number of categories for each input feature to 5. This means that there is more possibility of observing the same input being mapped to different output class labels (more

uncertainty). It is more difficult for a prediction algorithm to predict the exact output for a given input, when there is more than one acceptable output. It can be seen that for W_5 the performance reduces to around 52% with the CA metric as the algorithm has more difficulty in selecting the accurate output when there is uncertainty. However, the performance with the AD metric is almost the same as in case of W_{10}. This reflects that introducing little uncertainty does not really affect the performance of the PART algorithm as seen through the AD metric. When $O_P \neq O_A$, the PART algorithm predicts a relevant output despite the increased randomness. The performance measured using the RS_{CC} metric increases significantly to 85%, closer to the performance measured using the AD metric. This shows that when there are one-to-many input-output relationships in a dataset, RS_{CC} gets closer to the ground truth. In order to confirm, we further increase the uncertainty between the input and output by reducing the number of categories for each input feature to 3. In this case, the performance measured by the RS_{CC} metric is 95%, i.e. almost the same as measured using the AD metric, reflecting the ground truth. However, the performance measured using the CA metric reduces further to 48%.

Table 5.3 shows the standard deviation (SD) of the prediction performance measured using the three metrics for the PART algorithm. For the AD metric, the SD is high for W_{10}, which reduces as the uncertainty between the input and output increases. The SD measurement confirms that when there is less uncertainty the RS_{CC} metric is similar to CA. The SDs of RS_{CC} and CA metrics are almost the same. However, as the uncertainty increases, the SD of RS_{CC} becomes similar to that of AD, while the SD of CA is still high. This shows that the RS_{CC} metric is an appropriate metric for the non-deterministic multiple output problem and represents the ground truth better than the CA metric.

5.4.2 Experiments with Intelligent Lighting Data

Consider the example of intelligent lighting in Section 5.2. For simplicity, assume that a data sample has only one feature, the user-identity UID and a single user. The dataset contains 10 samples with output light conditions L_A, L_B and L_C selected 5, 3 and 2 times respectively. With this setting, we have L_A as the most frequently selected outcome. Let us examine several cases, where a prediction algorithm h selects an output lighting condition, y_P for a new input and the user selects a different lighting condition i.e. the actual output y_A.

Table 5.4 shows several cases for different predicted and actual outputs. In most cases, the CA performance is zero, whereas the RS_{CC} metric provides a different score based on y_P, y_A and the context x. In case 1, $y_H = y_P$ and y_A

Table 5.4: Possibilities that arise (for a context) when predicted and actual outcomes are not same and their corresponding relevance scores

Case	y_H	y_P	y_A	$P(y_H)$	$P(y_P)$	$P(y_A)$	$ErrScore$	RS_{CC}
1	L_A	L_A	L_B	0.5	0.5	0.3	0.066	93.33
2	L_A	L_A	L_C	0.5	0.5	0.2	0.100	90.00
3	L_A	L_B	L_C	0.5	0.3	0.2	0.433	56.67
4	L_A	L_B	L_A	0.5	0.3	0.5	0.466	53.33
5	L_A	L_C	L_B	0.5	0.2	0.3	0.633	36.67
6	L_A	L_C	L_A	0.5	0.2	0.5	0.700	30.00
7	L_A	L_A	L_A	0.5	0.5	0.5	0.000	100.00
8	L_A	L_B	L_A	1.0	0.0	0.5	1.000	0.00

is different. Therefore, the prediction algorithm must not be penalized for predicting a lighting condition that is not accurate. Thus, the calculated $ErrScore$ using equation 5.6 is minimum. In case 2, even though $y_H = y_P$, y_A is the least frequently selected output L_C. Therefore, the *ErrScore* is a bit higher than in the previous case i.e. 10%. In case 3, $y_H \neq y_P \neq y_A$ and $P(y_P)$ is in between $P(y_H)$ and $P(y_A)$. This means that neither $P(y_P) = P(y_H)$ nor $y_P = y_A$. Thus, the *ErrScore* is much higher than in case 2 i.e. 43%. In case 4, $P(y_H) = P(y_A)$, but the prediction algorithm made a mistake by selecting another output L_B, hence the $ErrScore$ is 47%. In cases 5 and 6, the prediction algorithm selects the least frequently selected output L_C. In case 5, $P(y_H) \neq P(y_A)$ and in case 6, $P(y_H) = P(y_A)$. Therefore, the *ErrScore* is the maximum in case 6. In case 7, $P(y_P) = P(y_A)$, thereby making the $ErrScore = 0$. In case 8, L_A is the most suitable lighting condition and also $y_A = L_A$. However, the prediction algorithm predicts L_B leading to an $ErrScore = 1$. Hence the RS_{CC} value is zero. There is also a special case as case 2 in Table 5.2 when all the light conditions are selected equally frequently i.e. $P(L_A) = P(L_B) = P(L_C) = 0.33$. In this case, the $d_{HP} = d_{PA} = 0$ and hence the *ErrScore* is zero. This means that we should not penalize the prediction algorithm if there is any mismatch between y_P and y_A, because any predicted lighting condition is likely to satisfy the user.

To summarize, for the non-deterministic multiple output problems in machine learning, prediction algorithms should not always be penalized for selecting an output that is not accurate. The experiments using the Wine (white)

Table 5.5: Description of the Datasets

Dataset	Breakout	Abalone	Iris (Modified)
Number of samples	236	4177	150
Number of input features	6	8	4
Type of input features	5-Categorical 1-Numeric	1-Categorical 7-Numeric	0-Categorical 4-Numeric
Output Cardinality	8	29	3
Type of output	Categorical	Categorical	Categorical

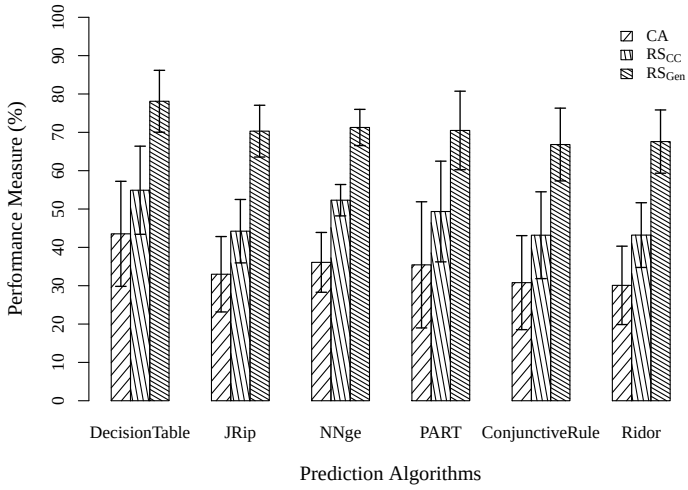


Figure 5.5: Performance of the prediction algorithms on the *Breakout* dataset

dataset and the intelligent lighting example have shown that the CA metric fails to capture the uncertainty between the input and output. The results also showed that the RS_{CC} metric is a more appropriate performance metric than CA.

5.5 Experiments and Results

The significance of the RS metric is demonstrated through various experiments. The experiments are performed primarily on the *Breakout* dataset [56] used for an intelligent lighting application and two other datasets, *Abalone* [99] and *Iris* [46] from the UCI ML repository. All of these datasets have a one-to-many relationship between input and output except for the Iris dataset. The main features of these datasets are given in Table 5.5. Six rule-based prediction algorithms are used to assess the datasets; DecisionTable (DT), JRip, NNge, PART, ConjunctiveRule and Ridor. We use the implementations available in WEKA for determining the performance of prediction algorithms (using CA and RS), using 10-fold cross-validation.

5.5.1 Experiments with the Breakout Dataset

The Breakout dataset for intelligent lighting consists of 236 samples with six input features, which are User-identity (UID), Type of Activity (ToA), Area of Activity (AoA), Intensity of Activity in other subarea (IoA), Time of the Day (ToD) and External Light Influence (ExLI). Of these features, the feature *ToD* is of numerical data type and the rest are of categorical data type. Since the categorical data types are needed for computing RS values, the feature ToD is converted into categorical data type by assigning a category to each feature value in the dataset. In our experiments, we have assigned the following categories; Category 1 [8.00, 12.00], Category 2 (12.00, 16.00], Category 3 (16.00, 20.00] and Category 4 (20.00, 8.00). We also reduce the number of categories for the feature IoA to two; Category 1 [0-2] and Category 2 [3-10].

Classification Accuracy vs. Relevance Score - Case-by-Case vs. Relevance Score - General

Fig. 5.5 shows the performance in three different metrics; CA, RS_{CC} and RS_{Gen}. The RS values are computed with $\alpha = 2$ and $\beta = 1$. The CA performance is low (for the reasons discussed in Section 5.2). This means that the prediction algorithms fail to provide the accurate lighting condition in more than 50% of the test samples. However, with RS_{CC} and RS_{Gen} metrics, a maximum of 55% and 78% performance is achieved with the DT prediction algorithm, as the RS metric measures how relevant a predicted light condition is for a given input.

An interesting observation is that the prediction algorithm Ridor has lower CA than Conjunctive Rule, whereas Ridor gives a better RS performance than

Conjunctive Rule. This can be justified as follows. For a given input, ConjunctiveRule provides more accurate light conditions than Ridor. However, when it comes to the relevance of the predicted outcomes, Ridor is more successful than ConjunctiveRule, i.e. it predicts the most relevant outcomes and avoids irrelevant outcomes. Even though ConjunctiveRule finds the best match slightly more often than Ridor, it comes up with less relevant predictions when it cannot find the best match. Similarly, the NNge and the PART prediction algorithms provide similar accuracies, whereas NNge provides more relevant predictions than PART in case of mismatch between the predicted and actual outcomes.

The RS_{Gen} metric is computed discarding the $abstract$ feature which in the Breakout dataset is UID. This is because UID only denotes the identity of a user, whereas user features such as mood or behavior are difficult to model. Also, each user may have their preference of light condition for a context. By discarding the UID feature, the context space is reduced to five features. Hence, the data can be viewed as if it were collected from a single user. This means the RS_{Gen} metric measures the relevance of the prediction output for a user that has the combined behavior from all users. From Fig. 5.5, we can see that the DT algorithm provides the maximum RS_{Gen} performance of 78%, whereas the ConjunctiveRule and Ridor algorithms provide the least performance of 67%. This means, for example, that when a user A is using the intelligent lighting system, the DT algorithm predicts the lighting conditions that are relevant not only for user A but also to most other users.

Significance of α and β

In this experiment, we investigate the influence that α and β have on the performance of the various prediction algorithms, using the RS_{CC} and RS_{Gen} metrics. Fig. 5.6 and Fig. 5.7 gives the results for RS_{CC} and RS_{Gen}, respectively for different combinations of α and β. We recall that the parameters α and β weigh the probabilistic distances d_{HP} and d_{PA}, respectively. When α is higher the product $\alpha \cdot (d_{HP})$ is emphasized more, i.e. the probabilistic distance between the most frequently selected output and the predicted output is given more importance. When β is higher, the product $\beta \cdot (d_{PA})$ is emphasized more, i.e. the probabilistic distance between the actual output and the predicted output is given more importance.

Generally, the RS values for all the prediction algorithms improve when α increases and β decreases. In both metrics (RS_{CC} and RS_{Gen}) the ConjunctiveRule prediction algorithm provides most consistent performance with changing values of α and β. On the other hand, the performance of the PART and the DT

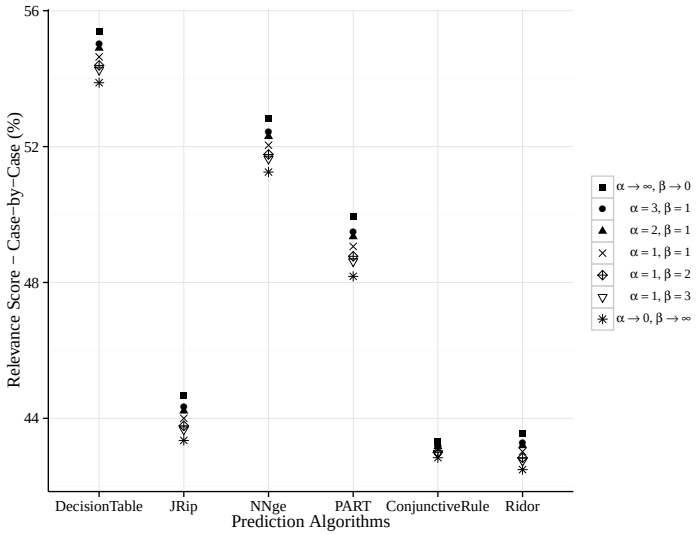


Figure 5.6: Relevance Score - Case-by-Case performance of the prediction algorithms for varying α and β on the *Breakout* dataset

prediction algorithms vary the most with RS_{CC} and RS_{Gen}, respectively. This means that when there is a mismatch between the predicted and the actual output, ConjunctiveRule selects the output such that its probability lies between that of the most frequently selected output and the actual output in most cases. This results in a performance that does not vary much with changing values of α and β. In contrast, in DT and PART the probability of the predicted output is close to either the most frequently selected output or the actual output. The low RS values in ConjunctiveRule are due to the reason that the prediction algorithm is not good at avoiding the less relevant outcomes.

It can be seen that with both RS_{CC} and RS_{Gen}, when α is very large compared to β, the Ridor prediction algorithm has better RS value than ConjunctiveRule, whereas when β is very large compared to α, ConjuctiveRule has better RS performance. As β goes to ∞, the probabilistic distance between the predicted and actual output is emphasized, meaning that ConjunctiveRule predicts closer to the actual output than Ridor. It can also be observed that in case of RS_{Gen}, as β grows larger than α, the difference in RS performance between NNge and JRip, and between NNge and PART increases. This shows that the predicted outcomes by NNge are closer to the actual outcomes than those of JRip and

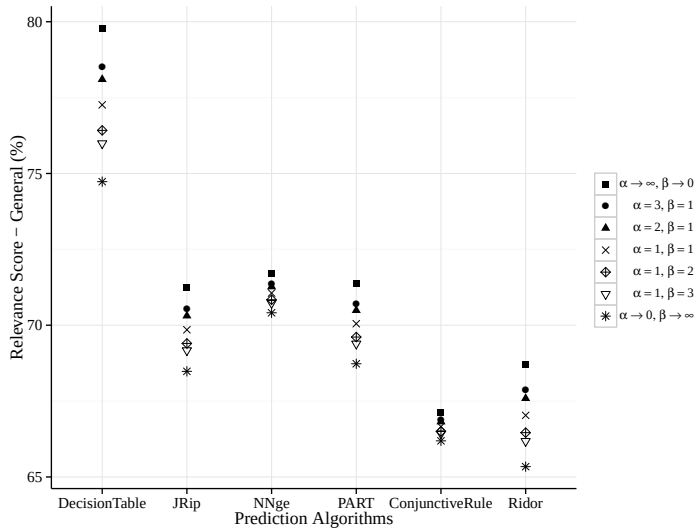


Figure 5.7: Relevance Score - General performance of the prediction algorithms for varying α and β on the *Breakout* dataset

PART.

For a given context x, the maximum value that the RS metric can achieve is 100%, which is possible in two cases. Case 1: When a prediction algorithm can accurately predict the output class for every context. Case 2: When output class labels appear equally frequently within a given observed context i.e. $P(y^{(1)}|x) = P(y^{(2)}|x) = \ldots = P(y^{(k)}|x)$ where $k \geq 2$. On the other hand, the minimum value that the RS metric can reach is zero. This occurs when an observed context x_i is completely distinct from the others in the dataset $(x_i \neq x_j | (i \neq j) \wedge i, j \in \{1, 2, \ldots, m\})$ and and the predicted output never match the actual output. Otherwise, the maximum or the minimum RS values that a prediction algorithm can achieve are found by taking either α or β very large. The lower and the upper bounds of RS for the considered prediction algorithms and the Breakout dataset are summarized in Table 5.5.

When α is very large compared to β, the error is computed as follows,

$$ErrScore = \lim_{\alpha \to \infty} \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta} = \lim_{\alpha \to \infty} \frac{\alpha(d_{HP})}{\alpha} = d_{HP}. \tag{5.9}$$

Table 5.6: Summary of the lower and upper *RS* bounds for various prediction algorithms on the *Breakout* dataset

	Relevance Score (Case-by-Case)		Relevance Score (General)	
	$\alpha \to \infty$	$\beta \to \infty$	$\alpha \to \infty$	$\beta \to \infty$
DecisionTable	55.40	53.88	79.79	74.73
JRip	44.66	43.35	71.23	68.48
NNge	52.83	51.25	71.71	70.41
PART	49.94	48.18	71.37	68.73
ConjunctiveRule	43.33	67.13	57.13	66.19
Ridor	43.56	42.49	68.71	65.34

When β is very large compared to α, the error is computed as follows,

$$ErrScore = \lim_{\beta \to \infty} \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta} = \lim_{\beta \to \infty} \frac{\beta(d_{PA})}{\beta} = d_{PA}. \tag{5.10}$$

Breakout Dataset with Random Output

In this experiment, we study the *RS* results, considering the same dataset but with randomly generated output. One of the main motivations for this case is to verify that the performance of the prediction algorithms is not penalized by inconsistencies in the data pattern, which are often caused by factors outside of the observed context. For example, a prediction algorithm whose goal is to determine the most desirable lighting conditions for a user can be expected to perform at best as good as the user himself. If a user generates inconsistent data, i.e. if an observed context maps to different outcomes at different times, then the student (the prediction algorithm) will be learning from an inconsistent teacher (the user), whose reasons for changing mind are completely hidden from the student, i.e. the reasons either don't exist or they are beyond the observed context. Consider the extreme case where there is no relation between the contexts and its corresponding output lighting conditions, i.e. the user throws an 8-sided dice to select the desired lighting condition and the dice is not part of the observed context. In this case, the CA metric would be equal

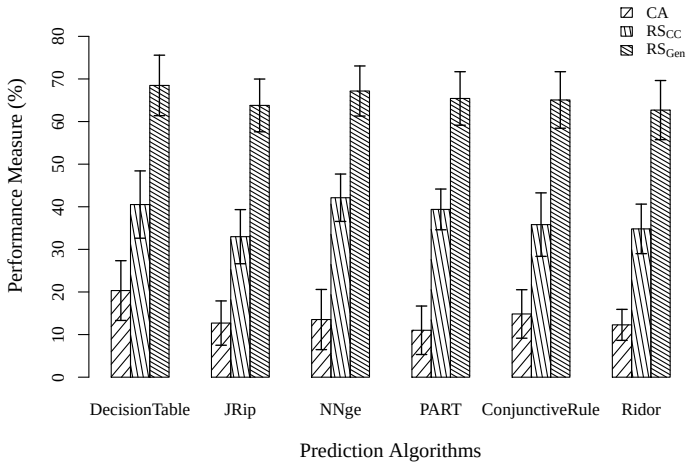


Figure 5.8: Performance of the prediction algorithms on the *Breakout* dataset with randomly generated outputs

to 1/8 as there are 8 possible lighting conditions. Note that the user makes no distinction between lighting conditions. For any given observed context, even though there could be exactly one desired lighting condition, all other possible lighting conditions are indeed somewhat relevant too. Therefore, we expect to achieve a higher RS in comparison to the scores on the dataset with real output. We also expect that RS will be always higher than the CA metric.

Fig. 5.8 shows the performance of the prediction algorithms for the Breakout dataset with randomly generated outputs. In the random output dataset, since one of the eight light conditions is chosen at random i.e. with probability 0.125, the output probability distributions are almost uniform for every context. This makes it difficult for prediction algorithms to select the right output. Therefore, the CA performance of all considered prediction algorithms lie in the range between 10 and 20%. The RS_{CC} and RS_{Gen} performance corresponding to the various prediction algorithms is lower in random output dataset compared to that of the real output dataset, which is unexpected. This is because of two reasons; 1) the number of samples are too small to capture the randomness and 2) only output class labels are selected randomly whereas the context (input)

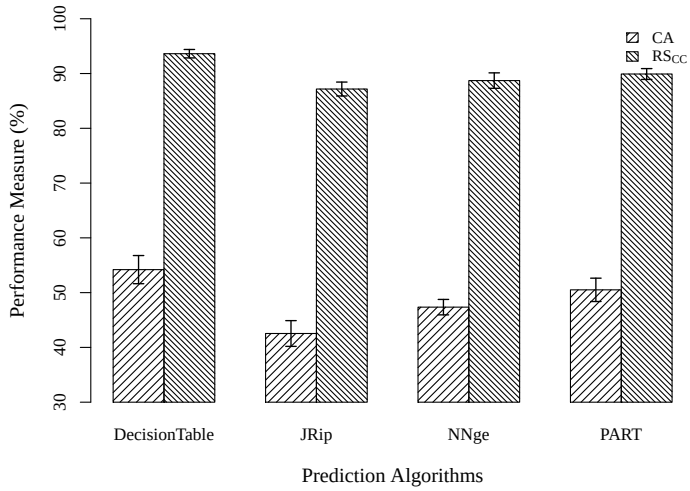


Figure 5.9: Performance of the prediction algorithms on the *Abalone* dataset

remains the same in the Breakout dataset. This causes the dataset to have less random input-output relationships than before. However, in the case of real output dataset, there are distinctive relationships between the input and output, causing the output probability distribution to be non-uniform. Thus the prediction model DecisionTable performs the best, with a *RS* of 74%.

5.5.2 Experiments with Commonly Available Datasets

Abalone Dataset

The Abalone dataset is used to determine the age of an abalone from physical measurements such as length of the shell, diameter and height [99]. The input features are not sufficient to determine the age of an abalone deterministically. This causes each instance in the dataset to have more than one possible output. In this dataset, there is no abstract feature that can be identified among all the features. Hence, RS_{CC} is a more suitable metric to evaluate the performance of the prediction algorithms on the abalone dataset. The dataset contains one categorical input and seven numerical input features. For computing RS values,

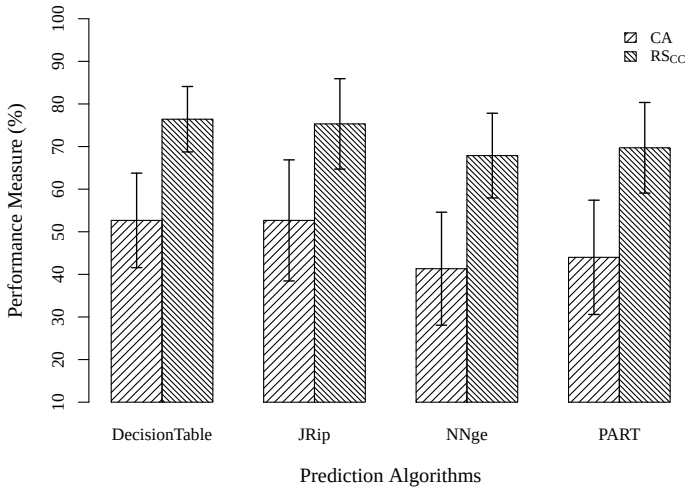


Figure 5.10: Performance of the prediction algorithms on the *Iris* dataset (Modified)

each numerical data type is divided into equal size pieces to obtain categorical data types. The range of input features; Length, Diameter, Height, Whole Weight, Shucked Weight, Viscera Weight and Shell Weight are divided into 10, 10, 4, 6, 6, 5 and 5 categories, respectively. The categories are formed based on the pattern of features visualized in WEKA [60].

Fig. 5.9 shows the performance of the prediction algorithms on the Abalone dataset including both CA and RS_{CC} as metrics. The performance with CA as a metric is very low, indicating that the prediction algorithms are not good in predicting the exact actual output. This is expected due to the one-to-many nature of the dataset. However, the performance measured using RS_{CC} shows that the prediction algorithms are indeed capable of predicting relevant outcomes. Also, the DT prediction algorithm provides an RS of 93%. This clearly shows that CA fails to capture the characteristic of the dataset and hence may lead to wrong conclusions while evaluating prediction algorithms. Meanwhile, RS provides more relevant results which makes conventional ML more effective.

Iris Dataset (Modified)

The Iris dataset is used to determine the type of Iris plant using features such as the petal length and width [46]. This dataset contains 150 samples with three output classes of 50 samples each. We use this dataset as another example to demonstrate the performance of CA and RS metrics when randomness is introduced artificially. We replace the outcomes of each class (20 samples each) to make the input-output relationship non-deterministic. This modified Iris dataset contains four numerical input data types. Each numerical data type is divided into equal size to obtain categorical data types for computing RS values. The range of input features; Sepal Length, Sepal Width, Petal Length and Petal Width are divided into 7, 8, 5 and 5 categories respectively. The categories are formed based on the pattern of features visualized in WEKA [60]. RS_{CC} is a suitable metric to evaluate the performance of the prediction algorithms as there is no feature that can be identified as abstract one.

The original Iris dataset leads to an accuracy of 95.3% with the C4.5 prediction algorithm [77] and 92% with DT. Considering the modified Iris dataset, Fig. 5.10 shows the performance of four prediction algorithms, using CA and the RS_{CC} as metrics. It can be seen that the CA performance of the prediction algorithms for the Iris dataset (modified) reduces to 53%, thereby reflecting the one-to-many relationship between the input and output. By contrast, RS leads to an average 25% improvement over CA. From this experiment, it is evident that when more than one output may satisfy a given context, RS metric is more appropriate than CA.

5.6 Discussion

From the performed study, we find that CA is not an appropriate metric for the mentioned class of non-deterministic applications. Metrics such as precision and recall are successfully used in binary classification problems, whereby the context should be predicted as either relevant or irrelevant (when calculating precision and recall). However, in applications such as intelligent lighting, there are contexts that may also be moderately relevant, slightly irrelevant, very relevant, absolutely irrelevant, and so forth. This problem becomes complicated when a predicted output that is relevant to one context may not be relevant to another. Therefore, in such applications, precision and recall are not suitable, while RS is more appropriate. Furthermore, two RS variants (RS_{CC} and RS_{Gen}) can be used to select prediction algorithms. The RS metric also provides a pro-

vision to select the α and β parameters to select prediction algorithms based on the need of the application. For example, a higher value of α is chosen (say $\alpha = 10$ and $\beta = 1$) for applications where capturing inconsistencies in the output for an observed context is not so critical. A small value of α is chosen (say $\alpha = 1$ and $\beta = 100$) for applications where it is necessary to select a prediction algorithm that is highly sensitive to the inconsistencies in the output.

It should be noted that RS is only a metric to evaluate the performance of a prediction algorithm. It does not influence its decision-making process. Moreover, it is difficult to develop a prediction algorithm that can give maximum RS performance. This is because RS is computed based on the information gained from an entire dataset. This means that the output distributions for all the observed contexts are known while computing RS. However, the prediction algorithms are trained on the training set and hence have information of the output distribution only for the training set but not for the entire dataset.

5.7 Conclusions

In ML applications such as intelligent lighting there is no single output that may be entirely right for a given context. In such cases, the dataset is said to have a one-to-many or non-deterministic relationship between input and output. In such cases, the performance evaluation of ML algorithms using commonly used metrics such as CA shows poor performance, which may be misleading. This is because, since more than one output may satisfy a given context, the prediction algorithm may fail in selecting the exact deterministic output. Yet, it is more appropriate to measure how relevant a given prediction is, rather than measuring accuracy. Therefore, we need a metric that can make the distinction between a totally irrelevant prediction and a relevant prediction when the prediction is not 100% accurate.

In this direction, we have devised a metric dubbed *Relevance Score*. The RS metric evaluates the performance of a prediction algorithm by the relevance of the predicted output rather than considering its accuracy. We further introduced two variants of RS; Relevance Score - Case-by-Case (RS_{CC}) and Relevance Score - General (RS_{Gen}) that can be used to study various aspects of a dataset. The RS for a predicted output and a given context is computed based on the posterior probabilities computed from all samples in a dataset. The benefits of the RS metric is presented through experiments on the *Wine* dataset and examples from an intelligent lighting application. The significance of the RS metric is demonstrated by various experiments on a dataset for intelligent lighting and

two other datasets obtained from the UCI repository. The parameters (α, β) to compute RS can be selected carefully to find the best suitable prediction algorithm based on the application requirement. Finally, we conclude that RS is an appropriate performance metric in the context of intelligent lighting where factors such as variable human perceptions lead to non-deterministic multiple output problems.

CHAPTER 6

Handling Missing Data in Intelligent Lighting

6.1 Introduction

Several practical applications, such as image processing, text classification and also intelligent lighting involve decision making based on knowledge gathered from data processing. However, it is common to have missing data in datasets due to sensor failures, equipment errors and lack of rich interfaces. Missing data may lead to performance degradation of applications, thereby motivating the need for an appropriate missing-data treatment.

In intelligent lighting, missing data occurs due to the lack of necessary interfaces that the users can use to interact with the lighting application. As discussed in Chapter 3, the values for input entries in the dataset are gathered from the breakout area both implicitly, by means of sensors that are deployed in the breakout area, and explicitly, by direct input from the users through an application installed on their smart phones. The users that do not use a smart phone interface can utilize a tangible user interface, i.e. an interaction cube, to interact with the system to select one of the available light presets. This causes missing values in the dataset for the two explicitly gathered features; User Identity (UID) and Type of Activity (ToA).

A suitable missing-data treatment may help improving the performance of the prediction algorithms. However, performance depends not only on the missing-data treatment method, but also on several factors such as the type of pattern present in the dataset, the very nature of missing data and the kind

of prediction algorithm used. All this makes it difficult to select the right combination of treatment method and prediction algorithm, which can achieve better performance over other approaches and for various amounts of missing data, without proper experimentation.

In this chapter, we describe the missing-data problem in intelligent lighting. We, then discuss some of the missing-data treatment approaches in the literature. We study different probability based treatment methods for handling the missing data for intelligent lighting. We use probability based methods because of the non-deterministic input-output relationships in intelligent lighting. The impact of these approaches on various amounts of missing data is investigated through in-depth experimentation using various rule-based prediction algorithms and the results are analyzed thoroughly using relevant statistical tests.

6.2 Problem Formulation

In this section, we formalize the missing-data problem. The following assumptions are made, the data collected implicitly via sensors do not have any missing data in them and the users with the breakout application installed on their smart phones do not use the tangible interface to interact with the system.

Let the input and output space be formalized as in Chapter 5. Let P denote the set of missing-data treatment methods and H denote the set of rule-based prediction algorithms under consideration. The percentage of samples with missing feature values is given by m ($0 \leq m \leq 100$). Various amounts of missing values $m \in M = \{m_1, m_2, \ldots, m_L\}$ are introduced into the Breakout dataset for the purpose of experimentation. The prediction performance of a selected (p, h) pair for treating the missing data is measured using *(RS_{CC})* and *(RS_{Gen})* as metrics.

Within this general setting, the following problems are investigated:

1. To analyze the impact of missing-data treatment methods in combination with different prediction algorithms for various values of m. Precisely, to analyze the RS performance of each (p, h) pair before and after the missing-data treatment.

2. To determine a (p, h) pair that gives maximum RS performance for any amount of missing data.

6.3 Related Work

The solution to address the missing-data problem may depend on the nature of missing data, i.e. whether the type of missing data is numeric, categorical or mixed; and/or the missing-data pattern, e.g. Missing Completely at Random (MCAR) or Missing at Random (MAR) [65]. There are also approaches that are independent of the nature of missing data, e.g. single and multiple imputation methods. Most of these studies are quite general, i.e. they consider several datasets of different patterns and collectively analyze the results [45]. Therefore, selecting a treatment approach for the missing-data problem in intelligent lighting becomes difficult.

Lakshminarayanan et al. [82] used various approaches to solve the missing-data problem. These are classified into two categories, data-driven and model-based. Machine-learning approaches (supervised and unsupervised) were used to treat the missing-data values. One technique involves modeling the data by supervised induction of a decision tree-based prediction algorithm. The second technique involves statistical imputation procedures. The experiments are performed on a specific industrial dataset wherein the input-output relationship is not known.

Acuna et al. [3] studied the missing-data treatment using four methods: *case deletion* (discarding samples with missing values); *mean imputation* (fill the missing feature values with the mean of all non-missing values of that feature); *median imputation* (fill the missing feature values with the median of all non-missing values of that feature); and *k-nearest-neighbor (kNN)* (fill the missing feature values considering a given number of instances that are most similar to the instance of interest). These methods were evaluated using two different classification techniques, *linear discriminant analysis (LDA)* and *kNN* classifier for twelve datasets. The results show that the treatment of missing data does not provide significant improvement in performance. They focussed mainly on comparison on different treatment methods. In their work, they considered only small amounts of missing data, i.e. between 0-21%. Furthermore, the amounts of missing data considered is not uniform (for example, 1%, 5%, 9%), which makes it difficult to analyze the influence of their missing-data treatment in the context of intelligent lighting and for a given amount of missing data.

Alireza et al. [45] investigated the influence of various missing-data treatment methods across different datasets. The methods include five single imputation methods: *mean imputation* (fill the missing feature values with the mean of all non-missing values of that feature); *the Hot-Deck method* (for a given sample with missing value, the most similar sample is found and the missing

values are filled from that sample); *the Naïve-Bayes method* (probability based technique); two improved methods based on *Hot-Deck* and *Naïve-Bayes* and one multiple imputation method based on *polytomous regression*. The classification accuracy of these techniques is then evaluated for six popular classifiers (RIPPER, C4.5, k-nearest-neighbor, support vector machine with polynomial and RBF kernels, and Naïve-Bayes) on 15 datasets from the UCI ML repository. The results show that, given a prediction algorithm, there is no unique efficient method to handle missing data for different amounts of missing data. In their work, the missing data and its subsequent treatment is considered only on training sets. However, in intelligent lighting, test sets (runtime data) also contain missing data values that need to be treated before prediction. This makes it is difficult to select a combination of treatment method and a prediction algorithm without experimentation.

Maytal-Saar et al. [121] studied the treatment of missing data through *reduced models* and compared the results with other two families of imputation methods, *(predictive) value imputation* (fill the missing feature values with a value using statistical methods such as mean and median) and *distribution-based imputation* (fill the missing feature values based on estimating the conditional distribution of the missing values) of different datasets. They introduced the concept of reduced-feature models to avoid imputation. The models are constructed based on a subset of features from the training data. The results show that the reduced-feature models give better performance than any imputation methods. However, these models are used less in practice as they are computationally expensive.

Xiaofeng et al. [135] have proposed estimation techniques for a mixed-attribute dataset that is comprised of both continuous and discrete feature values. They proposed a novel mixture-kernel (linear combination of two single kernel functions) based iterative estimator to treat the missing data in mixed-attribute datasets. The experiments are performed on several datasets from the UCI repository and the results are compared with existing treatment methods for various percentages of missing data. The performance of the proposed techniques has been measured using accuracy and root mean square error metrics. However, machine learning algorithms have not been used to investigate the prediction performance.

In general, the studies were performed using datasets that have a deterministic input-output relationship. However, datasets similar to that of intelligent lighting application have not been studied. Also, in most studies, many datasets have been used and the results are combined making it difficult to select one of the missing-data treatment methods with a prediction algorithm. Gopalakrishna

et al. [54] have investigated the problem using probability based approaches and rule-based prediction algorithms for the intelligent lighting application. The results showed that there is no single pair of treatment approach with a prediction algorithm that is best suitable for different amounts of missing data considered in the dataset. Typically, the pattern in the dataset (order of samples) has an influence on the prediction performance. In [54], the missing-data problem is investigated only for one dataset. This does not necessarily mean that the selected methods will provide the same performance for a differently patterned dataset. Moreover, the hold-out method was used to estimate the performance of the missing-data treatment whereas a cross-validation method is more suitable for datasets with a small number of samples [12].

6.4 Treatment of Missing Data

In this section, the experimental procedure followed for the missing-data treatment is introduced. Subsequently, probability based approaches to handle the missing data are discussed. The notations used to describe the datasets are summarized in Table 6.1.

6.4.1 Experimental Procedure

The experimental procedure followed for the treatment of missing data and its performance evaluation is shown in Fig. 6.1.

1. D' is created synthetically by removing values from UID and ToA features in the dataset D. The process of introducing missing values is discussed in Section 6.5.

Table 6.1: Notations of the datasets used in the experimental procedure

Notation	Description
D	Dataset without missing values
D_{train}	Training Set
D_{test}	Test Set
D'	Dataset with missing values
D'_{train}	Training set with missing values
D'_{test}	Test set with missing values
D''_{train}	Training set with missing values treated
D''_{test}	Test set with missing values treated

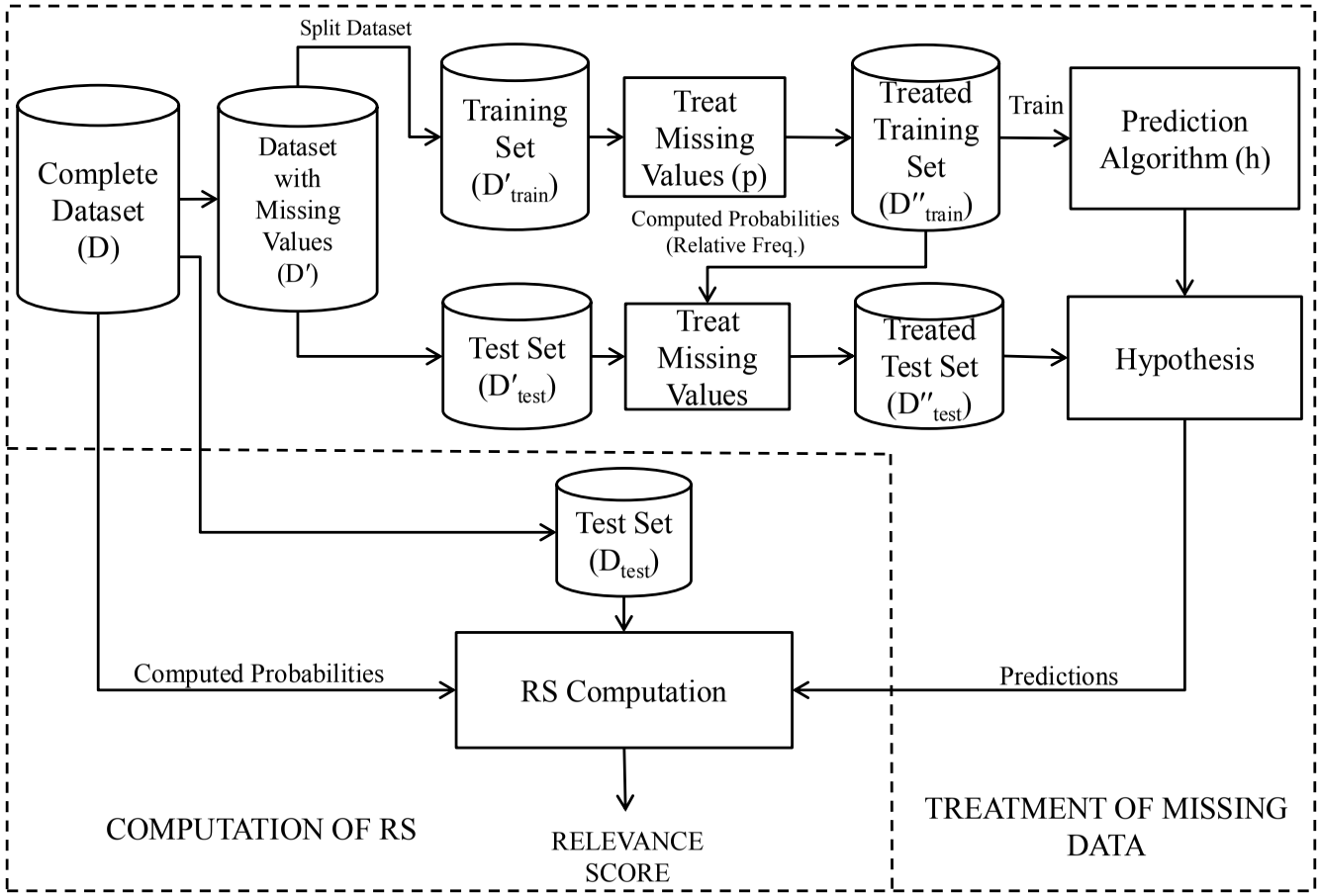


Figure 6.1: Experimental procedure followed for the missing-data treatment and its performance evaluation

2. The resulting D' is then divided into training set D'_{train} and test set D'_{test}.

3. The missing values in the D'_{train} are treated using one of the probabilistic approaches (p) to obtain D''_{train} as discussed in Section 6.4.3.

4. A prediction algorithm (h) is then trained using D''_{train}.

5. The missing values in the D'_{test} are then filled using feature values based on the relative frequencies of occurrence computed from the D''_{train}. (The different approach used in treating the missing values in the D'_{test} is because some of the missing-data treatment approaches for the D'_{train} involve the output class. This means that the knowledge from the samples without missing data is used to treat missing data in D'_{train}. However, samples in D'_{test} represent data at runtime and hence the desired output class is not known. This is indeed what needs to be predicted by the prediction algorithm after treating the missing values if any.)

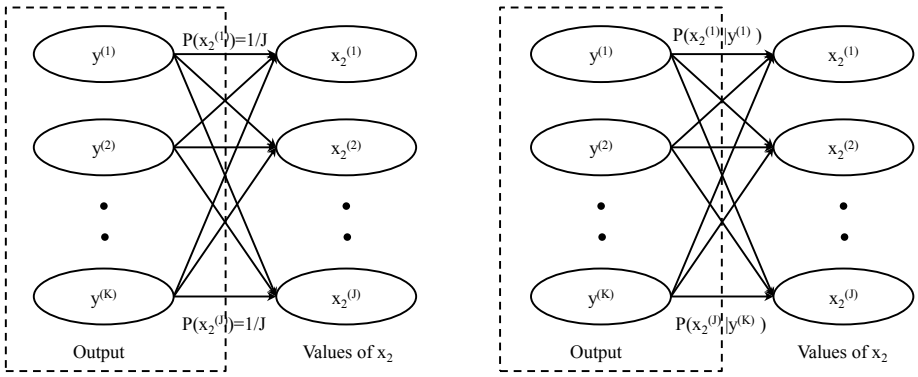
6. Subsequently, predictions are made for the contexts in D''_{test}.

7. The RS for a selected (p, h) pair is then computed as explained in Section 6.4.2.

8. The process is repeated for 10 different D'_{train} and D'_{test} generated using 10-fold cross-validation and also using different amounts of missing data with different (p, h) combinations.

6.4.2 Computation of Relevance Score

The RS metric provides a quantitative measure of how relevant a prediction is, for a given context (e.g. UID, ToA and ExLI). In the case of missing data in intelligent lighting, the predictions are made for the context in D''_{test}. However, the feature values in D''_{test} do not represent the true context whereas those in D_{test} do. This is because the treated values by D''_{test} may not be the same as their true values. For a given context, the RS is computed based on the true context as represented in D_{test} and various parameters such as predicted and actual output and their associated probabilities [55] that are calculated from D. Therefore, RS is computed using D_{test}, the predicted outputs for the contexts from D''_{test} and various probabilities computed from D.



(a) **(b)**

Figure 6.2: Pictorial representation of the *EP* and *CPO* method

6.4.3 Probabilistic Approaches

In this section, we discuss different probabilistic approaches for treating the missing data in D'_{train}. As mentioned earlier, there are two feature values (for UID and ToA) that may be missing. For every missing value for the feature UID, an additional category of user, i.e. $Unknown$ is assigned so that the patterns in the samples without missing values are not disturbed. For the i^{th} sample, if the feature UID value (x_{i1}) is missing, then $x_{i1} = Unknown$ is assigned. Similarly, for every missing value of the feature ToA (x_{i2}), a feature value needs to be assigned from the set $X_2 = \{x_2^{(1)}, x_2^{(2)}, \ldots, x_2^{(j)}, \ldots, x_2^{(J)}\}$. In the problem formulated here for the intelligent lighting, the ToA feature can take four different values ($J = 4$), i.e. X_2={Active_Group ($x_2^{(1)}$), Active_Alone ($x_2^{(2)}$), Relax_Group ($x_2^{(3)}$), Relax_Alone ($x_2^{(4)}$)}. The following methods are used to determine the value of $x_{i2} \in X_2$.

Equal distribution of Probabilities (EP)

In this method, a missing feature value x_{i2} is randomly assigned a value from the set X_2 assuming uniform probability distribution. Consider that x_{i2} is missing and the output is $y^{(k)}$ ($1 \leq k \leq K$). Then the missing entry is filled by $x_2^{(j)}$ with probability $1/J$, regardless of y. Figure 6.2a gives the pictorial representation of this method.

Conditional Probability based on the Outcome (CPO)

In this method, a missing value x_{i2} in the training set is assigned based on the conditional probability $P(x_2^{(j)}|y^{(k)})$. This conditional probability is computed over those entries (x, y) in the dataset that have no missing data using the relative frequencies of occurrence.

For example, if x_{i2} is missing and the output is $y_i^{(k)}$, then the missing entry is filled by $x_2^{(j)}$ with probability $P(x_2^{(j)}|y^{(k)})$ computed over the samples in the training set that have no missing data. If this conditional probability is equal to zero, then the EP method is used instead to fill in the missing value. Figure 6.2b gives the pictorial representation of this method.

Conditional Probability based on the Outcome and other Features (CPOF)

In this method, a missing value x_{i2} in the training set is assigned based on the conditional probability $P(x_2^{(j)}|x_{i3}, x_{i4}, x_{i5}, x_{i6}, y_i^{(k)})$. This conditional probability is computed over those entries (x, y) in the dataset that have no missing data using the relative frequencies of occurrence.

For example, if x_{i2} is missing and the output is $y_i^{(k)}$, then the missing entry is filled by $x_2^{(j)}$ with probability $P(x_2^{(j)}|x_{i3}, \ldots, x_{i6}, y^{(k)})$ computed over the samples in the training set that have no missing data. If this conditional probability is equal to zero, then the EP method is used instead to fill in the missing value.

Conditional Probability based only on other Features (CPF)

This method is similar to method CPOF, but the dependence on the outcome y_i is removed during the computation of conditional probability i.e. a missing value x_{i2} in the training set is assigned based on the conditional probability $P(x_2^{(j)}|x_{i3}, x_{i4}, x_{i5}, x_{i6})$.

Conditional Probability based only on other Features - modified (CPFm)

This method is similar to the method CPF, but the missing feature values in the test set are not filled as described in Section 6.4.3. Once the missing values in the training data is addressed, as in method CPF, $P(x_2^{(j)}|x_{i3}, \ldots, x_{i6})$ is recomputed over D''_{train} using relative frequencies. Then the missing values in the test set are filled based on this probability. If the conditional probability is equal to zero, then the EP method is used instead to fill in the missing value.

6.5 Experiments and Results

The performance of the probability-based missing-data treatment methods is evaluated using WEKA. The main objectives of the experiments are to analyze the impact of missing-data treatment and find a (p, h) pair that gives maximum prediction performance across any amounts of missing data in the Breakout dataset. We consider the following six rule-based prediction algorithms, DecisionTable (DT), JRip, Neareast Neighbor with generalization (NNge), PART, ConjunctiveRule and Ridor available in WEKA. The experiments are conducted with m = 10%, 20%, 30%, 40% and 50% of missing data. The prediction performance of a selected (p, h) pair on a given dataset is measured using RS_{CC} and RS_{Gen} as metrics. The RS values are computed using 10-fold cross-validation. We perform the following step-by-step analysis to achieve the mentioned objectives.

1. The datasets with missing values (D' instances) used in the experiments are described.
2. The experimental results of the missing-data treatment are analyzed.
3. The influence of missing-data treatment is studied using a *test of statistical significance*.

6.5.1 Datasets with Missing Values

The problem of missing data arises, when the context is known entirely i.e. when values for some features are missing. This happens explicitly, when a user does not use a smart-phone interface to interact with the breakout area. This creates missing values for features UID and ToA. In the Breakout dataset, the values for features UID and ToA are removed from every sample of the selected users to produce D' instances. Five different D' are considered for each value of m. Table 6.2 gives the UID of the users selected from which the values for the features UID and ToA were removed.

6.5.2 Experimental Results with RS_{CC} as a metric

The RS_{CC} value computed for a selected (p, h) pair and a given m is denoted by $(RS_{CC(avg)})$. It is determined by averaging the RS_{CC} values computed from the missing-data treatment over five D' instances. Subsequently, the standard

Table 6.2: UID of the users selected from which the values for the features UID and ToA were removed to produce *datasets with missing values*

Missing Data Amounts (m%)	Datasets				
	D1	D2	D3	D4	D5
10	U1, U5	U3, U9	U6, U7	U10, U11	U15, U16
20	U1-U2, U4	U3-U6, U9	U8-U10	U13	U14-U16
30	U1-U6	U5-U10	U1, U5-U6, U9-U11	U13-U14	U10-U12, U14-U16
40	U1-U4, U6-U8	U7-U12, U14	U1-U2, U4, U13	U5-U6, U8-U9, U14-U16	U13-U16
50	U1-U9	U1-U6, U14-U16	U7-U13	U1, U3, U5, U7, U9, U11, U13	U10-U16

Table 6.3: Summary of *Method (p), Prediction Model (h)* with best *Relevance Score - Case-by-Case* values

Missing Data Amounts (%)	Method (p)	Prediction Algorithm (h)	$RS_{CC(avg)}$ (%)
10	CPFm	DT	52.24
20	No Treatment	NNge	51.47
30	No Treatment	DT	50.34
40	CPO	NNge	47.11
50	No Treatment	NNge	45.03

deviation ($RS_{CC(std)}$) is computed to determine the deviation of RS_{CC} values from the $RS_{CC(avg)}$. We analyze the following experimental results.

1. Relevance Score - Case-by-Case ($RS_{CC(avg)}$)
2. Improvement in Relevance Score - Case-by-Case ($RS_{CC(imp)}$) after the missing-data treatment
3. Standard deviation of the Relevance Scores ($RS_{CC(std)}$)

Table 6.4: Summary of *Method (p), Prediction Algorithm (h)* with best *Improvement of Relevance Score - Case-by-Case* and their *Relevance Scores*

Missing Data Amounts (m%)	Method (p)	Prediction Algorithm (h)	$RS_{CC(imp)}$ (%)	$RS_{CC(avg)}$ (%)
10	CPFm	DT	1.64	52.24
20	CPO	Ridor	0.96	42.14
30	CPO	Ridor	1.16	40.45
40	CPO	Ridor	3.39	39.37
50	CPO	Ridor	6.28	40.48

Relevance Score - Case-by-Case $(RS_{CC(avg)})$

Table 6.3 summarizes the (p, h) combinations that gives the best $RS_{CC(avg)}$ for considered m values. The results show that different combinations of (p, h) provide the maximum $RS_{CC(avg)}$ across different m values. The NNge algorithm gives the maximum RS_{CC} performance in most cases except for the datasets with 10% and 30% missing data. Its performance for these cases is close to the best performing algorithm (DT). Even though the NNge algorithm does not give the best $RS_{CC(avg)}$ values in all the cases, it is still the suitable candidate as it gives maximum performance in most cases.

Improvement in *Relevance Score - Case-by-Case* $(RS_{CC(imp)})$

The *Improvement in* RS_{CC} $(RS_{CC(imp)})$ for a given (p, h) pair and m is the difference of the RS_{CC} values computed over the D''_{test} and D'_{test}. Table 6.4 summarizes the (p, h) pairs that give the best $RS_{CC(imp)}$ and the corresponding $RS_{CC(avg)}$. The results show that the CPO treatment method gives the maximum $RS_{CC(imp)}$ with the Ridor algorithm in most cases except for the datasets with 10% missing data. In the case of 10% missing data, $RS_{CC(imp)}$ is still positive for $(CPO, Ridor)$ pair, but it is not better than the $(CPFm, DT)$ pair. The results show that there is no single (p, h) pair that provides the maximum $RS_{CC(imp)}$ across all values of m considered. However, the maximum $RS_{CC(imp)}$ does not necessarily mean that maximum RS_{CC} performance is achieved. This means that even though the pair $(CPO, Ridor)$ gives the maximum $RS_{CC(imp)}$ in most cases, its corresponding $RS_{CC(avg)}$ is not superior to other combinations.

Table 6.5: The minimum and maximum standard deviation values of the Relevance Scores - General from five datasets, six prediction algorithms, five missing-data treatment methods and five different amounts of missing data. The worst performing (p, h) pairs are highlighted in *italics* for each case

Missing Data Amounts (%)	Method (p)	Prediction Algorithm (h)	$RS_{CC(imp)}$	Standard Deviation
10	CPF	DT	0.34	0.25
	$CPOF$	$Ridor$	*0.88*	*3.50*
20	CPO	ConjunctiveRule	0.04	0.94
	CPO	DT	*-1.88*	*3.59*
30	CPFm	JRip	-2.77	0.92
	CPO	$Ridor$	*1.16*	*4.75*
40	CPFm	PART	-3.23	1.04
	CPF	DT	*-0.23*	*6.00*
50	CPO	DT	0.45	0.90
	CPF	$PART$	*-2.58*	*3.83*

Standard deviation of the Relevance Scores - Case-by-Case $(RS_{CC(std)})$

Table 6.5 gives the minimum and maximum standard deviation of the RS_{CC} values $(RS_{CC(std)})$ and their corresponding $RS_{CC(imp)}$ for each considered m. The worst results are highlighted in italics. The $RS_{CC(std)}$ value shows the consistency of RS_{CC} performance for a (p, h) pair over a given dataset. The $RS_{CC(std)}$ value depends on the (p, h) pair, value of m and the pattern of missing data. The results show that there is no single (p, h) pair that has the lowest $RS_{CC(std)}$ value across all values of m considered. Note that the NNge algorithm, which gives the maximum $RS_{CC(avg)}$ does not appear among the minimum $RS_{CC(std)}$ values.

From the experiments performed, a (p, h) pair for implementation can be selected based on $RS_{CC(imp)}$, $RS_{CC(avg)}$ or $RS_{CC(std)}$ depending on the application requirement. The results show that these aspects also depend on the amount of missing data and the pattern of missing data. There is no unique combination of (p, h) that gives best performance in all the considered studies.

Generally, it is desirable to have a high prediction performance i.e. $RS_{CC(avg)}$ that makes the NNge algorithm the most suitable prediction algorithm. Figure 6.3 shows the graph of $RS_{CC(avg)}$ vs. $RS_{CC(std)}$ for the NNge algorithm with the missing-data treatment methods and the values of m. In general, treatment

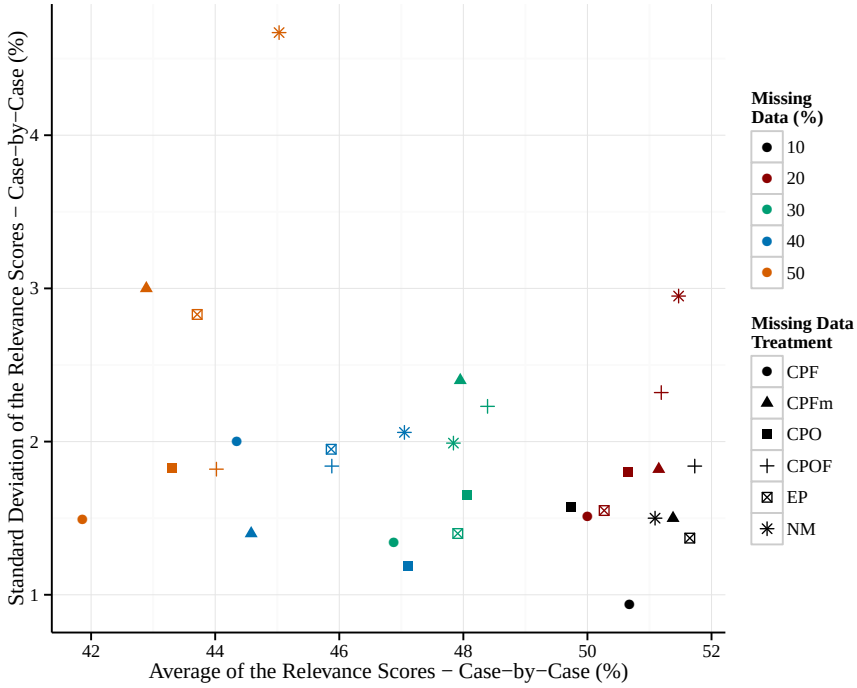


Figure 6.3: Average vs. standard deviation of the Relevance Scores - Case-by-Case for the *NNge* prediction algorithm resulting from the missing-data treatment of five datasets, five missing-data treatment methods and five different amounts of missing data

methods that have high $RS_{CC(avg)}$ also have high $RS_{CC(std)}$. Therefore, if the application demands good RS_{CC} performance, the $(CPOF, NNge)$ is the most suitable combination. If the application demands good RS_{CC} performance but also low $RS_{CC(std)}$ values, then the $(CPO, NNge)$ is the most suitable pair.

6.5.3 Impact of Missing-Data Treatment for the NNge Prediction Algorithm

It is important to determine whether the missing-data treatment methods have an impact on the performance of prediction algorithms. For this purpose, we

Table 6.6: Statistical significance of the Improvement of Relevance Scores - Case-by-Case for the NNge prediction algorithm and five different amounts of missing data

Missing Data Amounts (%)	Treatment Method (p)	t-value	p-value
10	CPO	-1.20	0.30
20	CPF	-1.73	0.16
30	CPF	-1.29	0.27
40	CPF	-2.08	0.11
50	CPF	-2.15	0.10

use the paired t-test that makes use of the RS_{CC} performance values of the prediction algorithms before and after the missing-data treatment (before = *RS without missing data treatment*). The paired tests are hypothesis tests that are used when each member of one numerical set can be paired up with a particular member of the other set [35]. In simple terms, they are used to investigate whether a new technique or method works better than an existing method, without worrying about other factors that influence the results. In our experiment, the null hypothesis H_o is defined as $RS_{CC(imp)} = 0$ i.e. there is no increase or decrease in RS_{CC} performance after the missing-data treatment. The alternative hypothesis H_a is defined as $RS_{CC(imp)} \neq 0$. The process of paired t-test for a given (p, h) pair and m yields a t-value and p-value. The t-value indicates the test statistic where a positive value means that $RS_{CC(avg)}$ improved after the missing-data treatment, while a negative value means that $RS_{CC(avg)}$ reduced. The p-value is a number between 0 and 1 that indicates the strength of evidence against the hypothesis being tested. We perform the paired t-test at the 95% significance level. The p-value indicates the probability of observing the t-value and in this case the p-value less than 0.05 means that $RS_{CC(avg)}$ is statistically significant. In other words, it is unlikely that $RS_{CC(imp)}$ has occurred by chance, i.e. there is strong evidence against H_o, and hence it can be rejected. If p-value is greater than 0.05, it is very likely that $RS_{CC(imp)}$ has occurred by chance, i.e. there is weak evidence against H_o, and hence it cannot be rejected. Table 6.6 summarizes the t-values and the least p-values of the paired t-test for the NNge prediction algorithm and for each m.

The results show that for all considered values of m, the t-values are negative, indicating that there is no improvement in RS_{CC} performance after the

Table 6.7: Summary of *Method (p), Prediction Model (h)* with best *Relevance Score - General* values

Missing Data Amounts (%)	Method (p)	Prediction Model (h)	$RS_{Gen(avg)}$ (%)
10	CPFm	DT	75.86
20	No Treatment	DT	74.93
30	No Treatment	DT	73.52
40	No Treatment	DT	69.91
50	CPF	DT	67.53

missing-data treatment. Moreover, the treatment methods with the NNge algorithm do not have impact on the RS_{CC} performance as indicated by the p-values. Here, the p-values greater than 0.05 indicate that there is weak evidence against the H_o. From the analysis, we conclude that there is no significant influence of the missing-data treatment methods on the RS_{CC} performance with the NNge algorithm and hence NNge is missing-data resistant.

From the results, we see that no (p, h) pair provides the maximum $RS_{CC(imp)}$ or the maximum $RS_{CC(avg)}$. However, the NNge algorithm provides the best $RS_{CC(avg)}$ among the considered prediction algorithms in most cases of m considered. The statistical test performed on $RS_{CC(imp)}$ for NNge shows that none of the considered treatment methods has a significant impact on the RS_{CC} performance.

6.5.4 Experimental Results with RS_{Gen} as a metric

The RS_{Gen} value for a selected (p, h) pair and a given m is denoted by $(RS_{Gen(avg)})$. It is determined by averaging the RS_{Gen} values computed from the missing-data treatment over five D' instances. Subsequently, the standard deviation $(RS_{Gen(std)})$ is computed to determine the deviation of RS_{Gen} values from the $RS_{Gen(avg)}$. We analyze the following experimental results.

1. Relevance Score - General $(RS_{Gen(avg)})$
2. Improvement in Relevance Score - General $(RS_{Gen(imp)})$ after the missing-data treatment
3. Standard deviation of the Relevance Scores $(RS_{Gen(std)})$

Table 6.8: Summary of *Method (p), Prediction Model (h)* with best *Improvement of Relevance Score - General* and their *Relevance Scores*

Missing Data Amounts (%)	Method (p)	Prediction Model (h)	$RS_{Gen(imp)}$ (%)	$RS_{Gen(avg)}$ (%)
10	CPFm	DT	0.81	75.86
20	CPO	Ridor	1.10	67.38
30	CPO	ConjunctiveRule	0.50	65.85
40	CPO	Ridor	1.90	64.63
50	CPO	Ridor	4.72	64.58

Relevance Score - General $(RS_{Gen(avg)})$

Table 6.7 summarizes the (p, h) combinations that give the best $RS_{Gen(avg)}$ for various values of m. The DT algorithm provides the best $RS_{Gen(avg)}$ without the missing-data treatment in most cases except for 10% and 50% of missing data. However, the $RS_{Gen(imp)}$ is not above 1% in these cases. From this study, we conclude that even though there is no single (p, h) pair that gives the best $RS_{Gen(avg)}$ across different m, DT is the most suitable prediction algorithm among the other considered algorithms as it gives maximum performance in most cases.

Improvement in *Relevance Score - General* $(RS_{Gen(imp)})$

The *Improvement in* RS_{Gen} $(RS_{Gen(imp)})$ for a given (p, h) pair and m is the difference of the RS_{Gen} values computed over D''_{test} and D'_{test}. Table 6.8 summarizes the (p, h) pairs that gives the best $RS_{Gen(imp)}$ and the corresponding $RS_{Gen(avg)}$. The CPO treatment method gives the maximum $RS_{Gen(imp)}$ with the Ridor algorithm in most cases except for datasets with 10% and 30% missing data. In the case of 10% and 30% missing data, the $RS_{Gen(imp)}$ is still positive for the $(CPO, Ridor)$ pair in these cases, but not the best. The results show that there is no single (p, h) pair that provides the maximum $RS_{Gen(imp)}$ across all values of m considered.

Table 6.9: The minimum and maximum standard deviation values of the Relevance Scores - General from five datasets, six prediction algorithms, five missing-data treatment methods and five different amounts of missing data. The worst performing (p, h) pairs are highlighted in *italics* for each case.

Missing Data Amounts (%)	Method (p)	Prediction Algorithm (h)	$RS_{Gen(imp)}$	Standard Deviation
10	CPF	DecisionTable	-0.31	0.53
	$CPOF$	$JRip$	*-0.43*	*2.77*
20	CPO	ConjunctiveRule	0.42	0.34
	$CPFm$	$Ridor$	*-0.27*	*3.05*
30	CPFm	Ridor	-1.06	0.59
	CPO	$Ridor$	*0.20*	*3.43*
40	CPF	Ridor	-1.02	0.99
	EP	$DecisionTable$	*-2.28*	*4.10*
50	CPFm	ConjunctiveRule	-1.33	0.33
	$CPOF$	$DecisionTable$	*-0.18*	*3.75*

Standard deviation of the Relevance Scores - General $(RS_{Gen(std)})$

Table 6.9 gives the minimum and maximum standard deviation of the RS_{Gen} values $(RS_{Gen(std)})$ and their corresponding $RS_{Gen(imp)}$ for each considered m. The worst results are highlighted in italics. The results show that there is no single (p, h) pair that has the lowest $RS_{Gen(std)}$ value across all values of m considered. The Ridor and ConjunctiveRule algorithms have the best $RS_{Gen(std)}$ performance in two cases each. The DT algorithm that gives the best $RS_{Gen(avg)}$ has the lowest $RS_{Gen(std)}$ value in case of 10%, whereas it has the highest values in case of 40% and 50%. From the experiments performed, there is no unique combination of (p, h) that has the lowest $RS_{Gen(std)}$ values for all considered m.

Based on the $RS_{Gen(avg)}$ performance, we select the DT algorithm as the most suitable choice. Figure 6.4 shows the graph of $RS_{Gen(avg)}$ vs. $RS_{Gen(std)}$ for the DT algorithm with the missing-data treatment methods and the values of m. Unlike the case of RS_{CC}, there is no single treatment method that gives a good performance of either $RS_{Gen(avg)}$ or $RS_{Gen(std)}$. It can also be seen that in most cases, the DT prediction algorithm gives the best RS_{Gen} performance without the missing-data treatment.

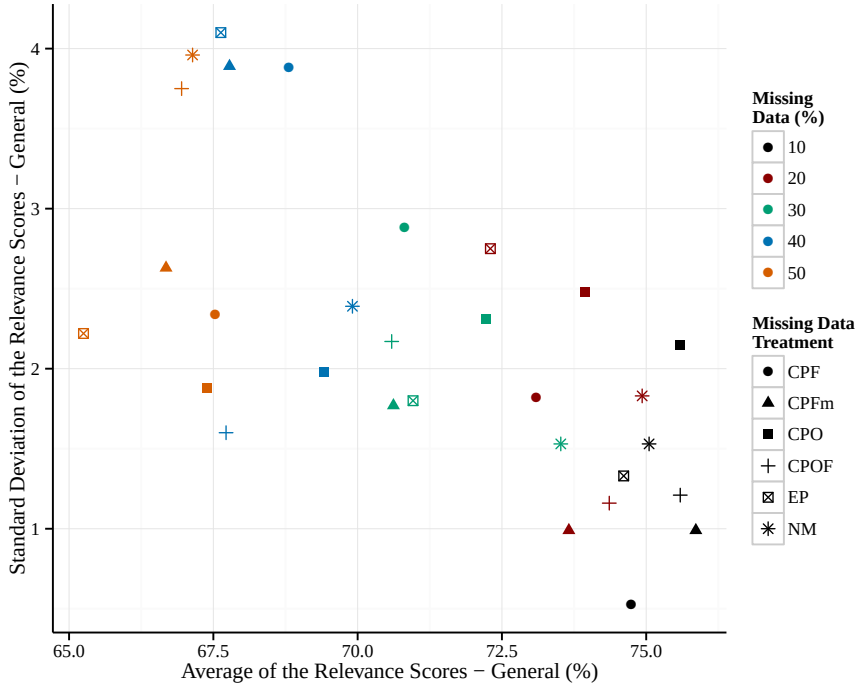


Figure 6.4: Average vs. standard deviation of the Relevance Scores - General for the *Decision Table* classification model resulting from the missing-data treatment of five datasets, five probabilistic treatment approaches and five amounts of missing data

6.5.5 Impact of Missing-Data Treatment for the DT Prediction Algorithm

As in the case of RS_{CC}, we performed the paired t-test to investigate the impact of missing-data treatment on the RS_{Gen} performance of the prediction algorithms. Here, the null hypothesis H_o is defined as $RS_{Gen(imp)} = 0$ i.e. there is no increase or decrease in the RS_{Gen} performance after the missing-data treatment. The alternative hypothesis H_a is defined as $RS_{Gen(imp)} \neq 0$. We perform the paired t-test at the 95% significance level. Table 6.10 summarizes the t-values and p-values of the paired t-test that have least or statistically significant p-values, for the DT prediction algorithm and for each m. The statistically sig-

Table 6.10: Statistical significance of the difference between classification accuracies using dataset with *missing data treated* and *missing data* for each of the five treatment methods (columns), five amounts of missing data (rows) and the *Decision Table* classification model

Missing Data Amounts (%)	Treatment Method (p)	t-value	p-value
10	CPFm	1.23	0.29
20	**EP**	**-2.95**	**0.00**
	CPF	**-22.99**	**0.00**
30	**EP**	**-7.23**	**0.00**
	CPOF	**-6.21**	**0.00**
	CPFm	**-3.50**	**0.02**
40	**CPOF**	**-4.77**	**0.00**
50	EP	-1.40	0.12

nificant values $p < 0.05$ are highlighted in bold i.e. when there is negative improvement after the missing-data treatment.

The results show that in most cases, the t-values are negative, indicating that there is no improvement in the RS_{Gen} performance after the missing-data treatment except for 10%. However, the p-value for 10% missing data shows that the $RS_{Gen(imp)}$ is not statistically significant. In case of 20% missing data, the p-values show that the $RS_{Gen(imp)}$ values are very significant for the EP and CPF methods. For 30%, it is the EP, CPOF and CPFm methods and for 40% missing data, it is the CPOF method. The p-values being much less than 0.05 indicate that there is strong evidence that the RS_{Gen} values will reduce if missing data are treated. This means that it is very unlikely that the negative RS_{Gen} performance have occurred by chance, and hence the H_o can be rejected. From the analysis, we conclude that the missing-data treatment has negative impact on the RS_{Gen} performance in the cases of 20%, 30% and 40% missing data for certain treatment methods. Moreover, the p-values do not indicate significant improvement in the cases of 10% and 50% missing data. Therefore, we conclude that the RS_{Gen} performance of the DT algorithm is better without the missing-data treatment.

From the experimental results, we have seen that the missing-data treatment does not provide significant improvement in prediction performance of

Table 6.11: The *Relevance Score* performance of the NNge and DecisionTable prediction algorithms without the missing-data treatment for five amounts of missing data

Missing Data Amounts (%)	RS_{CC}			RS_{Gen}		
	NNge	DT	Difference	NNge	DT	Difference
10	51.09	50.60	0.49	70.06	75.05	-4.98
20	51.47	51.21	0.25	69.91	74.93	-5.02
30	47.84	50.34	-2.50	68.60	73.52	-4.92
40	47.05	44.83	2.22	67.14	69.91	-2.77
50	45.03	42.25	2.78	66.26	67.74	-1.48

the prediction algorithms. In case of RS_{CC} as a metric, we have NNge as the best prediction algorithm and in case of RS_{Gen}, we have the DT algorithm. Table 6.11 compares the performance of the NNge and DT algorithms, without the missing-data treatment and with the RS_{CC} and RS_{Gen} metrics. It can be observed that the performance of the DT algorithm is much better using RS_{CC} than the performance of the NNge algorithm using RS_{Gen}. This means that, in general, the performance of DT is better than NNge. Hence, DT is the most suitable supervised prediction algorithm for the Breakout dataset, among the prediction algorithms considered.

6.6 Conclusion

The need to address the missing-data problem in intelligent room lighting implementations is motivated by the occurrence of missing data due to the lack of rich interfaces in the Breakout dataset. Based on the non-deterministic multiple-output nature of the dataset, it is not straightforward to select a treatment method in combination with a prediction algorithm that gives best prediction performance for different amounts of missing data present.

In this direction, several probability-based missing-data treatment methods were presented. Subsequently, their impact were investigated using various existing rule-based prediction algorithms. A comprehensive experimental study has been performed to evaluate the five probability based treatment methods in combination with six different rule-based prediction algorithms, for five different amounts of missing data (i.e., 10%, 20%, 30%, 40%, and 50%) in the

Breakout dataset. The experiments have been conducted using both RS_{CC} and RS_{Gen} as metrics. Various aspects of the results have been analyzed and the impact of missing-data treatment has been studied through statistical hypothesis test (paired t-test).

The selection of a missing-data treatment method and the prediction algorithm for implementation depends on the requirement of prediction performance from applications such as the improvement, stability or the best performance. The experimental results show that there is no single (p, h) pair that gives the best performance across various aspects and different amounts of missing data considered. The studies also show that the prediction performance of missing-data treatment depends on several factors such as the (p, h) pair, value of m and the dataset considered. In general, maximum prediction performance is desired. The NNge and DT prediction algorithms provided the best performance for most values of missing data with RS_{CC} and RS_{Gen}, respectively. The paired t-test is performed after the missing-data treatment, to study whether the improvement achieved occurred by chance or not. The significance analysis with 95% significance level showed that the improvement in prediction performance after the missing-data treatment is not significant, for both the prediction algorithms. Finally, we conclude that even though the missing-data treatment improves the prediction performance of some (p, h) pairs, the desired objective of achieving maximum performance is difficult to achieve using a single (p, h) pair. Therefore, DT is the most suitable prediction algorithm among the considered prediction algorithms on the Breakout dataset without the missing-data treatment.

CHAPTER 7

Instance based Learning for Intelligent Lighting

7.1 Introduction

Implementing intelligent lighting using supervised learning has several drawbacks. The reasons are as follows. Supervised learning algorithms are implemented in two phases. In the first phase, the prediction algorithm is trained using the collected data samples. In the second one, the trained algorithm is used to predict an output class for the new inputs. A prediction algorithm can be effective in predicting right outputs, if the training samples sufficiently represent the input space. The input space is best represented if the training samples are diverse and the training set is sufficiently large. In the intelligent lighting application considered in this thesis, the input space is of six dimensions and the feature User Identity can possibly have infinite values. Therefore, a large number of diverse samples would be needed to at least approximate the input space. Collecting a large number of data samples is a very difficult task, as the users would need to use the intelligent lighting system intensively.

Apart from good prediction performance, the selection of a learning approach to implement intelligent lighting also depends on whether the system should learn and adapt continuously or not. Supervised learning models are parametric models, meaning that the data is assumed to be derived from a known distribution [32], which can be summarized using a finite number of parameters. The performance of a supervised prediction algorithm also depends on whether the observed pattern in the dataset remains same or not. For exam-

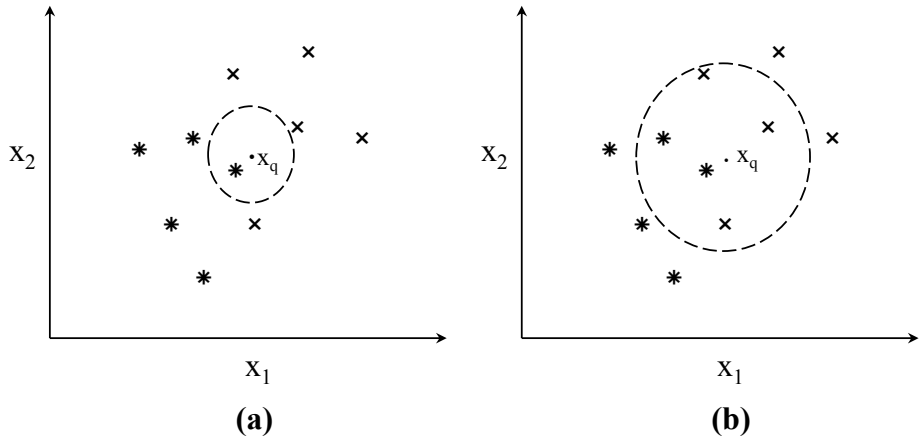


Figure 7.1: Demonstration of k-Nearest Neighbor algorithm on a set of $\times$ and $*$ samples with context x_q to be predicted
a. 1-Nearest Neighbor algorithm predicts x_q as $*$,
b. 5-Nearest Neighbor algorithm predicts x_q as $\times$

ple, the feature External Light Influence is influenced differently towards north and south poles in different seasons, that in turn affects the distribution in the dataset. Also, human factors such as mood and perception play a key role in the output selection. This means that a fixed distribution cannot be assumed for intelligent lighting. Supervised prediction algorithms do not adapt as the input-output relationships change in time due to dynamic factors such as changing user preferences and varying lengths of daytime in different seasons. Therefore, even when the prediction algorithm is trained on a particular dataset, it is difficult to guarantee a sustained performance over the time, as the variable distribution changes. This has motivated us to explore non-parametric models such as instance-based and online learning algorithms. Non-parametric models are those models that do not assume any fixed distribution [80] and hence can adapt continuously to unknown or changing distributions.

In this chapter, we study the instance-based learning algorithm, k-Nearest Neighbor (kNN), in the context of intelligent lighting. We first introduce kNN and then discuss the variants of kNN algorithms considered in our experiments. We perform experiments on the Breakout dataset, analyze and compare the results to those achieved with the supervised learning algorithm, DecisionTable (DT) (Chapter 5).

7.2 k-Nearest Neighbor Algorithm

kNN is the most popular instance-based algorithm as it is simple and efficient. It can be used both as a batch learner (as in supervised learning) and as an online learner. In batch learning mode, it works as follows; similarly to supervised learning, there is a large set of samples that form the training set. The samples are in the form of input-output pairs. When a new context or input arrives, for which the output class label needs to be determined, the new context is compared to every context in the training set. The most similar contexts (nearest neighbors) with their output class labels are taken. The integer value k, determines the number of nearest neighbors to consider. Finally, from the k nearest neighbors, the output class label that occurs the most number of times (majority voting) is assigned as the output class label for the new context.

Formalizing kNN algorithm: Let $X = X_1 \times X_2 \times X_3 \times \ldots \times X_n$ denote an n-dimensional input feature space where the i^{th} sample $x_i \in X$ is given by $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Let y_i denote the output class label where $y_i \in Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(P)}\}$ and P is the number of possible outcomes. Let m denote the number of samples in the training set i.e., $i = 1, 2, \ldots, m$. The goal is to determine the output class label for a new context x_q. The set of k nearest neighbors from the training set to x_q is determined by computing the distances from each training sample (d_i and $i = 1, 2, \ldots, m$). One of the following distance functions are commonly used to compute the distance between two contexts; 1) Manhattan $(d_1(x_i, x_q) = \sum_{j=1}^{n} |x_{ij} - x_{qj}|)$, 2) Euclidean $(d_2(x_i, x_q) = \sqrt{\sum_{j=1}^{n}(x_{ij} - x_{qj})^2})$, or 3) Minkowski $(d_z(x_i, x_q) = \sqrt[z]{\sum_{j=1}^{n}(x_{ij} - x_{qj})^z})$ where $z > 2$ [75]. The output class label y_q is assigned as follows; $y_q = \underset{y_i}{\operatorname{argmax}}(\#(y_i))$ for $i = 1, 2, \ldots, k$.

Consider an example application with two input features x_1 and x_2 and two output class labels $*$ and $\times$. Let the dataset have 10 samples and be distributed as shown in Fig. 7.1a. Let x_q be the new context for which the output class needs to be determined. In case 1, let the value of k be 1 as shown in Fig. 7.1a. The distances from all the 10 contexts to x_q are computed. The output class label $*$ is assigned to x_q as it is the output class label of the 1-nearest neighbor. In case 2, let the value of k be 5 as shown in Fig. 7.1b. In this case, the output class label that will be assigned to x_q is $\times$. This is because $\times$ appears the most, i.e. three times, among the 5 nearest neighbors of x_q ($\times$ wins majority voting).

Table 7.1: k-Nearest Neighbor Methods used in our Intelligent Lighting evaluation

Method	Description
k(1)	kNN with $k = 1$
k(15)	kNN with $k = 15$
Fwk(1)	Feature weighted kNN with $k = 1$
Fwk(15)	Feature weighted kNN with $k = 15$
Dwk(15)	Distance weighted kNN with $k = 15$
DFwk(15)	Distance and Feature weighted kNN with $k = 15$

7.3 k-Nearest Neighbor in Intelligent Lighting

In the case of intelligent lighting, we have six input features where the *Time of the Day* (ToD) is of numerical data type and the rest are of categorical data type. However, the features *Intensity of Users in other subarea* (IoU) and *External Light Influence* (ExLi) can be considered as numerical data types as well, since their feature values can be ordered. The range of feature IoU is [0, 10] where 0 represents no activity and 10 represent maximum activity in the area, and that of ExLI is [0, 3] where 0 represents the feature value *VeryLow* and 3 represents *VeryHigh*. Now, we have three features of categorical data type and three features of numerical data type.

We use Manhattan Distance to compute the distance between two contexts as it provides better Classification Accuracy (CA) performance than the Euclidean Distance. It is straightforward to compute Manhattan Distance (as in Section 7.2) for contexts with numerical features. However, for categorical features, we assign the distance between two different contexts as 1 if the compared features are not equal, otherwise it is 0. The distance computed between two contexts cannot be compared fairly, if we have different scales for the features of numerical data type. Therefore, we normalize the distance computed between two numerical features as follows. The *normalized* distance for a feature is the ratio of the *absolute* distance and the *range* for that feature. The absolute distance is the actual difference between the numerical features of two different contexts. The range for features ToD, IoU and ExLI are [0,24], [0,10] and [0,3], respectively.

For our experiments, we use the following methods and values for k as summarized in Table 7.1. The number of nearest neighbors (value of k) are indi-

cated inside braces.

k-Nearest Neighbor (k=1): We use the kNN algorithm with $k = 1$, i.e. one nearest neighbor ($k(1)$). In this case, the test context x_q will be assigned an output class label of the closest nearest neighbor.

k-Nearest Neighbor (k=15): In kNN, the value for k is usually selected as the $\sqrt{No.OfTrainingSamples}$ [37]. Since we use 10-fold cross-validation for our experiments, the $No.OfTrainingSamples = 0.9 \times TotalNo.OfSamples = 0.9 \times 236 \simeq 212$. Therefore, the value of $k = \sqrt{212} \simeq 15$ ($k(15)$).

Feature weighted kNN (Fwk): In general, for a context, the considered input features do not contribute equally in determining the output class. Some features have more significant influence than the others. The presence of irrelevant features will cause the performance of kNN to degrade [70]. If these features are known, higher weights can be assigned to them, while computing the distance between the contexts. This helps in reducing the influence of irrelevant features. In Chapter 4, we saw that the feature *Type of Activity* (ToA) is the most important feature for intelligent lighting to determine relevant output lighting conditions. Hence, in this method, we double the computed distance for the ToA feature between two contexts, i.e. if the ToA features do not match then we take the distance as 2 instead of 1. We select the weight for the ToA feature as 2 because the prediction performance of the kNN algorithm reduces for weights greater than 2. The Fwk algorithm uses $k = 1$ and $k = 15$.

Distance weighted kNN (Dwk): In kNN, equal weights are given to the distances computed from a new context x_q to the k-nearest neighbors. However, the output class labels to be assigned to the test context x_q may be influenced by the distances of its nearest neighbors, i.e. the closest neighbors may contribute more than the farther ones. This has motivated the need for applying weights to the computed distances between the contexts. For our experiments, we have considered applying weights for the distances from [38]. The weights on the distances are assigned as follows. If the distance of the k-nearest neighbors to a new context are ordered from closest to farthest as d_j where $j = 1, 2, \ldots, k$, the weight w_j attributed to the j-th nearest neighbor is defined by

$$w_j = \begin{cases} \dfrac{d_k - d_j}{d_k - d_1}, & d_k \neq d_1 \\ 1 & d_k = d_1 \end{cases}$$

Once the weights w_j are computed, the algorithm then assigns the new context an output class label, for which the weights among the k-nearest neighbors sum to the largest value. The Dwk algorithm is used with $k = 15$ (does not

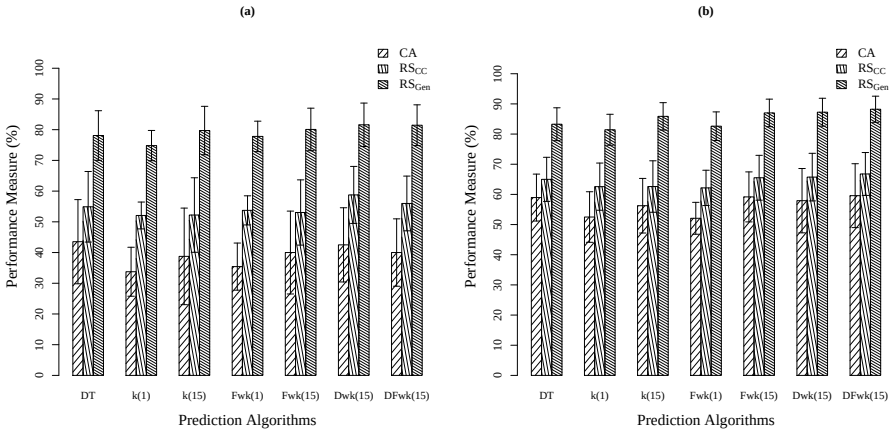


Figure 7.2: Performance of the prediction algorithms on the Breakout dataset
a. With 8-output lighting conditions,
b. With 4-output lighting conditions

make sense to use with $k = 1$).

Distance and Feature weighted kNN (DFwk): In this case, we combine the Fwk and the Dwk algorithms. First, we apply weights to the features (feature ToA for intelligent lighting) to reduce the influence of irrelevant features on the prediction performance. Next, we apply weights to the distances of the k-nearest neighbors to select a suitable output lighting condition. Like the Dwk algorithm, the DFwk algorithm is used with $k = 15$.

7.4 Experiments and Results

We perform experiments to investigate the prediction performance of kNN algorithms in intelligent lighting and compare against a supervised learning algorithm. From our experiments with the supervised learning algorithms in Chapter 5, we have seen that the DT rule-based prediction algorithm is the most suitable algorithm for intelligent lighting. Hence, we compare the performance of kNN algorithms with that of the DT algorithm. The experiments are performed on the Breakout Dataset using CA, Relevance Score - Case-by-Case (RS_{CC}) and Relevance Score - General (RS_{Gen}) as metrics. The results are obtained through 10-fold cross validation.

Table 7.2: Standard deviation (SD) of the performance for the prediction algorithms for 8-output lighting conditions

Prediction Algorithm	SD (CA)	SD (RS_{CC})	SD (RS_{Gen})
DecisionTable	13.7	11.49	8.07
k(1)	7.97	4.37	4.93
k(15)	15.72	12.13	7.88
Fwk(1)	7.67	4.74	4.96
Fwk(15)	13.49	10.62	6.92
Dwk(15)	12.08	9.28	7.07
DFwk(15)	10.97	8.91	6.64

7.4.1 Analysis of the Performance of kNN algorithms on the Breakout dataset with 8-outputs

Fig. 7.2 shows the performance of the variants of kNN algorithms and the DT algorithm for the Breakout dataset with 8 output lighting conditions. On average, the DT algorithm provides maximum CA performance (43.5%) among the considered prediction algorithms, meaning that the DT algorithm predicts more accurately. However, the difference of the CA performance with Dwk(15) is 1%, which is not significant. For the RS_{CC} metric, the Dwk(15) algorithm provides the maximum performance compared to other algorithms (3% better than DFwk(15) and 4% better than DT). In case of the RS_{Gen} metric, the Dwk(15) and DFwk(15) algorithms give maximum performance. The results for the RS metrics show that the Dwk(15) performs better (around 4%) than that of the DT algorithm. This means that the Dwk(15) algorithm provides most relevant predictions as compared to that of the DT algorithm. However, for all the three considered metrics, we can observe that the error bars overlap indicating that the difference in the performance between the prediction algorithms is not significant.

7.4.2 Analysis of the Performance of kNN algorithms on the Breakout dataset with 4-outputs

In this experiment, static and dynamic lighting conditions are combined, resulting in a total of 4-output classes. This is done because the users were not able to differentiate between the static and dynamic settings [104]. Fig. 7.2b shows

the performance of the variants of kNN algorithms and the DT algorithm for the Breakout dataset with 4 output lighting conditions. It can be seen that since the output space is reduced, the performance of prediction algorithms improve significantly as compared to the results of 8-output dataset. For the CA metric, we observe that the Fwk(15) and DFwk(15) algorithms provide the similar performance as that of the DT algorithms. Also, the CA performance of the Dwk(15) is close to that of the Fwk(15) and DFwk(15) algorithms. For the RS_{CC} metric, on average, the DFwk(15) provides the best performance than other algorithms. However, the difference of RS_{CC} values with that of the DT algorithms is 2% and with that of the Fwk(15) and Dwk(15) algorithms is 1%. For the RS_{Gen} metric, the DFwk(15) gives the best performance. Again, the RS_{Gen} performance of the Fwk(15) and Dwk(15) algorithms are close to that of the DFwk(15). The difference of RS_{Gen} values between the DFwk(15) and DT algorithms is high (5%). However, as in the case of 8-output lighting conditions, we can observe that the error bars overlap indicating that the difference in the performance between the prediction algorithms is not significant.

7.4.3 Dataset Dependency of the Prediction Algorithms

Table 7.2 shows the standard deviation of the performance of the variants of kNN algorithms and the DT algorithm computed using 10-fold cross validation. The values show that there is a high degree of inconsistency in the prediction performance for the metrics considered. These values indicate that the performance of the prediction algorithms varies significantly with different training and test sets. We observe that the standard deviation values are high for most prediction algorithms except for kNN algorithms with $k = 1$. This means that when $k = 1$, the prediction performance is less dependent on the dataset and is more consistent whereas when $k = 15$, the prediction performance depends significantly on the dataset.

From this study, we find that under batch learning setting, the kNN algorithms provide similar performance as that of the DT algorithm. Also, we observe that having more neighbors $(k = 15)$ represent the test context better than having one neighbor $(k = 1)$. Furthermore, the results show that there is significant influence of the closest neighbors than the farthest neighbors. This means that all the considered nearest neighbors do not contribute equally in assigning a class label to the test context. As the distance between the test context and neighbors increases, the neighbors tend to become less significant.

Parametric models such as supervised prediction algorithms are often being faster to use, but the main disadvantage is that they make stronger assumptions

about the nature of the data distributions [98]. If the parametric assumptions about the data does not hold, just like in the case of intelligent lighting, only non-parametric models such as instance-based and online learning algorithms are suitable [81]. Therefore, in Chapter 8, we study kNN algorithms in online setting and compare their performance to online learning algorithms.

7.5 Conclusions

In this chapter, we explored how the instance-based algorithm, *k-Nearest Neighbor* performs in the context of intelligent lighting. We have considered several variants of kNN algorithms based on the values of k, applying weights on the features and distances computed. To analyze the performance of the kNN algorithms, we have performed the experiments on the Breakout dataset, using CA, RS_{CC} and RS_{Gen} as metrics. From the obtained results, we observe that the kNN algorithms perform similar to that of DT supervised learning. Even though the number of test samples are too few to draw definite conclusions, the kNN algorithms are preferred for intelligent lighting. This is because, in intelligent lighting, we know that the training samples do not represent the entire input space. This means that the assumptions about the data distributions, made by the supervised predictions algorithms, do not hold. In such cases, statistical techniques such as instance-based algorithms that can be applied regardless of the true distribution of the data is preferred [81].

In intelligent lighting, learning needs to proceed online, using only the samples which are available at any particular time. A major disadvantage of instance-based learning algorithms is that they store the observed samples to predict future outputs. Furthermore, the time to predict output increases with the increasing observed samples. Therefore, we explore online learning algorithms for intelligent lighting in the next chapter and compare their prediction performance with that of instance-based learning algorithms (under online setting).

CHAPTER 8

Online Learning for Intelligent Lighting

8.1 Introduction

In online setting, instance-based learning algorithms adapt continuously over time. Hence, they are preferred over the supervised learning algorithms for intelligent lighting. However, they are not effective in terms of computational complexity and memory requirements. For a set of m samples with n-dimensional context space, calculating distance takes *O(mn)*time and $O(mn^2)$ including sorting (to identify k nearest neighbors). This is a disadvantage because the instance-based algorithms predict at the time when a context arrives, taking a significant processing time as the context needs to be compared with all the observed contexts [94]. They also need large memory as they store all the observed contexts. High-dimensional context spaces are a major drawback to implement instance-based algorithms due to the *curse of dimensionality* [94]. As the dimensionality increases the samples tend to become sparse, making the distance between contexts large. Consequently, the pair-wise distances become less significant. Hence, the sample size needed to maintain the same sampling density grows with increase in the number of dimensions. Since intelligent lighting is a multiple-output problem, adding more features to reduce the one-to-many input-output relationships, leads to more problems for instance-based algorithms. Also, these algorithms have other disadvantages such as they do not produce any useful representation (functions, rules or trees) from the data, as a result they are intolerant to noise in the features and they are sensitive to

the distance function used to measure similarities between contexts [4]. As a result, we explore *online* learning algorithms in this chapter.

Online learning algorithms are another class of machine learning algorithms where the contexts do not arrive as a batch but as a sequence and the algorithms must respond and learn from one context at a time [117]. Like instance-based learning, online learning algorithms also belong to non-parametric models that do not assume any particular underlying distribution. These algorithms take into account that at any point in time future contexts are unknown [5]. Moreover, unlike instance-based algorithms they do not store the observed samples, thereby making it efficient in terms of storage and computation requirements. Online learning algorithms retain the knowledge of the observed samples in the form of feature weights.

In this chapter, we discuss various online learning algorithms that we use for experiments on intelligent lighting. We then describe the process for computing the Relevance Score for online learning algorithms. Subsequently, we discuss the experimental results and compare the performance of online learning algorithms to that of k-Nearest Neighbor algorithms.

8.2 Online Learning Algorithms

Online learning algorithms operate on a sequence of data samples in consecutive steps. At each step i, the algorithm is given a context $x_i \in X$ where X is an n-dimensional input space and $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. The algorithm predicts an output class label $y_P \in Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(K)}\}$ as a function of the context x_i and weights w_i. Each weight w_i is an n-dimensional real-valued vector $(w_{i1}, w_{i2}, \ldots, w_{in})$ that determines the contribution of x_i $(x_{i1}, x_{i2}, \ldots, x_{in})$. These weights depend on the observed samples and their corresponding predicted outputs. After y_P is predicted, the algorithm receives the true output class label $y_A \in Y$. The quality of the y_P is then assessed by a loss function $l(y_P, y_A)$. Based on the result of $l(y_P, y_A)$, the algorithm modifies w_i based on an update function $\Delta = f(w_i, l(y_P, y_A))$ to predict better outputs for future contexts.

For our experiments, we have selected four online learning algorithms that are implemented in *LIBOL* (A Library for Online Algorithms) [63], based on their classification accuracy (CA) performance on the Breakout dataset. The algorithms are *Adaptive Regularization of Weights* (AROW), *Online Gradient Descent* (OGD), *Perceptron* and *Soft-Confidence Weighted* (SCW) algorithm. We use the default parameters implemented in LIBOL. Of the mentioned algorithms,

Perceptron and OGD are first-order algorithms whereas AROW and SCW are second-order algorithms. In multi-class prediction, online learning algorithms maintain classification functions for each output class label. First-order learning algorithms keep updating these functions using only the information from the input features. However, second-order online algorithms make use of underlying patterns in the data in addition to relationships among features.

8.2.1 Adaptive Regularization Of Weights

Crammer et al. [29] introduced AROW to address the need for fast convergence and resistance to training noise. The algorithm has several properties combined; large margin training, confidence weighting and the capacity to handle non-separable data. AROW performs adaptive regularization of the prediction function when processing each new context in each learning step. This makes it more robust to sudden changes in the prediction function due to noisy labels in the learning tasks. Noisy labels refer to a setting where the output class labels are corrupted [100]. The main idea is to retain the formalization of parameter confidence introduced by Confidence-Weighted (CW) [28], which is aggressive in learning. AROW removed the aggressiveness by softening the margin requirement thus achieving robustness to training noise without affecting the convergence speed.

8.2.2 Online Gradient Descent

Zinkevich [136] formulated the online learning problem as a convex optimization problem where a decision-maker (for example, a prediction algorithm) makes a sequence of decisions (predictions) from a fixed feasible set. After each decision, it observes a convex cost function. The objective is to make decisions that minimize the cost functions. OGD exploits the gradient descent updating approach for optimizing the objective function defined by different types of loss functions. This algorithm models many natural repeated decision-making problems and generalizes many existing problems such as *Prediction from Expert Advice*. The algorithm has several advantages. Firstly, the gradient descent is a simple, natural and widely used algorithm. Secondly, it can handle an arbitrary sequence of convex functions. Finally, in some cases, the algorithm can perform better than experts algorithm for online linear problems. In LIBOL several different types of losses have been implemented that can be selected by setting the parameter *loss type* (0 for 0-1 loss, 1 for hinge loss, 2 for logistic loss, and 3 for square loss). The default value is 1 that implements hinge loss [114].

8.2.3 Perceptron

Crammer et al. [30] extended the binary Perceptron algorithm [115] to a family of multiclass Perceptron algorithms. The algorithm maintains one prototype vector for each output class. When a new context arrives, the similarity-score is computed between the prototypes of each output class and the new context. The new context is then assigned an output class label that achieves the highest similarity-score. Once the correct output label is received, a set of prototypes corresponding to the *error-set* are updated. The *error-set* is a set of output class labels that have similarity-scores greater than the score of the correct label. This work also introduced the notion of *ultraconservativeness*. Generally online algorithms update their prediction rule only on rounds in which they make a prediction error. Such algorithms are called *conservative*. Ultraconservative algorithms are the algorithms that update only the prototypes corresponding to the error-set. In LIBOL, three variants of multiclass Perceptron algorithms are proposed; max-score multiclass Perceptron; uniform multiclass Perceptron; and proportion multiclass Perceptron that are based on different allocation strategies. For our experiments, we have used the uniform multiclass Perceptron (umP) as it provides the maximum CA performance on the Breakout dataset, than the max-score and the proportion multiclass Perceptrons.

8.2.4 Soft Confidence Weighted

Wang et al., proposed SCW [128] as an improvement over CW and AROW. As mentioned earlier, the CW learning method employs a very aggressive updating strategy, by changing the parameters sufficiently to satisfy the constraint imposed by the current sample. Even though this results in faster learning, it could wrongly change the parameters of the distribution while dealing with a noisy context. This can make the CW algorithm perform poorly in many real-world applications with relatively large noise. The SCW learning algorithm was proposed to overcome these limitations by relaxing the update strategy. Like the AROW learning algorithm, SCW has the following three properties; large margin training, confidence weighting, and capability to handle non-separable data. Additionally, SCW also employs adaptive margin. There are two variants of SCW learning that are denoted by *SCW-I* and *SCW-II* where SCW-II employs a squared penalty. We have used SCW-I for our experiments as it provides the maximum CA performance on the Breakout dataset, than SCW-II.

8.3 Relevance Score for Online Learning

As mentioned in Chapter 7, supervised learning algorithms can be effective, if the training samples are good representatives of an input space. This is very difficult to guarantee in intelligent lighting because of the large input space and limited dataset. Non-parametric models such as instance-based algorithms and online learning algorithms are preferred when the distribution from where the samples are drawn is unknown. Generally, the performance of an online learning algorithm is evaluated along its run on a sequence of question-answer pairs [117]. For intelligent lighting, we have seen that the RS metric is appropriate to measure the performance of machine learning algorithms for multi-output problems [55]. The RS metric is devised to evaluate the performance of the supervised learning algorithms. In this section, we modify the process of computing RS to evaluate online algorithms.

In supervised learning, RS is computed based on the knowledge acquired from the entire dataset. This means that the underlying distribution is known to the RS metric. However, in online learning, the samples arrive in sequence and there is no prior knowledge about the underlying distribution in the data. Therefore, to evaluate the predicted output for a new context, the posterior probabilities need to be recomputed when the actual output for that context is revealed. RS for online learning is computed in two phases; 1) computing posterior probabilities, and 2) evaluating the predicted output.

8.3.1 Computing Posterior Probabilities

Assume that for a context $x_q \in X$, an online algorithm predicts an output y_P, while the actual output is y_A where $y_P, y_A \in Y$. The posterior probabilities for the output class labels are then computed from the set of observed samples. Let $\{x_1, x_2, \ldots, x_m\}$ denote the set of observed samples where $x_m = x_q$, Compute $P(y^{(k)}|x_i)|x_i = x_q$ where $i = 1, 2, 3, \ldots, m$ and $k = 1, 2, \ldots, K$. These probabilities are used to compute $ErrScore$, a real number indicating the amount of mismatch between actual and predicted outputs. As in the case of supervised learning, the probability of occurrence of the predicted output is denoted by $P(y_P)$, the probability of occurrence of the actual output is denoted by $P(y_A)$ and the probability of the most frequently selected output is denoted by $P(y_H)$ are determined using the posterior probabilities.

8.3.2 Evaluating the Prediction

In this phase, the predicted output y_P is evaluated, similar as in supervised learning. The probabilistic distances between different outputs (d_{HP}, d_{PA}, d_{HA}) are computed. If y_P does not match y_A, the $ErrScore$ is computed using equation 8.1, where α and β are real parameters that depend on the requirements of the application.

$$ErrScore = \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta}. \tag{8.1}$$

For a context x_i, RS is thus computed as a function of $ErrScore$ as

$$RS_i = (1 - ErrScore_i) \times 100, \tag{8.2}$$

$$RS_i = (1 - \frac{\alpha(d_{HP(i)}) + \beta(d_{PA(i)})}{\alpha + \beta}) \times 100. \tag{8.3}$$

8.4 Experiments and Results

The performance of online learning algorithms is analyzed through learning curves. A learning curve shows a measure of predictive performance (for example, classification accuracy) on a given task as a function of some measure of varying amounts of learning effort such as time and number of samples [108]. Generally, in machine learning and statistics, a learning curve represents the generalization performance of the algorithm as a function of the number of observed samples (training set) [47], i.e. error measured on the test samples versus the number of observed samples. Learning curves also help in analyzing the performance of a learning algorithm over different numbers of observed samples. The crossing of learning curves indicates that some learning algorithms may perform better over another for a larger number of observed samples [107]. In our experiments, a learning curve represents the prediction performance of a learning algorithm as a function of the number of observed samples.

We explain the process of producing learning curves using Classification Accuracy (CA) as a metric. The process is repeated for Relevance Score - Case-by-Case (RS_{CC}) and Relevance Score - General (RS_{Gen}) as metrics in the same way. The RS metric is used with $\alpha = 1$ and $\beta = 1$. For each sample i in the Breakout dataset, we compute the CA of the predicted output (CA_i). For each sample i

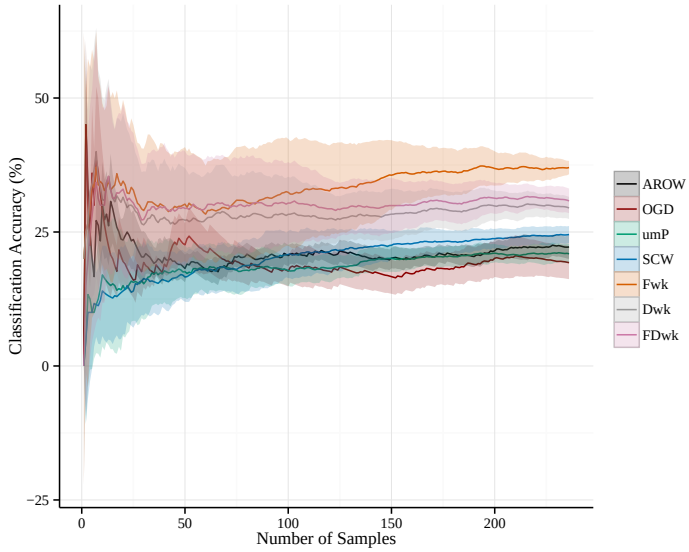


Figure 8.1: Learning curves of the Online learning and k-Nearest Neighbor algorithms for the Breakout dataset with *Classification Accuracy* as the metric

in the dataset, compute the average CA until the sample i written as $CA(i)$, as in equation 8.4.

$$CA(i) = \frac{\sum_{k=1}^{i} CA_k}{i} \tag{8.4}$$

$(CA(i))$ is then plotted as a function of i. In order to obtain a smooth learning curve, we repeat the process on 10 randomly shuffled datasets for each online learning algorithm and average the resulting curves. In Chapter 7, we have seen that the kNN algorithms, Feature weighted kNN (Fwk), Distance weighted kNN (Dwk) and Distance and Feature weighted kNN (DFwk) with $k = 15$ provide better prediction performance than other considered kNN algorithms. Therefore, we select Fwk, Dwk and DFwk algorithms to compare their prediction performance with that of the online learning algorithms. In online setting, we do not have any fixed training samples and a kNN algorithm is supposed to predict from the first context observed. Therefore, for practical purpose we do not have a fixed k value and the value of k is updated as the

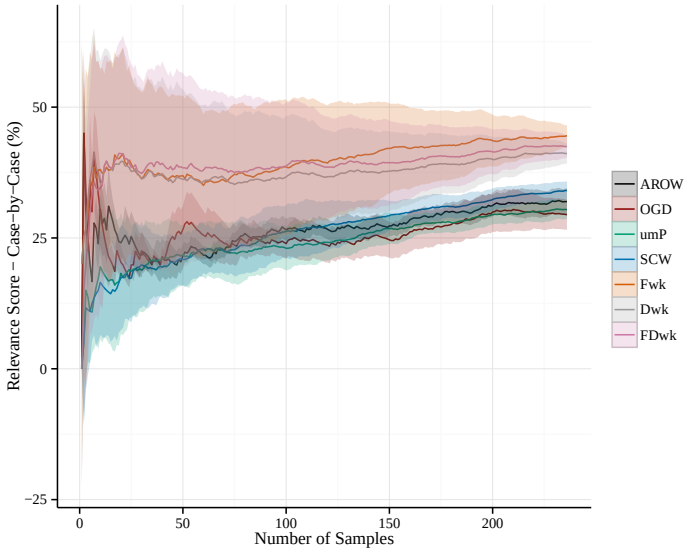


Figure 8.2: Learning curves of the Online learning and k-Nearest Neighbor algorithms for the Breakout dataset with *Relevance Score - Case-by-Case* as the metric

number of observed samples increases. For our experiments, we consider the value of k as the square root of the number of contexts observed [37], i.e. if at step i, a context x_i needs to be predicted then $k = \sqrt{i}$. For the Breakout dataset, the maximum value that k can reach is $sqrt(236) \simeq 15$.

8.4.1 Experiments on the *Breakout* dataset

Fig. 8.1 shows the learning curves for the online learning algorithms and the kNN algorithms with CA as the metric. The CA performance of the online learning algorithms is much lower compared to that of kNN algorithms. The CA performance of AROW and OGD online learning algorithms vary significantly as a function of the number of samples. However, the learning curves of the umP and SCW algorithms are smoother. It can be observed that after 70 samples, while the CA performance of umP almost converges, the CA performance of SCW still improves. Even though the initial learning rate is low for SCW, it consistently improves to provide the maximum performance among the con-

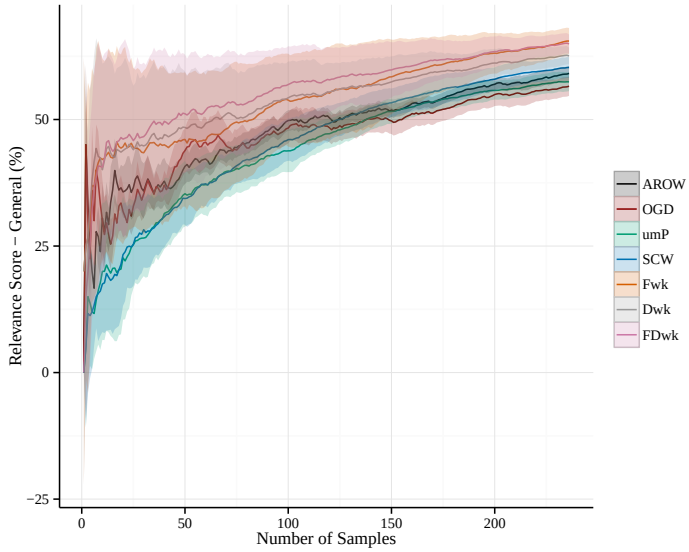


Figure 8.3: Learning curves of the Online learning and k-Nearest Neighbor algorithms for the Breakout dataset with *Relevance Score - General* as the metric

sidered online learning algorithms. In case of kNN algorithms, the Dwk and FDwk algorithms provide similar performance. Here also, after 70 samples the CA performance of Fwk becomes better than FDwk, while the performance of FDwk converges. The Fwk algorithm provides the maximum CA performance among all the considered prediction algorithms.

Fig. 8.2 shows the learning curves for the online learning algorithms and the kNN algorithms with RS_{CC} as the metric. As in the case of the CA metric, the RS_{CC} performance of online learning algorithms is much lower compared to that of kNN algorithms. The RS_{CC} performance of the learning algorithms are similar but higher than that of the CA performance. SCW gives the best RS_{CC} performance among the online learning algorithms and Fwk provides the maximum RS_{CC} performance among all the considered prediction algorithms. However, in case of kNN algorithms, the learning curve of the Fwk crosses that of FDwk at sample 110 indicating better RS_{CC} over FDwk on larger data. Also, we see that the sample size is not enough to observe the convergence of the performance of the considered prediction algorithms.

Fig. 8.3 shows the learning curves for the online learning algorithms and the kNN algorithms with RS_{Gen} as the metric. The RS_{Gen} performance of the online learning algorithms are lower compared to that of kNN algorithms. The learning curves of the prediction algorithms are smoother compared to that of the CA and RS_{CC}. RS_{Gen} is computed by ignoring *abstract* feature i.e. user-identity UID from the Breakout dataset. This means that to RS_{Gen} the dataset looks as if it is collected from a single user. The RS_{Gen} learning curves indicate that when there are large number of samples for each user, the RS_{CC} performance will also be smoother and keeps improving. Again, SCW gives the best RS_{Gen} performance among the online learning algorithms and Fwk provides the maximum RS_{Gen} performance among all the considered prediction algorithms. Here, the RS_{Gen} of Fwk becomes better than that of FDwk after 200 samples. As in the case of RS_{CC}, the sample size is not enough to observe the convergence of the performance of the considered prediction algorithms.

From this study, we observe that even though the considered prediction algorithms are not designed for the non-deterministic multiple output problems, the prediction algorithms continue learning as the number of observed samples increases. The learning curves using RS_{Gen} indicate that as more samples are collected from each user, the prediction gets better and consistent irrespective of the dataset. However, these algorithms cannot predict more accurately with increasing samples as seen in the learning curves using the CA metric. Also, we observe that the standard deviation of the prediction performance, which is very high initially, reduces as the more number of samples are observed, meaning that the performance becomes more consistent.This is because, in these types of problems, it is more appropriate to measure how relevant is the predicted output for the given context, than measuring whether it is accurate or not. Therefore, the experimental results using the RS metrics indicate that the prediction algorithms still learn and also provide better relevant outputs.

8.4.2 Experiments on the *Wine* dataset

As mentioned earlier, the number of samples in the Breakout dataset is not sufficient to understand the behavior of prediction algorithms in online setting for non-deterministic multiple output problems. In this direction, we performed experiments on the Wine (White) dataset available from the UCI machine learning repository. We have seen in Chapter 5 that the Wine dataset also has one-to-many input-output relationships. In order to apply RS, we need the input features to be of categorical data type whereas the input features in the Wine dataset are of numerical data type. Thus, for our experiments, we consider two

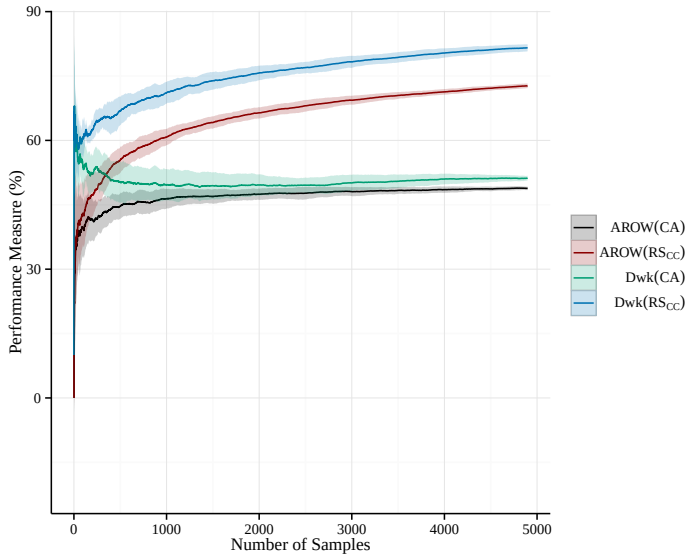


Figure 8.4: Learning curves of the AROW and Dwk prediction algorithms for the Wine dataset with each feature having 5 categories

cases. In each case we divide each input feature into 5 and 3 equal categories. We denote the corresponding datasets by W_5 and W_3, respectively. The output class labels are unaltered. We select the Dwk instance-based algorithm for our experiments. This is because the Fwk and DFwk algorithms can only be used when important features in the Wine dataset are known. Among the online learning algorithms, we select AROW based on their CA and RS_{CC} performance. We do not consider the RS_{Gen} metric, as there are no abstract features identified among the input features in the Wine dataset.

As we have seen in Chapter 5, W_3 has more uncertainty between the input and output than in W_5. This is because by reducing the number of categories for each input feature from 5 to 3, we increase the overlapping of the input space. It becomes more difficult for a prediction algorithm to predict the exact output for a given input, when uncertainty increases. Fig. 8.4 shows the learning curves of the considered prediction algorithms for W_5. While the CA performance of the prediction algorithms converge after 1000 samples, the RS_{CC} performance still keeps improving even after 4500 samples. This shows that for non-deterministic problems, prediction algorithms can provide more relevant output with more

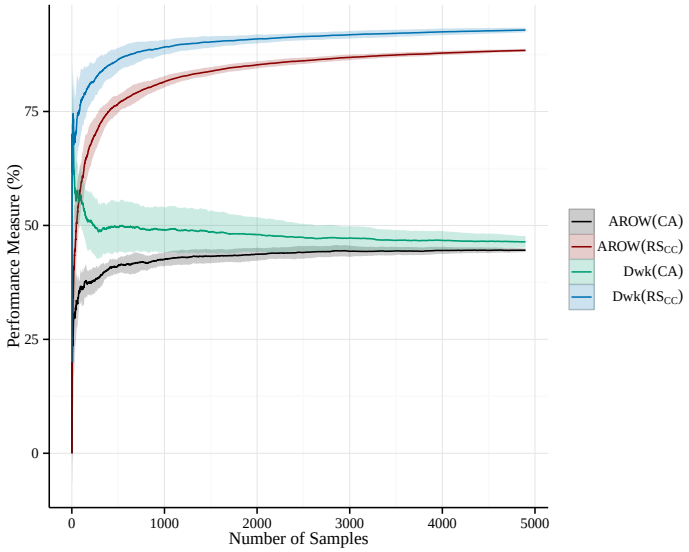


Figure 8.5: Learning curves of the AROW and Dwk prediction algorithms for the Wine dataset with each feature having 3 categories

number of observed samples, although their accuracy does not improve after a certain number of samples. As in the case of the Breakout dataset, the Dwk algorithm predicts significantly better (about 10%) than AROW. An interesting behavior of the prediction algorithms is observed with the CA metric. Initially, when there are few samples, the performance of Dwk increases drastically to 60%. The performance gradually decreases to 50%, when more samples are observed. However, in case of AROW, the learning is steadily increasing. The reason is that the Dwk algorithm predicts an output using the knowledge from the stored samples. When more samples are collected, the value of k becomes higher and also the uncertainty between the input and output among the observed contexts increases. In case of AROW, the weights are modified based on the observed context and the actual output, when new samples are encountered. This makes the learning of AROW, slow and steady.

Fig. 8.5 shows the learning curves of the prediction algorithms for W_3. As in the case of W_3, Dwk performs better than AROW. However, the prediction algorithms achieve high RS_{CC} performance more quickly than in the case of W_5. After the high performance is achieved, the learning rate is very slow. Moreover,

the CA performance of the Dwk algorithm starts to decrease after 1000 samples. This shows that when the input-output relationship in the dataset becomes more uncertain, i.e. when the randomness increases, it becomes more difficult for Dwk to predict accurate outputs. However, since the predicted outputs are still relevant for the contexts, the high RS_{CC} performance is achieved, even with few samples. When more samples are observed, the uncertainty between the input-output still remain high and hence the RS_{CC} performance remains high, but with reduced learning rate. From this study, we find that when there is less uncertainty in the dataset, the initial RS_{CC} performance of prediction algorithms will be low, after that the learning proceeds in a faster rate. When there is more uncertainty in the dataset, the initial RS_{CC} performance of prediction algorithms will be high, after that the learning proceeds in a slower rate. For the non-deterministic multiple output problems, using CA as a metric shows that the learning converges faster with lower performance, which is not really the case.

In general, for intelligent lighting, the performance of online learning algorithms is lower than that of kNN algorithms for all the considered metrics. This may be because the online algorithms do not store the observed samples, significant information is lost. In online learning, the SCW complements the AROW by using an adaptive margin, resulting in smoother behavior and higher performance. For the kNN algorithms, even though the learning curve is not smooth initially, they become smoother with more observed samples. This becomes more evident from the learning curves using the RS_{Gen} metric, meaning that sufficient information (underlying distribution for the contexts) is obtained with more samples. The results also show that the Fwk algorithm gives better prediction performance among the considered prediction algorithms.

8.5 Conclusion

The instance-based learning algorithms are not effective in terms of space and time complexity. This has motivated us to investigate online learning algorithms for intelligent lighting. Like instance-based algorithms, online learning algorithms are also non-parametric models. However, they do not store any observed samples thereby making them more efficient in terms of computation time and memory requirements.

In this chapter, we applied four online learning algorithms for intelligent lighting, implemented in LIBOL, an online learning library. The experiments are performed, first on the Breakout dataset using CA, RS_{CC} and RS_{Gen} as metrics. The performance of the online learning algorithms is analyzed using learning

curves and is compared with that of several variants of kNN algorithms. From the obtained results, we find that the kNN algorithms provide better prediction performance than that of online learning algorithms (about 10%) and Fwk provides the maximum performance. Among online learning algorithms, the SCW and umP algorithms provide smooth performance and the SCW algorithm provides the best performance. The learning curves also show that using the RS metric, the performance of the prediction algorithms still improves after 200 samples. To better understand the behavior of prediction algorithms for non-deterministic multiple-output problems with large sample size, we performed experiments on the Wine dataset. The results show that the RS_{CC} performance of prediction algorithms still improves with more observed samples and that the kNN algorithms give best performance. However, the learning rate depends on the degree of uncertainty in the dataset.

CHAPTER 9

Conclusions and Future Work

Research has shown that *good quality* lighting in work places can improve the health and well-being of humans. However, good quality lighting is subjective i.e. a given lighting condition that is acceptable for a user may not be acceptable to another. Also, the desired lighting conditions may depend on several factors such as user identity, type of activity and time of the day that form the context. *Intelligent lighting* aims at providing desired lighting conditions to its users for the observed context. However, realizing intelligent lighting system involves several issues such as finding a suitable approach for implementation, identifying relationships between context and desired lighting conditions, and determining appropriate performance metrics to evaluate intelligent algorithms.

In this chapter, we summarize the main contributions of the thesis and then open the scope for possible future work.

9.1 Main Contributions

We studied machine learning approaches to realize intelligent lighting applications. Generally, machine learning algorithms are used to identify patterns in the data, determine relationship among variables or an input-output relationship in the data and subsequently use this knowledge to predict future outputs. In Chapter 2, we discussed some of the adaptive lighting implementations described in the literature. These implementations do not use machine learning whereas most of them aim at satisfying a certain objective function to distribute the task of providing light among different light sources. In our case of intelligent lighting, we aimed at identifying a suitable machine learning approach

to provide maximum performance based on a selected performance metric. We then discussed several machine learning approaches to implement intelligent lighting. Based on the objective of intelligent lighting that we aim to achieve, we selected supervised, instance-based and online learning approaches to implement intelligent lighting. This is because, we expect the machine learning algorithm to learn from desired lighting condition provided by the user, for a given context. However, in reinforcement learning, learning occurs through exploring different possible lighting conditions and continuous interaction until the user is satisfied, which is not desirable in most cases. Therefore, we concluded that reinforcement learning is not a suitable approach.

In Chapter 3, we focused on a pilot implementation for intelligent lighting. We selected a room, dubbed the *breakout* area, which the office employees may use to have informal meetings or for personal retreat, to implement intelligent lighting. We explained the physical layout of the breakout area and discussed the network and physical architecture of the intelligent lighting system. Based on the physical layout, we listed the input features that may influence a user in selecting a particular lighting condition. At any time t, the values of these input features together form a context. Subsequently, we described the features of the *Breakout* dataset collected from the breakout area that includes the user-sample distribution and the output distribution. We then detailed the data collection. In this chapter, we found that the physical layout where the intelligent lighting system should be installed plays an important role in determining the features for collecting the data.

In Chapter 4, we performed experiments using the supervised learning approach on the Breakout dataset. The objectives were to determine the input-output relationship in the dataset and to analyze the significance of the selected input features. Intelligent lighting is an application where we expect that more than one lighting condition may be acceptable for a given context, which we call *non-deterministic multiple-output* problems. For such application, performance metrics such as *Classification Accuracy* (CA) can only measure the accuracy of the predicted lighting conditions. Therefore, we introduced *Relevance Score* (RS) that we devised to evaluate non-deterministic multiple-output problems. We then applied various rule based prediction algorithms available in WEKA, on the Breakout dataset, using both CA and RS as metrics. The experimental results showed that there is a *one-to-many* relationship between the input and output. This means that for a given context there is more than one acceptable output lighting condition. Furthermore, we identified that the feature *Type of Activity* is the most important feature for prediction algorithms in selecting a desired lighting condition.

For the non-deterministic multiple output problems, i.e. in the applications that have one-to-many input-output relationships, the commonly used CA metric is not an appropriate metric to analyze the performance of prediction algorithms. This is because the CA metric aims at measuring whether a predicted output is accurate or not; whereas it makes more sense to measure its relevance for a given context. In this direction, in Chapter 5, we further detailed the performance metric *Relevance Score* introduced in Chapter 4. For a given context, when there is a mismatch between the predicted and the actual output, the RS metric measures how relevant the predicted output is. We further explained two variants of RS: *Relevance Score - Case-by-Case* (RS_{CC}) and *Relevance Score âĂŞ General* (RS_{Gen}. The RS for a predicted output and a given context is computed based on the posterior probabilities computed from all samples in a dataset. We demonstrated the significance of the RS metric through experiments on the Breakout dataset and three other datasets from the UCI repository. Interestingly, we found that RS performs similarly to CA, in the case of deterministic datasets. For the non-deterministic datasets, RS made more sense than CA by capturing the uncertainty between the input and output effectively. From the obtained results, we concluded that RS is an appropriate performance metric for the non-deterministic multiple output problems such as intelligent lighting where factors such as varying human perceptions lead to one-to-many input-output relationships.

The missing-data problem arises in intelligent lighting, when the users do not use a dedicated interface (e.g. smart phone) to communicate with the intelligent lighting system. Missing data occurs for the features *User Identity* and *Type of Activity* that are collected through a smart phone. In Chapter 6, we discussed some of the missing-data treatment methods available in the literature. It is not straightforward to select a treatment method and a prediction algorithm for handling missing data in intelligent lighting without experimentation. This is because the prediction performance depends not only on the treatment method, but also on the prediction algorithm, the pattern of missing data and the relationship among variables in the considered dataset. Based on the non-deterministic multiple output nature of intelligent lighting, we applied several probability based solutions to address the problem of missing data. The performance of the missing-data treatment methods are evaluated using several rule based prediction models with RS_{CC} and RS_{Gen} as metrics. For our experiments, we considered 10%, 20%, 30%, 40% and 50% of missing data. The impact of the missing-data treatment on the performance of the prediction algorithms is analyzed through *paired t*-tests. From the experimental results, in general, we find that the missing-data treatment does not noticeably improve

the performance of the prediction algorithms. This can be attributed mainly to the non-deterministic nature of the data in intelligent lighting. Other factors include the considered missing-data treatment methods, prediction algorithms and low number of samples in the dataset. Finally, we concluded that the *DecisionTable* is the most suitable supervised prediction algorithm for intelligent lighting without missing-data treatment.

The main disadvantage of applying supervised learning to realize intelligent lighting is that it is a parametric model. Supervised learning assumes that the distribution from where the samples are drawn is known. However, this rarely holds true in intelligent lighting because of factors such as varying human perceptions and changing pattern in the data over time. In Chapter 7, we studied the most popular instance-based learning algorithm, *k-Nearest Neighbor* (kNN), in the context of intelligent lighting. The kNN algorithm is a non-parametric algorithm that does not assume any underlying variable distribution. We performed experiments on the Breakout dataset using variants of the kNN algorithm such as Feature weighted and Distance weighted kNN with $k = 1$ and $k = 15$. We analyzed and compared the results to that of the supervised learning algorithm, specifically the *DecisionTable*. The experimental results showed that the kNN algorithms provide similar to that of the DecisionTable supervised learning algorithm. We concluded that the kNN algorithms are preferred in intelligent lighting as they can adapt to the changing pattern in the data over time, when used in online mode.

The kNN algorithms, when used as online learner may adapt over time. However, they are not effective computationally and in terms of storage requirements. Moreover, kNN algorithms are sensitive to noise in features and irrelevant features. In this direction, in Chapter 8, we explore online learning algorithms for intelligent lighting. Online learning algorithms also belong to the non-parametric models that assume the future inputs are unknown. Also, online learning algorithms store the information acquired from the observed samples in the form of weights. This reduces the computational complexity of these algorithms. We investigated four online learning algorithms that are implemented in LIBOL, a library for online learning algorithms. We made modifications to the process of computing RS to suit online algorithms. We applied the online learning algorithms on the Breakout dataset and analyzed the performance of the online algorithms using their learning curves. As expected, the results showed that the kNN algorithms provide better prediction performance than that of online learning algorithms for all the three metrics (CA, RS_{CC} and RS_{Gen}) considered. This is because online learning algorithms do not store samples for future reference, which causes them to lose some information about the

history. An interesting observation is that the learning curves using the CA metric showed that the performance of the prediction algorithms converges faster. This means that prediction algorithms reach their maximum performance faster and do not learn when more samples are added. However, the learning curves using RS showed that performance of prediction algorithms do not converge faster and still continue learning with more number of samples added.

From the studies performed using different machine learning approaches for realizing intelligent lighting, we find that supervised learning algorithms are not suitable for intelligent lighting. In intelligent lighting, the underlying distribution (input-output relationship) is not consistent over time due to factors such as varying human perceptions. Supervised learning algorithms do not continuously learn from new incoming data, as they are trained on a batch of samples that do not represent the input space entirely. This motivated us to consider non-parametric algorithms such as instance-based algorithms and online learning algorithms for intelligent lighting. From our experiments, we find that instance-based algorithms provide the maximum prediction performance, but demand more storage requirements $(O(m))$ and the time to predicted an output increases with increasing number of samples (m). On the other hand, online learning algorithms provide significantly lower performance, more than 10%, with CA and RS_{CC} metrics. However, they do not store samples and predict outputs instantaneously. We performed our experiments on the Breakout dataset having 236 samples. The number of samples is not sufficient enough to conclude whether instance-based or online learning is the most suitable approach to implement intelligent lighting. Based on the maximum performance achieved, we select instance-based algorithms for our intelligent lighting implementation.

9.2 Future Work

We have seen that the instance-based approach for intelligent lighting makes use of the history of observed samples that helps in providing better predictions. However, this is also a disadvantage, that as the number of observed samples grows, the demand for storage increases. Also, since instance-based algorithms predict only when a new context is encountered, it takes $O(m)$ time to predict for m observed samples. To address this drawbacks, there are several versions of kNN [19] such as the Condensed kNN [58] [7] and Orthogonal Search Tree NN [89] proposed. Now that we know instance-based algorithms provide the maximum performance for intelligent lighting, as a future work,

these algorithms can be studied to mitigate the drawbacks associated with the kNN algorithm. Furthermore, a window-based approach may also be investigated for intelligent lighting. In this approach, a window that consists of a fixed number of samples is assumed. When a new sample is observed, the oldest sample in the window is removed and the new sample is added. Using this approach, we expect that the kNN algorithms to adapt and run faster, and also to address the storage problem.

In this thesis, we considered eight preset lighting conditions from which the users can select a suitable lighting condition for a given context. The use of Light Emitting Diode (LED) technologies enables the lighting system to provide millions of colors. Therefore, the studied intelligent lighting system can be extended by allowing the users to modify the lighting conditions by varying light parameters such as color temperature and intensities. In this setting, the users are not limited to a fixed set of lighting condition and have freedom to manipulate the lighting conditions that best suits them. However, the learning problem becomes more complicated by introducing more dimensions to the output space. For example, suppose that for a given context, the output lighting condition is defined by hue-saturation-brightness (HSB). This means that a machine learning algorithm has to predict values for HSB separately and also the HSB values need to be correlated to provide a meaningful lighting condition. Generally, most problems in machine learning are designed to predict in one dimensional output space. Recently, Balasubramanian et al. [14] have developed a method for predicting in multi-dimensional output space. Also, clustering techniques such as k-mean [74] can be investigated to address this problem.

Research has showed that light not only supports day-to-day activities but also helps in reducing psychological disorders such as mental stress and unhappiness [1]. Recent works showed that the level of stress experienced by humans can be measured using wearable devices, smart phones [97] or by using modern sensor technologies to measure galvanic skin response and heart-rate [13] that highly correlates with stress. This means that stress experienced by humans can also be included as an input feature in intelligent lighting. Now that we can monitor the stress levels of users, the objective of intelligent lighting can now be redefined as to predict desired lighting conditions that help in reducing the stress levels of users. Here, intelligent lighting can be formulated as reinforcement learning problem where the predicting suitable lighting conditions are the *actions* and the monitored stress levels are the *feedback*. Also, the performance of prediction algorithms can be implicitly evaluated by continuously monitoring the stress levels as feedbacks. By this we mean that if the stress level of a user is high, we expect the predicted lighting conditions to reduce the stress level. In

this example, we show a slightly different objective of intelligent lighting and how reinforcement learning can be investigated to realize it.

The nature of intelligent lighting i.e. the non-deterministic input-output relationship offers new challenges during its development. For example, we have seen that the missing-data treatment for intelligent lighting was not effective. Here, we used the missing-data treatment on the Breakout dataset in the supervised learning setting. In the training set, some of the treatment approaches made use of the knowledge available from the output class labels. We have also seen that non-parametric models such as instance based and online learning algorithms are more suitable for intelligent lighting. In online learning setting, we cannot extract information about the context from output class labels, as they need to be predicted at runtime. Also, useful knowledge cannot be gained from the history of observed samples. This is because, when a missing feature value for a context is treated, the true feature value for that context is never revealed. This makes the missing-data problem more complex in online setting. To tackle missing-data problem in online setting, we propose the similar probability based approaches as in the case of supervised learning because of the non-deterministic nature of intelligent lighting.

There are also opportunities in the domain of *performance metrics*. The objective of our intelligent lighting application was to provide desired lighting conditions to its users based on the observed context. The performance of the intelligent lighting application was measured by evaluating the lighting conditions predicted by the prediction algorithms by using the RS metric. Suppose that the objective of the intelligent lighting is to select a prediction algorithm that not only provides desired lighting condition but also makes the users physically active, by making them frequently interact with the system. In this case, using RS as a metric can only evaluate the relevance of predicted lighting conditions, but it cannot measure a user's interaction with the system or the users quality of experience, thereby motivating the need for a new metric. A possible approach could be to combine RS with some sort of user interaction measurement. A user's interaction with the intelligent lighting system can be measured by counting the number of different lighting conditions explored by them during a session. From this example, we highlight that the objective of intelligent lighting defined may drive the need for devising new performance metrics.

9.3 Final Remarks

To summarize, intelligent lighting not only helps in improving the well-being of humans but also poses several challenging problems starting from defining objectives to devising performance metrics. In our study, we have identified that intelligent lighting has a one-to-many input-output relationship, thereby classifying it as a *non-deterministic multiple output* problem. The major contribution of this thesis is that we devised a performance metric dubbed RS. The RS metric can be used to evaluate the performance of machine learning algorithms for the class of applications that can be categorized as non-deterministic multiple output problems. The experiments on the $Wine$ dataset showed that the performance of prediction algorithms using RS is similar to that of CA, in case of deterministic input-output relationships. For the non-deterministic multiple output problems such as intelligent lighting, an important observation is made from the experiments performed using online learning algorithms. The results showed that using the CA metric, the prediction algorithms do not perform well and converge faster. This means that prediction algorithms reach their maximum performance with fewer samples but do not improve with the addition of more samples. However, this is contradicted using the RS metric, which showed that prediction algorithms can still learn and improve their performance with the increase in number of samples. Finally, to conclude, the objective of the intelligent lighting application may lead to identify several challenging problems. Addressing these problems and realizing intelligent lighting efficiently not only helps in improving the well-being of humans, but also enable to determine solutions that may be applicable to other real world problems.

Appendix A

Breakout Dataset

UID ToA ToD IoU AoA ExLI OP
2 RELAX_GROUP 08.50 3 MA High 7
2 ACTIVE_ALONE 08.50 0 MA High 8
2 RELAX_ALONE 08.50 0 RA VeryLow 11
2 ACTIVE_GROUP 09.00 2 RA VeryLow 6
13 ACTIVE_ALONE 09.50 2 MA High 6
13 RELAX_ALONE 09.50 2 RA VeryLow 9
13 RELAX_GROUP 09.50 0 RA High 6
13 ACTIVE_GROUP 09.50 1 MA VeryLow 9
1 RELAX_ALONE 10.00 1 MA High 11
1 RELAX_GROUP 10.00 1 MA VeryLow 7
1 ACTIVE_ALONE 10.00 0 RA High 9
1 ACTIVE_GROUP 10.00 2 RA VeryLow 6
5 ACTIVE_ALONE 10.50 2 RA VeryLow 9
5 ACTIVE_GROUP 10.50 3 MA High 8
5 RELAX_ALONE 10.50 3 MA VeryLow 11
5 RELAX_GROUP 10.50 2 RA High 7
12 ACTIVE_ALONE 12.00 2 MA VeryLow 6
12 ACTIVE_GROUP 12.00 2 RA High 11
12 RELAX_ALONE 12.00 4 RA VeryLow 12
12 RELAX_GROUP 12.00 4 RA High 11
15 ACTIVE_ALONE 13.00 0 RA High 8
15 RELAX_ALONE 13.00 4 RA High 6
15 RELAX_GROUP 13.00 3 MA VeryLow 7
15 ACTIVE_GROUP 13.00 2 MA VeryLow 8
10 RELAX_ALONE 14.00 0 RA VeryLow 6
10 RELAX_GROUP 14.00 2 MA High 13
10 ACTIVE_ALONE 14.00 2 MA High 9
10 ACTIVE_GROUP 14.00 4 RA VeryLow 8
19 RELAX_GROUP 14.50 2 MA VeryLow 8
19 ACTIVE_GROUP 14.50 1 MA High 10
19 RELAX_ALONE 14.50 3 RA VeryLow 13
19 ACTIVE_ALONE 14.50 0 RA High 10
9 RELAX_GROUP 15.00 0 RA High 6
9 ACTIVE_ALONE 15.00 8 RA VeryLow 9
9 ACTIVE_GROUP 15.00 2 RA High 9
9 RELAX_ALONE 15.00 1 MA VeryLow 12
17 ACTIVE_GROUP 09.50 2 MA High 8
17 ACTIVE_ALONE 09.50 4 MA High 11
17 RELAX_ALONE 09.50 0 RA High 10
17 RELAX_GROUP 09.50 1 RA VeryLow 6
8 ACTIVE_ALONE 10.00 2 MA High 9
8 RELAX_GROUP 10.00 2 MA VeryLow 10
8 RELAX_ALONE 10.00 2 RA High 6
8 ACTIVE_GROUP 10.00 2 RA VeryLow 7
6 RELAX_ALONE 10.50 1 RA VeryLow 6
6 ACTIVE_ALONE 10.50 2 RA High 6
6 RELAX_GROUP 10.50 2 MA VeryLow 9

6 ACTIVE_GROUP 10.50 1 MA High 7
18 ACTIVE_GROUP 14.50 3 MA VeryLow 9
18 RELAX_GROUP 14.50 1 MA High 7
18 ACTIVE_ALONE 14.50 7 RA High 7
18 RELAX_ALONE 14.50 6 RA VeryLow 6
20 RELAX_ALONE 09.00 0 MA Low 9
20 RELAX_GROUP 09.00 0 MA VeryLow 7
20 ACTIVE_ALONE 09.00 7 RA Low 11
20 ACTIVE_GROUP 09.00 3 RA VeryLow 7
20 RELAX_GROUP 09.00 2 MA VeryLow 6
20 ACTIVE_ALONE 09.50 0 MA Low 8
20 RELAX_ALONE 09.50 7 RA VeryLow 12
20 ACTIVE_GROUP 09.50 1 RA VeryLow 7
20 RELAX_GROUP 09.50 2 MA VeryLow 6
20 ACTIVE_GROUP 09.50 1 MA VeryHigh 8
20 RELAX_ALONE 09.50 0 RA Low 6
20 ACTIVE_ALONE 10.00 2 RA High 6
13 RELAX_ALONE 10.00 1 MA VeryHigh 12
13 RELAX_GROUP 10.00 0 MA High 6
13 ACTIVE_GROUP 10.00 4 RA Low 9
13 ACTIVE_ALONE 10.00 0 RA Low 7
13 ACTIVE_GROUP 10.00 3 MA VeryHigh 12
13 RELAX_ALONE 10.00 2 MA VeryLow 8
13 ACTIVE_ALONE 10.00 5 RA VeryLow 7
13 RELAX_GROUP 10.50 5 RA High 8
13 RELAX_ALONE 10.50 4 RA Low 12
13 ACTIVE_ALONE 10.50 6 RA High 10
13 RELAX_GROUP 10.50 1 MA Low 8
13 ACTIVE_GROUP 10.00 3 MA VeryHigh 12
1 ACTIVE_GROUP 10.50 1 MA VeryLow 7
1 ACTIVE_ALONE 10.50 1 MA VeryHigh 6
1 RELAX_ALONE 10.50 2 RA VeryHigh 12
1 RELAX_GROUP 10.50 3 RA Low 10
1 RELAX_ALONE 10.50 3 RA VeryHigh 12
1 ACTIVE_GROUP 11.00 3 RA Low 6
1 RELAX_GROUP 11.00 4 MA Low 7
1 ACTIVE_ALONE 10.50 1 MA VeryHigh 6
19 ACTIVE_ALONE 11.00 0 MA VeryLow 8
19 RELAX_ALONE 11.00 0 MA VeryLow 12
19 RELAX_GROUP 11.00 4 RA VeryHigh 13
19 ACTIVE_GROUP 11.00 2 RA VeryHigh 9
19 RELAX_ALONE 11.00 0 RA VeryLow 10
19 ACTIVE_GROUP 11.00 0 RA VeryLow 8
19 RELAX_GROUP 11.00 1 MA VeryHigh 9
19 ACTIVE_ALONE 11.50 2 MA High 11
19 ACTIVE_GROUP 11.50 2 MA VeryLow 8
19 RELAX_GROUP 11.50 1 MA Low 11
19 ACTIVE_ALONE 11.50 3 RA High 13
19 RELAX_ALONE 11.50 2 RA VeryLow 10
21 RELAX_GROUP 12.50 0 MA VeryLow 7
21 ACTIVE_ALONE 12.50 0 MA VeryLow 8
21 RELAX_ALONE 13.00 4 RA Low 11
21 ACTIVE_GROUP 13.00 0 RA Low 9
21 RELAX_GROUP 13.00 2 RA VeryLow 7
21 ACTIVE_GROUP 13.00 0 RA VeryLow 9
21 ACTIVE_ALONE 13.00 2 MA Low 9
21 RELAX_ALONE 13.00 0 MA Low 10
21 RELAX_GROUP 13.00 1 MA VeryLow 7
21 ACTIVE_GROUP 13.00 1 MA Low 9
21 ACTIVE_ALONE 13.00 5 RA Low 9
21 RELAX_ALONE 13.00 3 RA Low 11
2 RELAX_GROUP 13.00 3 RA VeryLow 6
2 ACTIVE_GROUP 13.00 4 RA Low 8
2 RELAX_ALONE 13.00 1 RA Low 10
2 RELAX_GROUP 13.00 3 RA VeryLow 6
2 ACTIVE_ALONE 13.00 4 MA VeryLow 9
2 RELAX_ALONE 13.50 2 MA VeryLow 11
2 ACTIVE_GROUP 13.50 1 MA Low 8
2 ACTIVE_ALONE 13.50 1 MA Low 9
16 ACTIVE_GROUP 13.50 2 MA Low 7
16 RELAX_GROUP 13.50 2 RA Low 13
16 RELAX_ALONE 13.50 1 RA Low 6
16 ACTIVE_ALONE 13.50 0 RA Low 6
16 ACTIVE_GROUP 13.50 0 RA Low 7
16 ACTIVE_GROUP 13.50 2 MA Low 9
16 ACTIVE_ALONE 13.50 2 MA Low 6
16 ACTIVE_ALONE 13.50 1 MA VeryLow 9
16 RELAX_ALONE 13.50 0 MA VeryLow 8

16 RELAX_ALONE 13.50 0 MA VeryLow 9
16 RELAX_GROUP 13.50 3 RA VeryLow 9
16 RELAX_GROUP 13.50 2 RA VeryLow 10
9 RELAX_GROUP 15.50 3 MA Low 6
9 RELAX_GROUP 15.50 2 MA Low 6
9 RELAX_GROUP 16.00 1 MA Low 7
9 RELAX_GROUP 16.00 0 MA Low 6
10 ACTIVE_GROUP 15.50 1 MA VeryLow 8
10 ACTIVE_ALONE 15.50 2 MA VeryLow 9
1 RELAX_ALONE 16.50 0 RA VeryLow 6
1 ACTIVE_GROUP 17.00 1 MA VeryLow 9
18 ACTIVE_ALONE 10.00 2 RA VeryLow 8
18 RELAX_ALONE 10.50 8 RA VeryLow 6
18 RELAX_GROUP 10.50 10 MA Low 13
18 ACTIVE_ALONE 13.00 0 RA VeryLow 9
18 RELAX_ALONE 15.00 0 RA VeryLow 10
15 RELAX_GROUP 13.50 1 MA VeryLow 7
8 ACTIVE_ALONE 14.00 0 RA VeryLow 9
10 RELAX_GROUP 14.50 1 MA Low 7
10 ACTIVE_GROUP 15.00 0 RA VeryLow 7
19 ACTIVE_ALONE 13.50 2 MA VeryLow 9
19 ACTIVE_ALONE 13.50 6 MA VeryLow 9
1 ACTIVE_ALONE 14.00 0 MA VeryLow 7
18 ACTIVE_ALONE 14.50 0 RA VeryLow 9
18 ACTIVE_ALONE 15.00 0 RA VeryLow 8
18 ACTIVE_ALONE 10.50 0 RA VeryLow 9
19 ACTIVE_GROUP 11.00 10 MA VeryLow 9
19 ACTIVE_GROUP 11.00 5 MA VeryLow 9
19 ACTIVE_GROUP 11.00 0 MA VeryLow 8
12 RELAX_ALONE 15.00 0 RA VeryLow 6
18 ACTIVE_GROUP 09.50 5 MA VeryLow 9
18 RELAX_ALONE 09.50 0 RA VeryLow 7
18 ACTIVE_GROUP 10.50 1 MA VeryLow 8
18 ACTIVE_GROUP 10.50 0 MA VeryLow 8
18 ACTIVE_ALONE 13.50 0 RA VeryLow 9
1 ACTIVE_ALONE 14.00 0 MA VeryLow 9
5 RELAX_ALONE 15.50 0 RA VeryLow 6
18 RELAX_ALONE 13.00 0 RA VeryLow 6
12 RELAX_GROUP 16.00 0 RA VeryLow 7
18 ACTIVE_ALONE 08.50 0 RA VeryLow 9
18 ACTIVE_GROUP 09.00 0 MA Low 8
8 ACTIVE_GROUP 10.00 3 MA VeryLow 8
18 ACTIVE_GROUP 13.50 0 MA VeryLow 8
10 RELAX_ALONE 14.50 2 MA VeryLow 6
10 ACTIVE_GROUP 15.00 2 MA VeryLow 6
10 RELAX_GROUP 16.00 0 MA VeryLow 7
10 ACTIVE_ALONE 16.50 0 RA VeryLow 9
10 ACTIVE_ALONE 16.50 0 RA VeryLow 9
19 ACTIVE_ALONE 10.00 0 MA VeryLow 9
12 ACTIVE_ALONE 10.50 1 MA VeryLow 9
12 ACTIVE_GROUP 10.50 0 MA VeryLow 8
2 ACTIVE_ALONE 09.00 1 MA VeryLow 7
12 ACTIVE_GROUP 10.50 3 MA VeryLow 12
18 RELAX_ALONE 14.00 0 RA VeryLow 6
18 ACTIVE_ALONE 16.50 0 RA Low 12
1 ACTIVE_ALONE 11.00 1 RA Low 9
12 RELAX_ALONE 12.00 0 RA VeryLow 6
12 RELAX_ALONE 12.00 0 RA VeryLow 6
2 RELAX_GROUP 09.50 1 MA Low 6
2 ACTIVE_GROUP 09.50 0 MA Low 8
2 ACTIVE_GROUP 11.00 3 MA Low 7
18 ACTIVE_GROUP 11.50 0 MA Low 8
2 RELAX_GROUP 11.50 1 MA Low 7
18 ACTIVE_GROUP 13.50 10 MA Low 8
18 ACTIVE_GROUP 13.50 0 MA Low 8
18 ACTIVE_GROUP 14.00 0 MA Low 8
18 ACTIVE_GROUP 14.50 10 MA High 8
5 ACTIVE_ALONE 10.50 0 RA VeryLow 9
12 RELAX_ALONE 14.50 1 MA Low 6
12 ACTIVE_ALONE 14.50 1 MA Low 6
12 RELAX_ALONE 15.00 1 MA Low 12
18 ACTIVE_GROUP 09.50 1 MA Low 7
18 ACTIVE_GROUP 10.50 1 MA Low 8
2 RELAX_ALONE 13.50 10 MA Low 6
15 ACTIVE_ALONE 14.50 0 RA Low 9
15 RELAX_GROUP 14.50 0 RA Low 7
15 RELAX_ALONE 15.50 0 RA Low 7
15 RELAX_ALONE 15.50 0 RA Low 6

15 RELAX_ALONE 16.00 0 RA Low 6
5 ACTIVE_ALONE 16.00 0 RA Low 9
5 ACTIVE_ALONE 16.00 0 RA Low 9
2 RELAX_ALONE 12.50 0 RA Low 6
2 RELAX_GROUP 10.50 0 RA Low 7
2 ACTIVE_GROUP 11.50 1 MA Low 6
2 ACTIVE_GROUP 11.50 7 MA Low 6
2 ACTIVE_GROUP 11.50 9 MA Low 6
2 RELAX_ALONE 13.00 8 RA Low 12
2 RELAX_ALONE 13.50 0 RA Low 6
12 ACTIVE_GROUP 13.50 2 MA Low 8
12 ACTIVE_ALONE 11.50 1 MA Low 8
12 RELAX_GROUP 13.00 2 MA Low 8
18 ACTIVE_ALONE 11.50 0 MA Low 7
2 ACTIVE_GROUP 15.00 0 MA Low 9
9 RELAX_ALONE 14.50 0 MA Low 8
18 ACTIVE_GROUP 09.50 1 MA Low 8
18 ACTIVE_GROUP 09.50 0 MA Low 6
18 ACTIVE_GROUP 09.50 0 MA Low 6
18 RELAX_ALONE 11.50 5 RA Low 8
18 RELAX_ALONE 12.00 7 RA Low 6
18 ACTIVE_GROUP 09.00 0 MA VeryLow 6
18 ACTIVE_GROUP 09.00 8 MA VeryLow 8
18 ACTIVE_ALONE 10.50 4 MA VeryLow 12
18 ACTIVE_ALONE 10.50 0 MA VeryLow 9
18 ACTIVE_ALONE 11.00 10 MA VeryLow 9
18 ACTIVE_ALONE 11.00 3 MA VeryLow 9
18 ACTIVE_ALONE 11.00 0 MA VeryLow 9
18 ACTIVE_ALONE 11.50 10 MA VeryLow 6
18 ACTIVE_ALONE 11.50 2 MA VeryLow 9
18 ACTIVE_ALONE 11.50 2 MA VeryLow 9
18 ACTIVE_GROUP 11.50 10 MA VeryLow 9
18 ACTIVE_GROUP 09.50 0 MA Low 9

Bibliography

[1] M. aan het Rot, D. S. Moskowitz, and S. N. Young. Exposure to bright light is associated with positive social interaction and good mood over short time periods: A naturalistic study in mildly seasonal people. *Journal of Psychiatric Research*, 42:311–319, 2008.

[2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*, 2007.

[3] E. Acuna and C. Rodriguez. The treatment of missing values and its effect in the classifier accuracy. *Classification, Clustering and Data Mining Applications*, -:639–648, 2004.

[4] D. W. Aha, D. Kibler, and M. K. Albert. Instance-Based Learning Algorithms. *Machine Learning*, 6:37–66, 1991.

[5] S. Albers. *Online Algorithms*, pages 143–164. Springer-Verilag, 2006.

[6] M. Aly. Survey of Multiclass Classification Methods. Technical report, California Institute of Technology, 2005.

[7] F. Angiulli. Fast Condensed Nearest Neighbor Rule. In *22nd International Conference on Machine Learning*, pages 25–32, 2005.

[8] K. Aquino, D. Freeman, A. Reed II, and V. K. G Lim. Testing a Social-Cognitive Model of Moral Behavior: The Interactive Influence of Situations and Moral Identity Centrality. *Journal of Personality and Social Psychology*, 97(1):123–141, 2009.

[9] Arduino. Arduino Uno. http://arduino.cc/, Jun 2012. [Online: accessed Jun-2012].

[10] Atmel Corporation. *8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash: ATmega328p*, Oct 2009.

[11] K. Bache and M. Lichman. UCI Machine Learning Repository, 2013.

[12] Y. Bai and Y. Ku. Automatic Light Detection and Control Using a Microprocessor and Light Sensors. *IEEE Transactions on Consumer Electronics*, 54(3):1173–1176, 2008.

[13] J. Bakker, L. Holenderski, R. Kocielnik, M. Pechenizkiy, and N. Sidorova. Stess@work: From Measuring Stress to its Understanding, Prediction and Handling with Personalized Coaching. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 673 – 678, 2012.

[14] K. Balasubramanian and G. Lebanon. The Landmark Selection Method for Multiple Output Prediction. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 983–990, 2012.

[15] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 568–573, 2005.

[16] S. H. A. Begemann, G. J. van den Beld, and A. D. Tenner. Daylight, artificial light and people in an office environment, overview of visual and biological responses. *International Journal of Industrial Ergonomics*, 20:231–239, 1997.

[17] D. M. Berson. Strange vision: ganglion cells as circadian photoreceptors. *TRENDS in Neurosciences*, 26(6):314–320, 2003.

[18] S. Bhardwaj, T. Ozcelebi, and J. J. Lukkien. Smart Lighting using LED Luminaries. In *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 654–659, 2010.

[19] N. Bhatia and Vandana. Survey of Nearest Neighbor Techniques. *International Journal of Computer Science and Information Security*, 8(2):302 –305, 2010.

[20] D. E. Blask. Melatonin, sleep disturbance and cancer risk. *Sleep Medicine Reviews*, 13:257–264, 2009.

[21] C. E. Boehnert and R. A. Alberts. Seasonal Affective Disorder in Women. *WOMEN'S HEALTH in Primary Care*, 6(1):32–36, 2003.

[22] M. Canazei. Laboratory experiment regarding impact on productivity through dynamic lighting. Technical report, Zumtobel Research, 2013.

[23] P. Carner. Bringing Intelligence to LED Lighting Applications. Technical report, Texas Instruments, 2012.

[24] W. H. Cohen. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[25] Pentaho Community. ConjunctiveRule. http://wiki.pentaho.com/display/DATAMINING/ConjunctiveRule, Nov 2014. [Online; accessed Nov-2014].

[26] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20 (3):273–297, 1995.

[27] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[28] K. Crammer, M. Dredze, and A. Kulesza. Multi-class Confidence Weighted Algorithms. In *Empirical Methods in Natural Language Processing*, pages 496 – 504, 2009.

[29] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. *Machine Learning*, 91(2):155–187, 2013.

[30] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

[31] P. Cunningham, M. Cord, and S. J. Delany. *Supervised Learning*, chapter 2, pages 21–49. Springer Berlin Heidelberg, 2008.

[32] A. C. Davison. Statistical models. Technical report, Swiss Federal Institute of Technology, 2003.

[33] Y.A.W. de Kort and K.C.H.J. Smolders. Effects of dynamic lighting on office workers: First results of a field study with monthly alternating settings. *Lighting Research and Technology*, 42(3):345–360, 2010.

[34] A. Dean and D. Voss. *Design and Analysis of Experiments*. Springer, 1999.

[35] J. Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[36] W. J. Diamond. *Practical Experiment Designs For Engineers and Scientists*. Wiley, 2001.

[37] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., 2001.

[38] S. A. Dudani. The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4):325–327, 1976.

[39] Jeanne F. Duffy and Charles A. Czeisler. Effect of Light on Human Circadian Physiology. *Sleep Medicine Clinics*, 4:165–177, 2009.

[40] M. Dumont and C. Beaulieu. Light exposure in the natural environment: Relevance to mood and sleep disorders. *Sleep Medicine*, 8:557–565, 2007.

[41] C. I. Eastman, M. A. Young, L. F. Fogg, L. Liu, and P. M. Meaden. Bright Light Treatment of Winter Depression. *Archives of General Psychiatry*, 55:883–889, 1998.

[42] L. Edwards and P. Torcellini. A Literature Review of the Effects of Natural Light on Building Occupants. Technical report, National Renewable Energy Laboratory, Golden, Colorado, July 2002.

[43] M. S. El-Nasr. Intelligent Lighting For Game Environments. *Journal of Game Development*, 1(2):17–50, 2005.

[44] A. J. Elliot and M. A. Maier. Color Psychology: Effects of Perceiving Color on Psychological Functioning in Humans. *Annual Review of Psychology*, 65:95–120, 2014.

[45] A. Farhangfar, L. Kurgan, and J. Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705, Dec 2008.

[46] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[47] B. W. Flury, M. J. Schmid, and A. Narayanan. Error Rates in Quadratic Discrimination with Constraints on the Covariance Matrices. *Journal of Classification*, 11:101–120, 1994.

[48] E. Frank and I. H. Witten. Generating Accurate Rule Sets Without Global Optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 144–151, 1998.

[49] J. Fürnkranz, D. Gamberger, and N. Lavrac. *Foundations of Rule Learning*. Springer, 2012.

[50] M. F. A. Gadi, X. Wang, and A. P. do Lago. Credit Card Fraud Detection with Artificial Immune System. *Lecture Notes in Computer Science*, 5132:119–131, 2008.

[51] B. R. Gaines and P. Compton. Induction of ripple-down rules applied to modeling large databases. *Journal of Intelligent Information Systems*, 5(3):211–228, November 1995.

[52] U. Gneezy, S. Meier, and P. Rey-Biel. When and Why Incentives (Don't) Work to Modify Behavior. *Journal of Economic Perspectives*, 25(4):191–210, 2011.

[53] S. Godbole and S. Sarawagi. Discriminative Methods for Multi-labeled Classification. In *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *Lecture Notes in Computer Science*, pages 22–30. Springer Berlin Heidelberg, 2004.

[54] A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, and J. J. Lukkien. Treatment of Missing Data in Intelligent Lighting Applications. In *Proceedings of The 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, pages 1–8, Fukuoka, Japan, 2012.

[55] A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, and J. J. Lukkien. Relevance as a Metric for Evaluating Machine Learning Algorithms. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 7988 of *Lecture Notes in Computer Science*, pages 195–208. Springer Berlin Heidelberg, 2013.

[56] A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, and J. J. Lukkien. Statistical Inferernce for Intelligent Lighting: A Pilot Study. In *Eighth International Symposium on Intelligent Distributed Computing*, 2014.

[57] L. S. Gottfredson. The General Intelligence Factor. *Scientific American Presents*, 9(4):24–29, 1998.

[58] K. C. Gowda and G. Krishna. The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighbor. *IEEE Transactions on Information Theory*, IT - 25(4):488–490, 1979.

[59] E. Grandjean. *Ergonomics of the Home*. Wiley, 1973.

[60] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):–, 2009.

[61] E. Halliwell, L. Main, and C. Richardson. The Fundamental Facts. Technical report, Mental Health Foundation, 2007.

[62] M. Hatori and S. Panda. The emerging roles of melanopsin in behavioral adaptation to light. *Trends in Molecular Medicine*, 16:435–446, 2010.

[63] S. C. H. Hoi, J. Wang, and P. Zhao. LIBOL: A Library for Online Learning Algorithms. *Journal of Machine Learning Research*, 15:495–499, 2014.

[64] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. In *Machine Learning*, pages 63–91, 1993.

[65] D. C. Howell. Treatment of Missing Data. Technical report, University of Vermont, 2002.

[66] S. Hubalek, M. Brink, and C. Schierz. Office workers' daily exposure to light and its influence on sleep quality and mood. *Lighting Research and Technology*, 42:33–50, 2010.

[67] Digi International Inc. XBee ZigBee/802.15.4 RF Modules. www.digi.com/products/xbee, Jun 2012. [Online: accessed Jun-2012].

[68] R. K. Jain. *The Art of Computer System Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley and Sons, Inc., 1991.

[69] N. Japkowicz. Why question machine learning evaluation methods? In *AAAI'06 workshop on evaluation methods for machine learning*, pages 6–11, 2006.

[70] L. Jiang, H. Zhang, and Z. Cai. Dynamic K-Nearest-Neighbor Naive Bayes with Attribute Weighted. In *Fuzzy Systems and Knowledge Discovery*, volume 4223 of *Lecture Notes in Computer Science*, pages 365 – 368. Springer Berlin Heidelberg, 2006.

[71] A. Joseph. The Impact of Light on Outcomes in Healthcare Settings. Technical report, The Centre for Health Design, 2006.

[72] A. Juels and M. Wattenbergy. Stochastic Hillclimbing as a Baseline Method for Evaluating Genetic Algorithms. Technical report, University of California at Berkeley, 1994.

[73] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[74] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[75] M. Khan, Q. Ding, and W. Perrizo. k-nearest Neighbor Classification on Spatial Data Streams Using P-trees. In *Advances in Knowledge Discovery and Data Mining*, volume 2336 of *Lecture Notes in Computer Science*, pages 517 – 528. Springer Berlin Heidelberg, 2002.

[76] R. Küller, S. Ballal, T. Laike, B. Mikellides, and G. Tonello. The impact of light and colour on psychological mood: a cross-cultural study of indoor work environments. *Ergonomics*, 49(14):1496–1507, 2006.

[77] R. Kohavi. The Power of Decision Tables. In *The Eighth European Conference on Machine Learning*, volume 912 of *Lecture Notes in Computer Science*, pages 174–189. Springer Berlin Heidelberg, 1995.

[78] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273 – 324, 1997.

[79] A. Krioukov, S. Dawson-Haggerty, L. Leeand O. Rehmane, and D.Culler. A living laboratory study in personalized automated lighting controls. In *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 1–6, 2011.

[80] Y. Kumar and G. Sahoo. Analysis of Parametric & Non Parametric Classifiers for Classification Technique using WEKA. *International Journal on Information Technology and Computer Science*, 7:43–49, 2012.

[81] P. H. Kvam and B. Vidakovic. *Nonparametric Statistics with Applications to Science and Engineering*. John Wiley and Sons, Inc., 2007.

[82] K. Lakshminarayan, S. A. Harp, and T. Samad. Imputation of Missing Data in Industrial Databases. *Applied Intelligence*, 11:259–275, 1999.

[83] P. Langley and H. A. Simon. Applications of Machine Learning and Rule Induction. *Communications of the ACM*, 38(11):54–64, 1995.

[84] D. Leger, V. Bayon, M. Elbaz, P. Philip, and D. Choudat. Underexposure to light at work and its association to insomnia and sleepiness: A cross-sectional study of 13 296 workers of one transportation company. *Journal of Psychosomatic Research*, 70:29–36, 2011.

[85] S. W. Lockley. Circadian Rhythms: Influence of Light in Humans. *Encyclopedia of Neuroscience*, 2:971–988, 2009.

[86] R. Magielse, S. Rao, P. Jaramillo, P. Ross, T. Ozcelebi, and O. Amft. An Interdisciplinary Approach to Designing an Adaptive Lighting Environment. In *The 7th International Conference on Intelligent Environments*, Nottingham, UK, July 2011.

[87] B. Martin. Instance-based Learning: Nearest Neighbor with Generalization. Technical report, University of Waikato, 1995.

[88] A. K. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, 1999.

[89] J. McNames. A Fast Nearest Neighbor Algorithm Based on a Principal Axis Search Tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:964–976, 2001.

[90] V. Menkovski, G. Exarchakos, and A. Liotta. The value of relative quality in video delivery. *Journal of Mobile Multimedia*, 7(3):151–162, 2011.

[91] V. Menkovski and A. Liotta. Adaptive psychometric scaling for video quality assessment. *Signal Processing: Image Communication*, 27(8):788–799, 2012.

[92] M. Miki, E. Asayama, and T. Hiroyasu. Intelligent Lighting System using Visible-Light Communication Technology. In *IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6, 2006.

[93] M. Miki, T. Hiroyasu, and K. Imazato. Proposal for an intelligent lighting system, and verification of control method effectiveness. In *IEEE Conference on Cybernetics and Intelligent Systems*, volume 1, pages 520 – 525, 2004.

[94] T. Mitchell. *Machine Learning*, chapter 8, pages 239–258. Mcgraw Hill, 1996.

[95] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, Inc., 1991.

[96] L. M. Mäthger, N. Shashar, and R. T. Hanlon. Do cephalopods communicate using polarized light reflections from their skin? *The Journal of Experimental Bioloty*, 212:2133–2140, 2009.

[97] A. Muaremi, B. Arnrich, and G. Troster. Towards Measuring Stress with Smartphones andWearable Devices DuringWorkday and Sleep. *BioNanoScience*, 3(2):172 – 183, 2013.

[98] K. P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2013.

[99] W. J. Nash and T. Marine Research Laboratories. The Population biology of abalone (Haliotis species) in Tasmania. 1, Blacklip abalone (H. rubra) from the north coast and the islands of Bass Strait. Technical report, Sea Fisheries Division, Tasmania, 1994.

[100] N. Natarajan, I.S. Dhillon, P. D. Ravikumar, and A. Tewari. Learning with Noisy Labels. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 1196–1204, 2013.

[101] U. Neisser, G. Boodoo, T. J. Bouchard, A. W. Boykin, N. Brody, S. J. Ceci, D. F. Halpern, J. C. Loehlin, R. Perloff, R. J. Sternberg, and S. Urbina. Intelligence: Knowns and unknowns. *American Psychologist*, 51:77–101, 1996.

[102] N. J. Nilsson. *Introduction to Machine Learning*. Stanford AI Laboratory, 1998.

[103] S. Offermans, A. K. Gopalakrishna, H. A. van Essen, and T. Ozcelebi. Breakout 404: A Smart Space Implementation for Lighting Services in the Office Domain. In *Proceedings of the 9th International Conference on Networked Sensing Systems*, pages 1–4, 2012.

[104] S. Offermans, H. v. Essen, and B. Eggen. Exploring a Hybrid Control Approach for enhanced User Experience of Interactive Lighting. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, pages 1–9, 2013.

[105] S. Offermans, H. v. Essen, and B. Eggen. User interaction with everyday lighting systems. *Personal and Ubiquitous Computing*, 18(8):2035 – 2055, 2014.

[106] H. Park, J. Burke, and M. B. Srivastava. Intelligent Lighting Control using Wireless Sensor Networks for Media Production. *KSII Transactions on Internet and Information Systems*, 3(5):423–443, 2009.

[107] C. Perlich, F. Provost, and J. S. Simonoff. Tree Induction vs. Logistic Regression: A Learning-Curve Analysis. *Journal of Machine Learning Research*, 4:211 – 255, 2003.

[108] Claudia Perlich. *Learning Curves in Machine Learning*, pages 577–580. Springer US, 2010.

[109] D. M. W. Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation. Technical report, Flinders University, 2007.

[110] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[111] V. K. Rai and M. M. Laloraya. Correlative Studies on Plant Growth and Metabolism II. Effect of Light and of Gibberellic Acid on the Changes in Protein and Soluble Nitrogen in Lettuce Seedlings. *Plant Physiology*, 42(3):440–444, 1967.

[112] T. R. Reed, N. E. Reed, and P. Fritzson. Heart sound analysis for symptom detection and computer-aided diagnosis. *Simulation Modelling Practice and Theory*, 12 (2):129–146, 2004.

[113] W. Research. Analyzing and Interpreting Data. Technical report, Wilder Foundation, 2009.

[114] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.

[115] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[116] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education Limited, 3 edition, 2009.

[117] S. Shalev-Shwartz. *Online Learning: Theory, Algorithms and Applications*. PhD thesis, Hebrew University, 2007.

[118] V. Singhvi, A. Krause, C. Guestrin, J. H. Garrett Jr., and H. S. Matthews. Intelligent Lighting Control Using Sensor Networks. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pages 218–229, New York, NY, USA, 2005.

[119] K.C.H.J. Smolders and Y.A.W. de Kort. How do you like your light in the morning? Preferences for light settings as a function of time, daylight contribution, alertness and mood. In *22nd International Association People-Environment Studies Conference*, 2012.

[120] K.C.H.J. Smolders, Y.A.W. de Kort, and P.J.M. Cluitmans. A higher illuminance induces alertness even during office hours: Findings on subjective measures, task performance and heart rate measures. *Physiology and Behavior*, 107:7–16, 2012.

[121] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In *AI 2006: Advances in Artificial Intelligence*, volume 4304, pages 1015–1021. Springer Berlin Heidelberg, 2006.

[122] R. J. Sternberg and W. Salter. *Handbook of Human Intelligence*. Cambridge University Press, 1982.

[123] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[124] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multi-Label Classification of Music into Emotions. In *Proceedings of The International Society for Music Information Retrieval*, pages 325–330, 2008.

[125] G. Tsoumakas and I. Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing & Mining*, 3(3):1–13, 2007.

[126] J. A. Veitch and R. Gifford. Lighting effects on health, performance, mood, and social behavior. *SAGE social science collections*, 28(4):446–470, 1996.

[127] B. Vyssokia, N. Praschak-Riedera, G. Sonneckb, V. Blümlc, M. Willeita, S. Kaspera, and N. D. Kapusta. Effects of sunshine on suicide rates. *Comprehensive Psychiatry*, 53(5):535–539, 2012.

[128] J. Wang, P. Zhao, and S. C. H. Hoi. Exact soft confidence-weighted learning. In *International Conference on Machine Learning*, 2012.

[129] A. B. Watson and L. Kreslakeb. Measurement of visual impairment scales for digital video. In *Proceedings of the SPIE*, pages 79–89, 2001.

[130] Wikipedia. Intelligence. http://en.wikipedia.org/wiki/Intelligence, Jul 2014. [Online; accessed Jul-2014].

[131] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*, chapter 1, pages 21–26. Morgan Kaufmann, 2005.

[132] R. J. Wurtman. The Effects of Light on the Human Body. *Scientific American*, 233(1):66–77, 1975.

[133] P. V. Young. *Scientific Social Surveys and Research*. Prentice-Hall, 1966.

[134] S. N. Young. How to increase serotonin in the human brain without drugs. *Journal of Psychiatry and Neuroscience*, 32(6):394–399, 2007.

[135] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu. Missing Value Estimation for Mixed-Attribute Data Sets. *IEEE Transactions on Knowledge and Data Engineering*, 23:110–121, Jan 2011.

[136] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.

List of Publications

This list includes all the publications of Aravind Kota Gopalakrishna till date.

Conference Papers

- A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, J. J. Lukkien, *Statistical Inference for Intelligent Lighting: A Pilot Study*, 8th International Symposium on Intelligent Distributed Computing (IDC 2014), Madrid, Spain, Sep 2014.
- A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, J. J. Lukkien, *Relevance Prevails: Missing Data Treatment in Intelligent Lighting*, The 3rd International Conference on Man-Machine Interactions (ICMMI 2013), pp 369-376, The Beskids, Poland, Oct 2013.
- A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, J. J. Lukkien, *Relevance as a Metric for Evaluating Machine Learning Algorithms*, The 9th International Conference on Machine Learning and Data Mining (MLDM 2013), New York, USA, Jul 2013, Lecture Notes in Computer Science (LNAI 7988), pp. 195-208, Berlin: Springer.
- A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, J. J. Lukkien, *Treatment of Missing Data in Intelligent Lighting Applications*, The 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (UIC/ATC 2012), pp. 1-8, Fukuoka, Japan, Sep 2012.
- A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, J. J. Lukkien, *Exploiting Machine Learning for Intelligent Room Lighting Applications*, 6th IEEE International Conference on Intelligent Systems (IS 12), pp. 406-411, Sofia, Bulgaria, Sep 2012.
- S. Offermans, A. K. Gopalakrishna, H. van Essen, T. Ozcelebi, *Breakout 404: A Smart Space Implementation for Lighting Services in the Office Domain*, Ninth International Conference on Networked Sensing Systems (INSS 2012), pp. 1-4, Antwerp, Belgium, June 2012.
- R. Sivaraj, A. K. Gopalakrishna, M. G. Chandra, P. Balamuralidhar, *QoS-enabled Group Communication in Integrated VANET-LTE Heterogeneous Wireless Networks*,

The 7th IEEE International Conference on Wireless and Mobile Computing (WiMOB 2011), pp. 17-24, Wuhan, China, Oct 2011.

- K. G. Aravind, M. M. M. Pai, *A Novel Adaptable Routing Protocol for Wireless Sensor Networks*, International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA 2010), pp. 373-378, Fukuoka, Japan, Nov 2010.
- K. P. Chethan, K. G. Aravind, S. Jayaraman, P. Balamuralidhar, *FENCE to Prevent Deforestation using An Event Based Sensor Network*, International Conference on Chemistry and Chemical Engineering (ICCCE 2010), pp. 322-324, Kyoto, Japan, Aug 2010.
- S. G. Harihara, M. G. Chandra, B. Janakiraman, K. G. Aravind, S. Kadhe, P. Balamuralidhar, B. S. Adiga, *SpreadStore: A LDPC Erasure Code Scheme for Distributed Storage System*, The 2010 International Conference on Data Storage and Data Engineering (DSDE 2010), pp. 154-158, Bangalore, India, Feb 2010.
- K. G. Aravind, T. Chakravarty, M. G. Chandra, P. Balamuralidhar, *On the Architecture of Vehicle Tracking System Using Wireless Sensor Devices*, International Conference on Ultra Modern Telecommunications (ICUMT2009), pp. 1-5, St. Petersburg, Russia, Oct 2009.

Journals

- A. K. Gopalakrishna, T. Ozcelebi, J. J. Lukkien, A. Liotta, *Relevance as a Metric for Handling Non-Deterministic Multiple Output Problems in Machine Learning* (Submitted to Data Mining and Knowledge Discovery, Springer)

Patent

- A. K. Gopalakrishna, M. M. M. Pai, *Multi-service Adaptable Routing Protocol for Wireless Sensor Networks*, Manipal Institute of Technology, India, Patent No. US8295280.

Technical Reports

- A. K. Gopalakrishna, T. Ozcelebi, A. Liotta, J. J. Lukkien, *Relevance As a Metric for Evaluating Machine Learning Algorithms*, Computer Science Report, No. 13-04, Eindhoven University of Technology, April 2013.
- K. G. Aravind, *Wireless Sensor Network for Transmission and Distributed Line Monitoring*, TCS Internal Report, 2009-10.

Demos

- S. Offermans, A. K. Gopalakrishna, H. van Essen, T. Ozcelebi, *Breakout 404: A Smart Space Implementation for Lighting Services in the Office Domain*, Ninth International Conference on Networked Sensing Systems (INSS 2012), Antwerp, Belgium, June 2012.

Honors

- Paper entitled *Treatment of Missing Data in Intelligent Lighting Applications* received Outstanding paper nominee.
- Paper entitled *Relevance as a Metric for Evaluating Machine Learning Algorithms* is one of the top publications of Intelligent Lighting Institute (ILI), TU/e (2013-14).

Curriculum Vitae

Aravind Kota Gopalakrishna was born on 7th of October 1983 in Bangalore, India. He finished Bachelor of Engineering (B.E.) in Electronics and Communication Engineering from Visvesvaraya Technological University, India during June 2005. Subsequently, he completed Master of Technology (M.Tech) in Information and Communication Network Engineering from Manipal University, India during July 2008.

Before joining the System Architecture and Networking (SAN) group at the Eindhoven University of Technology (TU/e) as a PhD student, he worked at the Tata Consultancy Services (TCS) Innovation Labs, Bangalore, India as a researcher from September 2008 to December 2010. He worked on routing issues in wireless sensor networks and designing communication network architectures for specific applications. From January 2011, he started PhD at the SAN group, TU/e, of which the work is presented in this dissertation. His current research interests include context-aware systems, intelligent systems, applied machine learning, data mining and statistical analysis.