Olof Andreas Bergman

Prediction of personalized speed skating results using Case-Based Reasoning

Master's thesis in Artificial Intelligence
Supervisor: Agnar Aamodt
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Olof Andreas Bergman

Prediction of personalized speed skating results using Case-Based Reasoning

Master's thesis in Artificial Intelligence
Supervisor: Agnar Aamodt
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Case-based reasoning (CBR) is an approach to problem-solving used in research for sports science in the past years. CBR is an intelligent experience-based solution solving system explained as similar problems have similar solutions, and easily adapted to various fields. In this work, we use case-based reasoning for predicting best possible finish-times for speed skaters given various external conditions.

With inspiration from related research in recommendation systems for other sports, we studied a system handling the factors affecting speed skating and retrieving the most similar races for further prediction. The CBR system was modeled with the open-source software *myCBR Workbench* and *SDK*. This software retrieves cases with a restful API provided by the *SDK* based on the local-global similarity principle also defined in *myCBR Workbench*.

Looking at the results, we conclude that a CBR system like this is suitable for our problem statement. Speed skating offers multiple non-numeric features that can make a significant difference in the results. We tested two strategies for calculating new finish-times, where we found that the median strategy performed the most optimistic results, and mean strategy had less consistency. We experimented with two retrieval approaches where the use of non-personal-best times gave the most consistent results due to the knowledge base included more applicable cases than the season-best approach. A possible improvement upon our system is to implement the *revise* and *retain* process, so the CBR model use experience from solved cases and evaluates the non-numerical parameters.

Sammendrag

Case-based reasoning (CBR) er en metodikk for problemløsning som de siste årene har blitt brukt i forskning for idrett. CBR er et intelligent opplevelsesbasert system, som baserer seg på teorien om at lignende problemer har lignende løsninger, og dermed lett kan tilpasses en rekke ulike fagområder. I dette arbeidet bruker vi CBR for å forutsi en best mulig sluttid for en skøyteløper gitt ulike eksterne forhold.

Med inspirasjon fra relatert forskning om lignende systemer, har vi undersøkt er system som håndterer de viktigste faktorer som påvirker skøyteløp og finner de mest like tilfellende som grunnlag for prediksjonen. CBR-systemet ble modellert med den åpen programvaren *myCBR Workbench* og *SDK*. Denne programvaren henter like tilfeller med et API laget av *SDKen*. Prosessen er basert på det lokale-globale likhetsprinsippet og blir definert *myCBR Workbench*.

Når vi ser på resultatene, kan vi konkluderer med at et CBR-system som dette passer problemstilling. Skøyteløp tilbyr flere ikke-numeriske faktorer som gjør en betydelig forskjell på resultatene. Vi har testet to strategier for å beregne nye sluttider, og fant median strategien som den mest optimistiske, og gjennomsnittlig strategi den som hadde mest variasjoner. Vi eksperimenterte med to ulike metoder for å hente ut like tilfeller der bruk av ikke-personlige-beste tider ga de mest konsistente resultatene på grunn av at kunnskapsbasen inkluderer flere gyldige tilfeller enn metoden som bruker sesong-best. En mulig forbedring til vårt system er å implementere *revise* og *retain* prosessen, slik at CBR-modellen bruker erfaring fra allerede løste problemer og evaluerer betydelsen av ikke-numeriske parametere.

Preface

This master thesis is a part of the master degree at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU). Agnar Aamodt (IDI NTNU) has supervised our work at NTNU, and Håvard Myklebust at University of Stavanger (UiS) helped us with speed skating specific problems.

In our work, we define case-based reasoning (CBR) system for predicting best possible finish-times for speed skaters given various external conditions. The goal is to design, implement and test a CBR system that takes the factors affecting speed skating results into account. We used *myCBR Workbench* and *SDK* as a tool for CBR modeling.

This work has been challenging and inspiring since we cooperated with two different knowledge areas. We find it important that this work can contribute to both communities, computer science, and speed skating. In the pre-study phase, a lot of research had to be done in order to understand the important factors of speed skating and how to explain them in a machine learning environment.

We want to thank Amar Jaiswal (Ph.D. Candidate at NTNU) for guidance with *myCBR* and Håvard Myklebust for support with domain-specific problems in speed skating. Lastly, we would like to acknowledge Agnar Aamodt for competent supervision and support throughout this year.

Olof Andreas Bergman

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this work, we study the use of case-based reasoning (CBR) systems for predicting best possible finish-times for speed skaters given various outer conditions. Speed skating is a competitive form of ice skating [63] and there are many variations of speed skating. We will be referring to long-track speed skating as speed skating, as in the Olympics. This thesis takes inspiration from related studies [52; 53; 27] about using Artificial Intelligence (AI) for recommending and predicting results in sports. The goal is to design, implement, and test a knowledge-based system that takes the factors affecting speed skating results into account.

1.1 Motivation

This work is a collaboration between the Department of Computer Science (NTNU) and Håvard Myklebust from University of Stavanger (UiS). Myklebust is experienced within the speed skating domain and used as a domain knowledge expert in this work. There are well-documented studies [32; 6] on AI experiments including sports, however, never applied in speed skating. Speed skating is different from other sports studied because the results are highly sensitive for external conditions [31]. This research is the first step towards a system where coaches can provide live feedback and anticipate consequences from mistakes to their athletes during a race. We want to investigate what sequences of the CBR Cycle [2] that matches the problem statement best, and the benefits of interdisciplinary research. Recently, Smyth and Cunningham published two articles [52; 53] that caught attention studying running a marathon with pacing recommendations from a CBR system. Both articles are reviewed in the related research chapter, and influence this implemented system design.

Numerous studies have been published directly towards speed skating with focus on pacing patterns, endurance, race analysis for long distance and middle distance and altitude differences [56; 31; 11; 57]. We discovered in early stages that speed skating has great potential for improvement as we study how a race can be carried out. Pacing, altitude, and physiological differences will affect the results [31; 58; 11] and will give a huge advantage if a prediction ahead of a competition can give the athletes some guidelines. The objective of this thesis is to combine the studies that focus on speed skating with a problem-solving method. There have not been many studies combining speed skating and artificial intelligence, and it is an undiscovered field of expertise combining these two domains and therefore will this provide knowledge and experiences to the areas.

1.2 Goals

This work aims to predict an achievable and individual finish-time for a given speed skater with a priority of $3000m$ for women. We want to investigate how to use CBR and knowledge-based systems to find similar cases for predicting the new finish-times. The goal is to integrate numeric and non-numeric parameters to see if there is a difference in similarity and prediction result. There are different ways to predict results, where this work focuses on achieving reasonably improved finish-time based on the athlete's previous races and query conditions instead of predicting personal bests. During the design and implementation, an important finding will be to see what challenges working with AI and sports will evolve.

In this work, we divide the desirable achievements and goals into two segments — first, abstract goals with a focus on strengthening domain expertise to both the speed skating and the AI. Second, specific goals for developing and implementing CBR systems. In the literature review, related studies connecting AI and sport science build the domain knowledge. It is important to emphasize that the research questions and goals of this work will focus more on the system design and case representations rather than achieving the best predictions. As this task aims at two fields of study, it is essential to provide an understanding of both the AI community and the speed skating world.

1.3 Scope of Work

The scope of this work includes design, implementation, and testing portions of a bigger intentional CBR system. The intentional design scope would have been to implement and evaluate a system that is fully responsive with the CBR Cycle [2] and uses the athletes and coaches opinions and expertise to improve the solution space. We focus on the first design components and then evaluate the opportunities to continue developing.

The scope also includes researching the essential factors affecting speed skating races, CBR methodologies, and technologies essential for achieving the goals of this work. The implementation of the CBR system include evaluating and using the open-source software *myCBR*[1]. However, Chapter 4 cover the full system design and the phases implemented and evaluated in this work.

[1] `https://github.com/amardj/mycbr-rest-example`

1.4 Research Questions

In light of the goals presented, related work analyzing prediction using AI and the possibility to improve speed skaters' results, there are three research questions the thesis is designed to examine. The goal is to design a case-based reasoning system with numeric and non-numeric parameters. Based on the retrieved cases, use various strategies to determine the best possible and achievable finish-times. To clarify these goals, we have formulated three questions:

Q1: **How can we use CBR for prediction of results for speed skating?**

Q2: **How can an combination of numeric and non-numeric parameters improve the prediction?**

Q3: **Which strategy is the most suitable for prediction of speed skating results and which attributes are the most valuable for an accurate prediction?**

Throughout the thesis, these research questions are referred to and will be the foundation of the system designed. In the Discussion, we will go through all three questions and summarize the experiment.

1.5 Expected Results

The expected results of this work are:

∗ A CBR system that predicts a new best possible finish-time for a given scenario. The focus will be on designing a system rather than achieving least possible error.

∗ A reproducible method for future stakeholder to improve the CBR system.

This project includes two different disciplines, where expectations focusing on the ability of machine learning to improve sports science with recommendation systems.

1.6 Summary

We investigated how to use CBR for predicting results for speed skating, where we designed a system with several case features that characterize a speed skater. Our work concludes that a CBR system like this is suitable for our problem statement. We found multiple race factors affecting speed skating results that could transform into a non-numeric case feature; however, our results are mainly affected by numerical features We tested two strategies for calculating new finish-times and concluded that *median* strategy is the most encouraging strategy — and *mean* the most stable for retrievals with a larger amount of cases. The most valuable attributes are the external race condition features, such as altitude. We experimented with two retrieval approaches where the use of non-personal-best times gave the most consistent results due to the knowledge base included more applicable cases than the season-best approach.

1.7 Structure of Thesis

The thesis contains eight chapters grouped in three essential parts, design, implementation, and experiment. Figure 1.1 illustrates the three parts, as an inverted pyramid where we study a full system design, implement components from the design, and run experiments on parts from the implementation. The design chapter overview of the CBR system design and detailed description of all pieces involved. Second, is an accurate representation of the implementation of the system with case representation, CBR phases, and the prediction strategies. Lastly is the experiment results where we test and evaluate parts of the implemented systems.

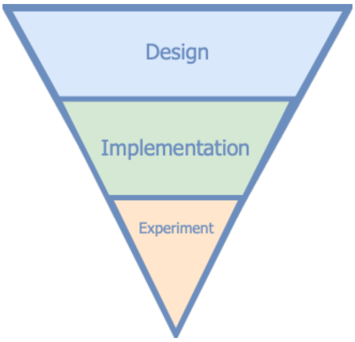


Figure 1.1: Pyramid Structure of this thesis

The eight chapters creating this research are Chapter 1) A high-level introduction to the topic area and our motivation for the project. It also provides the projects research questions and goals. Chapter 2) Background knowledge about speed skating and case-based reasoning with a focus on the CBR Cycle phases. Chapter 3) A description of related work with a focus on AI in sports and important factors in speed skating. Chapter 4) Describes the intentional system design for a desirable CBR system and an explanation of the retrieval and reuse phase. Chapter 5) A detailed documentation of the implementation of the described system. Chapter 6) A visualization of the experiment results. Chapter 7) Discussing the findings in this work and evaluate obstacles and limitations. Chapter 8) Conclude the findings according to the research questions and discuss improvements for further work.

Chapter 2

Background

In this chapter is background knowledge about speed skating and case-based reasoning explained. It is necessary to clarify concepts and methodology to understand the system.

2.1 Speed Skating

There are a few things one need to know about ice skating before continuing the thesis. Ice skating contains two different disciplines, short track and long track speed skating [64]. This work refers to long track speed skating as speed skating. Both disciplines include fast ice skating in circles, however, with a few differences. The biggest difference is the size of the rink, where long-track rinks have 200 m of corners and 200 m straight on every lap making it a total of 400 m. Short track rinks are 111 m in length, and the same size as an international-sized ice hockey rink [60] and holds a different competition style than long track. Figure 2.1 shows the Olympic Oval in Calgary.

Mid-19th-century Norway held the first ice skating race [62]. Speed skating entered the Olympic program in 1916. There are several different competition formats where Allround, Sprint, Single distance, Team pursuit, and marathon are the most popular. In the single distance are the usual distances 500 m, 1000 m, 1500 m, 3000 m, 5000 m, and 10,000 m. 3000 m are for women only and 10,000 m for men only [29; 62]. Allround is the oldest format where the skaters skate four different distances (500 m, 1500 m, 5000 m and 10,000 m for men and 500 m, 1500 m, 3000 m and 5000 m for women) and the total time from all distances create the ranking.

We will experiment using results from the World Allround Championships to predict the next World Cup race. In World Cup competitions are skaters competing in the single distance. During a race do the skaters' match in pairs, and switch lanes every round so that both skaters cover the same distance. There are situations where some of the skaters start in a quartet, explained later in Chapter 3.2.2. The most influential country in speed skating is the Netherlands, while many other countries are highly competitive such as Canada, Norway, Germany, Russia, Czech Republic.

Figure 2.1: Olympic Oval in Calgary, Alberta [17]

In this work, we are referring to *PB*, *SB* and *nPB*. They are abbreviations for a speed skater's *personal best* time, *season best* time and *non-personal best* time. A *nPB* represent a a recorded finish-time that not is a personal best. *PB* and *SB* represents best time ever performed, and best time performed last season (2017/2018).

2.2 Case-Based Reasoning

Case-Based Reasoning (CBR) is a knowledge, and experience-based methodology explained as similar problems have similar solutions [2; 24]. We will use CBR as AI methodology in our system. CBR combines machine learning with relateable problem-solving, and the machine learning community have a major influence on the development of CBR. They are the driving power to keep developing the methodology [1]. The reason we use CBR instead of Neural Networks is that all cases are different with distinctive prerequisite where this thesis will explore whether similar cases can be used to predict improved finish-times, which is achieved by implementing experience. Figure 2.3 shows the four different processes in the CBR cycle (*retrieve*,*reuse*, *revise*, *retain*), known as the "4 REs" [30]. In short, various events are collected, in our thesis, speed skating races, which become cases in the case base. After retrieving the most similar cases from a given query, we are reusing the proposed solutions. Then revise the outcome and retain the improvements or changes in the case solution.

Our CBR system will mostly involve the *retrieve* and reuse process, as we are not modeling an extensive *revise* process. The *revise* process is, in many cases, a human process where the objective is to decide if the input problem received the best solution. In our work, this could have been an evaluation from the coach and the athlete. Chapter 8.1 will dig deeper into how this system can use the remaining processes in the CBR cycle.

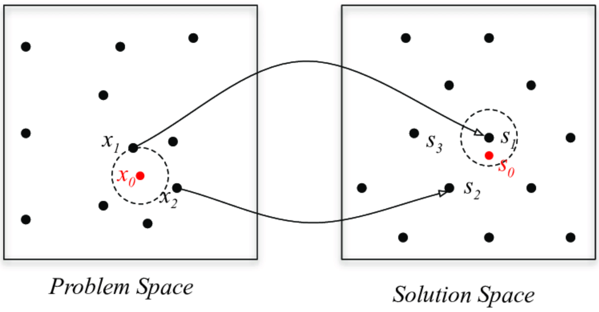


Figure 2.2: Relationship between problems and solutions in CBR [15]

Since CBR is memory driven and learns from experiences, are environments where it is difficult to formalize an active area [16]. What makes CBR so important and exciting are the various areas with implementation possibilities.

An example is finding solutions and treatments for diseases and symptoms [9; 7]. What is strengthen the CBR approach is that the solved cases will be immediately available and retained to the problem and solution space for future problems. CBR is an incremental and sustained learning system [2]. Figure 2.2 illustrates the relationship between the problem space and the solutions space. X_0 is the new problem to solve and S_0 the new solution created. X_1 represents a solved problem and S_1 a stored solution. The distance retrieved $(X_1 - X_0)$ between the new problem and the solved problem increases when the similarity between them decreases [15].

Generally, the CBR cycle contains the following four processes:

1. `Retrieve` the cases with highest similarity to query case
2. `Reuse` the solution and experiences from the retrieved case for solving the query case.
3. `Revise` the recommended solution
4. `Retain` the useful parts for future similar problems.

In this thesis, we (1) retrieve the most similar cases defined by similarity functions, which will be explained later in this chapter. (2) Reusing the nPB, SB, and PB times from the retrieved cases to calculate a finish-time. In a complete system, then continue with (3) revising the solved solutions and lastly (4) retain the experiences to future problems. Figure 2.3 illustrates the CBR cycle.

A complete CBR system needs to develop the four knowledge containers, case base, similarity measures, adaptation knowledge, and vocabulary where each container have specific tasks [4; 39]. Figure 2.4 shows the knowledge containers and the interaction between

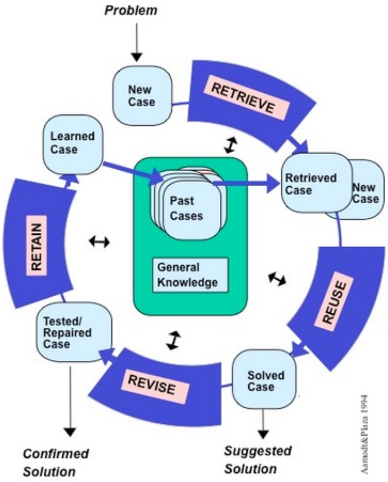


Figure 2.3: CBR Life Cycle [2]

them. In a CBR system will these containers include sub-containers and all containers needs to be interacted for a problem to be solved. Examples of sub-containers in the vocabulary are retrieval attributes, input attributes, and output attributes. Retrieval attributes are beneficial in similarity measures, and input attributes for experience rules and output attributes for information regarding the user. Vocabulary explains the data structure and how to represent the data in the form of attributes, functions, and relations. Usually and in this thesis, the structure are an attribute-value representation. Similarity measures calculate the distance (with the unit interval [0,1]) between two problem descriptions in a continuous feature space F. Sub containers in Similarity measures are local similarity and amalgamation function. The local similarity includes knowledge on the feature level, and amalgamation function calculates similarity on a concept level using the local similarities (utility knowledge). The Case Base (CB) is the memory of the system. CB contains the experience as cases or combination of cases. Adaptation knowledge is usually called solution transformation and takes care of transforming the stored cases to fit the query problem. [44; 39].

There are several possibilities and advantages with the CBR structure. One of them is that the containers can be changed locally. Problems can not be solved without all four containers, however, are the containers independent and can we change the containers without affecting the other. Because of the container independence can one develop knowledge by update the containers separately. Another advantage is that containers make the system flexible and easy to maintain.

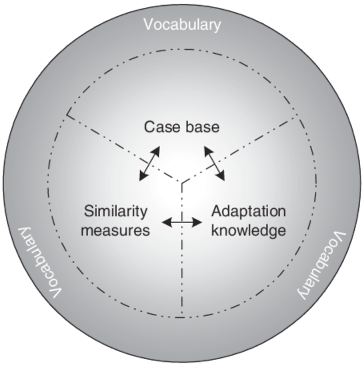


Figure 2.4: The four knowledge containers in CBR [44]

2.2.1 Similarity in CBR

As mention earlier in this chapter, the purpose of CBR is to solve a new problem based on experience and knowledge from similar problems. There are two ways to deciphering the problem, either find cases with a similar problem to the query problem or find cases easily adaptable to solve the query problem [59; 2]. We find similar cases using the *local-global principle* [55]. The principle divides the similarity measures in local similarities on individual attributes, and global similarities combining all local similarity functions [40; 22]. Local similarities define the functions that compare specific features in each case. Global similarity functions combine local similarity functions and compare cases. A global function can be complex but normally are a simple Euclidean Distance used. Other functions often used are weighted average and sum (min and max).

We use asymmetric similarity functions in the CBR system. The asymmetric similarity is defined as the distance between two cases in a continuous feature space F_i, where the roles in the case representation are important [38]. Feature a_i and b_i are dependent of each other and needs to be represented the same way to be equal, $d_i(a_i, b_i) \neq d_i(b_i, a_i)$ [59]. In Figure 2.5 we can see that figure number two illustrates that smaller values are better, and the third figure prefers larger values. Symmetric functions will behave independent of the roles of features being compared, $d_i(a_i, b_i) = d_i(b_i, a_i)$. Global similarities are comparing on the case level, and normally using Euclidean Distance as similarity measures. Similarity is represented in interval $[0, 1]$ as the relation: $Sim(a, b) = 1 - Dis(a, b)$. The Euclidean distance is the most common use of distance and calculates the root square differences between two objects in Euclidean space, where most similar objects will have the smallest distance. Euclidean Distance is often the standard nearest neighbor classifier together with Weighted Sum [33; 46]. Equation 2.1 illustrates how the Euclidean Distance

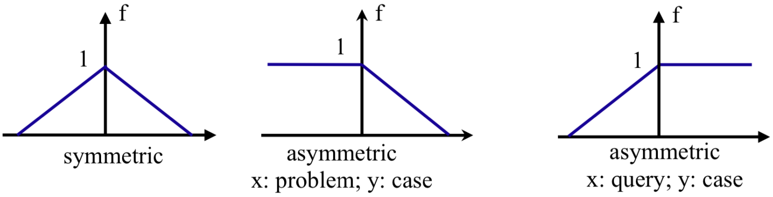


Figure 2.5: Symmetric and asymmetric similarity functions [40]

d between two points (q and p) is calculated with Pythagorean formula.

$$d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2.1}$$

2.2.2 Retrieval in CBR

The purpose of the retrieval process is to retrieve similar cases to a query problem and then use the retrieved solutions to solve the new problem. Since CBR uses memory (i.e., case base)[41; 2] as a basis for retrieving cases, will the next question be how to compare and find the best cases for the specific purpose? CBR systems are dynamic in the way that one can use the same case base for various retrievals and purposes, where only the similarities functions differ. What defines a retrieval is the similarity functions and the objective of the problem. In some CBR designs, including this design, is it be more efficient to use the similarity of the cases called surface features [30], which are attribute-value pairs provided as a part of the description. These features will correspond to nPB, PB, and SB in this system. We want to receive the most similar and better-performing cases in a similarity-based retrieval for the query. The similarity of each surface feature computes from the local similarities and a global similarity function.

2.2.3 Reuse in CBR

In the reuse process is knowledge from the retrieved cases adapted. The process is suggesting a new solution based on the adapted experience from the old problems and solutions [41], see Figure 2.6. Reuse can vary in complexity, in some systems the reuse phase return the old solution, whereas in other systems will adaption be necessary. For medical use will adoption be essential [30; 7] as the retrieved problems contain differences from the new problems. In this work, we reuse the retrieved cases for calculation a new predicted finish-time. Adaptation can be obtained in various ways, and with varying complexity. Mántaras et al. [30] refers to two dimensions: what is changed in the retrieved solution, and how the changes are achieved. Aamodt and Plaza [2] states the same dimensions as *transformational* reuse when cases are reused, and *derivational* reuse when the method achieving the solution is adapted.

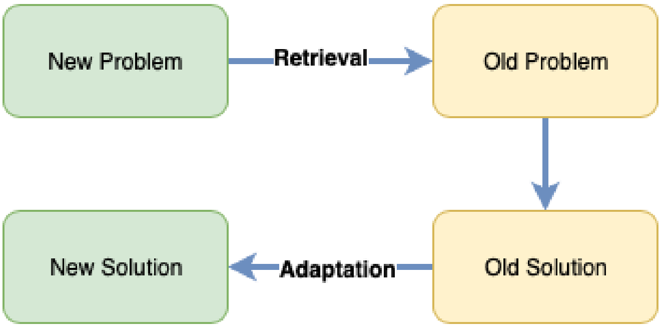


Figure 2.6: The Reuse principle (Adapted by [41])

The aim for the reuse process is to identify the differences between the new problem and the retrieved cases, as well ass which features from the retrieved cases that can be assigned to the new solution [49]. There is no guarantee that the retrieved cases offer a proper solution, and is why the reuse phase is essential for a CBR system. After adaption will the suggested solution be tested and if the system is satisfied, will the new solution be retained to the case base [41]. In Chapter 4, we will explain how we implement the reuse phase, and in Chapter 5 show examples of how we use the retrieved cases in our study.

2.2.4 Revise in CBR

It is needed to explain the revise process more careful because it is essential in an efficient and complex CBR system. The other processes are fundamental for the CBR system to work efficiently, but revising solved solutions precise increase the similarities in future work.

To maintain the CBR system is case retainment and learning from solved solutions essential. The revise process includes evaluating the new solved problem and merges with already existing knowledge in the memory [2]. When a problem is solved differently from the solutions in the solution space will a new case be generated and retained to the case base, and when the solution builds upon existing cases will relevant decisions and knowledge be included. This process is often manual where domain experts review the solutions and decide what to reuse. It is also important to emphasize that failures also needs to be retained. A new problem can, therefore, noted the possible failures. Search Engines, decide diagnoses and e-commerce are areas where revise is highly implemented [26; 7].

The reason why we do not include this step in our system is that our goal is to investigate the possibilities to use CBR in prediction and a combination of numeric and non-numeric parameters. Retrieval with efficient similarity function will be the first step in the system design where revision will be the next natural addition. We are doing a similarity-based retrieval where we reuse similar cases in another procedure, and therefore will a simple revise process require a richer case representation with more than surface features.

2.2.5 Retain in CBR

The last process in the CBR Cycle regards the retain of solved and tested cases. In this step, do the system update the case base with the new/learned case for future problems [41]. The retain process makes it possible to improve the CBR system and include new knowledge and experiences. If the revising process contains manual interaction will the retain process be essential so the solved case can be reused in future problems.

Usually can cases in the case base only be imported and not forgotten [41]. This can be a disadvantage, but by retaining cases, the case base will change and smoothly integrate necessary improvements. There are various strategies concerning how to best include the new cases in the knowledge base [30]. In general, will the solved solution be added as a new case in the case base, where there also exist more advanced strategies for adapting specific feature, and also manually processes are used.

This CBR system does not include the retain process, as it mainly focuses on retrieving previous results, and the new cases will be hypothetical cases and not actual results.

Chapter 3

Related Work

3.1 Related Research

The use of Artificial Intelligence (AI) to predict a result has been around for many years, and the case-based reasoning (CBR) has not only been used in the computer science field but used on various fields for many years. In Finance are there enormous opportunities with AI techniques. One example is how AI is frequently used by financial institutions to avoid bankruptcy and provide risk calculations [3]. In Health Science, AI and CBR is an essential aid [7; 9] and using Expert Systems helps solve issues and diagnoses more frequently. What makes CBR suitable for the medical domains is the ability to build advanced instance-based and expert systems handling challenging case representations and experiences for solving diagnoses, classifications, treatment planning, and knowledge management [7]. CBR systems quickly adopt new information and can retain and reuse knowledge and experience. CBR can reuse the data without a need for generalizing, which is an obstacle when using statistics as a solution solving method. In work by Bichindaritz and Marling [9] they also state that the health science and AI fields fit each other well because both fields are expanding and health science provides complex cases that challenge and pushes state of the art in AI forward. The two fields will continue to expand and the number of papers published every year will keep on increasing.

In the sports community, machine learning and AI is in its early phases [8]. In the last ten years, the field has been extended to multiple sports and areas where data collection and analyses can improve the results. In Weight Training [32], a study explained how AI techniques were used for the evaluation of exercises performed on training machines. Combining AI and sports allows for instant feedback and analysis, which is a breakthrough. Novatchkov and Baca concluded in the Weight Training article that by connecting sensors on the training machines and a supervised learning process the risk of injuries can be reduced as well as an optimization of the Weight Training based on the athletes professionally. Medicine and Sports are also in use of machine learning and AI. Bartlett [6] reviews the development of Artificial Intelligence in sports biomechanics over the years and concludes that in the future multi-layer Artificial Neural Networks (ANNs) will have an essential role in the analysis phase of sports and biomechanics. He states that the understanding of movement, techniques, and skill learning will increase. Bartlett was right, ever since he stated in 1995[25] that there was no evidence of the use of AI and sports biomechanics, the expert systems, and knowledge-based reasoning systems have increased. What this article shows is that ever since the early 90s the interest in moving the AI community into the sports field has been comprehensive.

Since this thesis involves prediction, it is essential to mention what kind of predictions that have been made associated with sports. We will discuss CBR and other prediction methods later in the chapter; however, it is essential to see how we can use other fields of AI in sports prediction. McCabe and Trevathan [27] presented in 2008 a paper about AI to predict of sporting outcomes. With Neural Networks only given necessary information will they predict the outcome of a sporting contest. This work is inspired by an earlier article also written by McCabe, and they conclude that there is an interest in modeling of features in a different noisy environment. A noisy environment is an environment with details that are affecting the data set, and in this case and also typically in cases associated with the sport will noise be related to the human factors, such as individual “form” of the athlete, injuries, motivation, and skills. What made the prediction process challenging and attractive at the same time was the numerous elements that can contribute to winning results. We proceed with the observation that the models were able to adapt quickly despite the basic information prepared. McCabe also mentions that further work will be moving towards different sports, whereas his work only included data from Rugby and Soccer. However there have been many different sports involved in work for prediction and analysis, as in 1981 was a work published by Riegel, where he inspected and analyzed the endurance in multiple sports such as running and swimming.

One of the sports that affects and engages most people is running. According to Statista [18], in the US only, over 60 million people attended a running or jogging trail in 2017. The interest in exploring running and the advantages people can gain from data analysis in the sport of running is big. The popularity in predicting and recommending runners pacing plans, tactical advice, and finish-times have increased in the last years. In 2017 and 2018, several reports were published regarding prediction approaches [52; 53; 8], and we will continue discussing this work and ideas later in this chapter. Data collection is easier in sports such as running simply because o the number of athletes participating in events all around the world. Therefore more relatable to the recommendations and analysis presented. In the article Running with Recommendation by Berndsen et al. [8], they examined the opportunities for systems based on knowledge from coaches and runners to give suitable recommendations for marathon runners. A simple K-Nearest Neighbours (KNN) model were used in the prediction. In the future, other endurance sports will apply the same techniques. An issue in this work is the time scale and how a runner can vary in achieving best times after their first marathon. What makes prediction and analysis in sports challenge is the human factor in that there are numerous of variables which are uncontrollable and vary from individual to individuals, such as age, mental and physical health and motivation during races. [8] illuminates an important factor in providing a recommendation that is understandable, engaging, and improves the runners confident before and during in this case, a marathon. A future add-on to a system like this is to involve a more personalized explanation and recommendation to increase the motivation and achieve maximum for the plans and race times predicted.

Moving towards the purpose of this work, we focus on the prediction and CBR approaches. The relevant part of the reviewed work was the issues stated by [8] and using the ideas of individual predictions. Dealing with noisy data set discussed in the work by McCabe and Trevathan [27] will also be discussed especially since the sport of speed skating includes numerous factors in addition to the actual race.

In 2017 Smyth and Cunningham used CBR and marathon runners as a study for predicting a best possible finish-time and a suitable race plan [52]. The work contains two parts, where both parts together will help achieve a new personal best time. Part one is to predict the new finish-time and the second to find a reasonable pacing plan for achieving the predicted time. We concentrate this work on predicting finish-times not pacing plans.Smyth and Cunningham 's work is relevant because it sits at the intersection between personal sensing, big data, and machine learning. For the research to be successful, they include runners that have completed at least two marathons. The data included only results from the London Marathon. The case representation includes one race with a non-personal-best time (nPB) and the fastest race, which is a personal-best race (PB). nPB is the case description and PB the case solution. Equation 3.1 shows a case c where m_i is a nPB race and m_j is a PB race for the runner r. By retrieving similar cases to c, the model can calculate a new finish-time and a pacing plan.

$$c_{ij}(r, m_i, m_j) = \langle nPB_i(r, m_i), PB(r, m_j) \rangle \tag{3.1}$$

The retrieved cases are filtered based on gender and finish-time within t minutes in nPB times. A weight w, illustrated in Equation 3.2 based on the difference between the query runners nPB finish-time and the retrieved nPB is the foundation in the three different approaches for calculating a new PB, **Best PB**, **Mean PB** and **Even PB**.

$$w(q, c) = \frac{q(nPB).finish}{c(nPB).finish} \tag{3.2}$$

Best PB is only using the case retrieved with the best PB. Shown in Equation 3.3 where q is a query runner, C the case retrieved, w the weight equation 3.2 and $Time$ the PB time from the best case.

$$PB_{best}(q, C) = w(q, C_{best} * Time(C_{best}(PB)) \tag{3.3}$$

Mean PB (Eq 3.4) calculates the weighted mean PB from the retrieved cases where C is a list of all retrieved cases and k number of cases retrieved.

$$PB_{mean}(q, C) = \frac{\sum_{\forall i \in 1..k} w(q, C_i) * Time(C_i(PB))}{k} \tag{3.4}$$

Even PB (Eq 3.5) uses the evenest race from the retrieved cases and have the same equation as Best strategy except C_{even} returns the evenest race instead of the best race.

$$PB_{even}(q, C) = w(q, C_{even}) * Time(C_{even}(PB)) \tag{3.5}$$

We will adopt the idea of having multiple approaches when calculating a new best finish-time to our work. This because we easily compare different approaches that take various parameters into account.

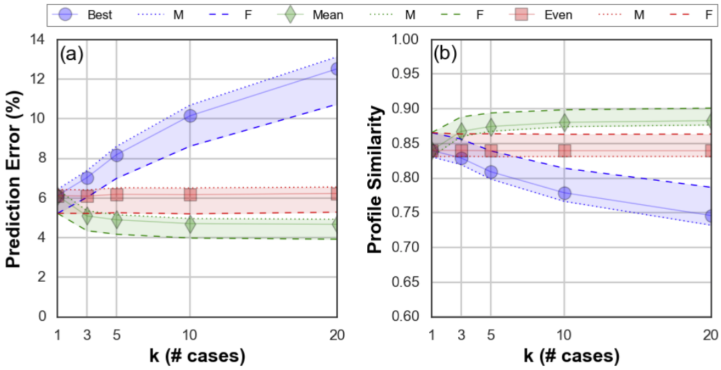


Figure 3.1: Prediction error (a) and pacing profile similarity (b) versus k for Best, Mean and Even strategies, and both genders [52].

For results and conclusions, they found out that the Mean CBR strategy was the most suitable approach for predicting an achievable PB finish-time. The three strategies did not behave equally when increasing the number of cases retrieved. Mean strategy has increasingly profile similarity and decreasing prediction error as k increases which is the most proper prediction strategy and predicts more accurate finish-times compared to the runners actual best time. Even strategy produces an error of 6% regardless of k compared to the lowest Mean error of 4.5%. However, and not surprisingly. Best PB strategy will perform worse with more cases retrieved, see Figure 3.1. Since the Best strategy performs well with a small case base, and the error increases as k increases make the approach too ambiguous. With a 12%, faster PB than nPB the Best CBR strategy wins over the other two strategies, however an improvement that big is not normal in marathon running. In the data set under 20%, of the runners achieved a personal best that big.

The runners finish-time have a notable on the error and similarity. Figure 3.2 (c) and (d) illustrates that fast runners will have an advantage of using the Best strategy and more ambiguous approaches than slow runners who benefit more from the Mean strategy. Mean strategy performs well across all finish-times, and Even comes in between Best and Mean.

To notice is the difference in accuracy between women and men. All of the three strategies perform better for women than men regardless of the number of cases retrieved. Women performing more even than men is not surprising because related work has shown that female runners usually stick to their pacing plans while men often tend to run more ambitious than the predefined plan [58]. As for future work, the article states that PB quality is an issue for prediction, and the time interval between the two compared marathons is essential to consider. They also plan to test the approach in more various marathons and see how that reflects the results.

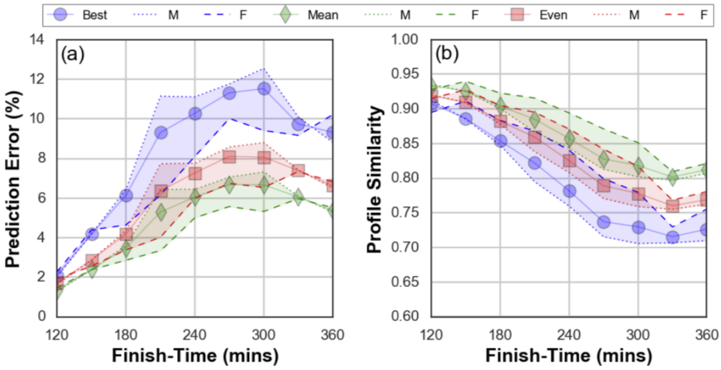


Figure 3.2: Prediction error (a) and pacing profile similarity (b) versus nPB finish-time for Best, Mean and Even strategies, and both genders [52].

We will not only use races from one specific tournament or competition but multiple World Cup races around the world with different conditions. According to the conclusion from Barry and Cunningham, a more tailored and personalized prediction could be achieved by using training data from various marathons/races. Whats strengthen the article related to our work is that both types of research involve individual performances where one can easily perceive a finish-time. On the other hand, what is different for the research compared to our work is that the prediction only involves numeric analysis, where the nPB and PB are the basis for the prediction. For a runner to achieve a personal-best, a lot of other parameters can be involved and discussed, such as weather conditions with wind and temperature, starting group of the marathon, the runners' health conditions. We will touch upon some of these parameters, and mostly focusing on the outer race conditions.

In 2018 a follow-up article [53] was published with a focus on what Smyth and Cunningham wanted to improve from the 2017 marathon prediction. A weakness with the 2017 article is that it is suitable for the runners with a reasonably recent race; however, the reality is that runners might have a more extended break between races and those who only run few races focus more on finishing than on achieving a personal best. They extend the case representation and includes more races in what they call landmark races as an abstract case feature, as they are higher-level features than the regular finish-times. A landmark race is a race that most likely can influence the case representation and prediction. The goal is to identify these landmark races and see whether a more extensive case representation is better for the prediction. In other words, since a landmark race represents an nPB race, will a richer case representation include more nPB races. The different landmark races are the following: a) The most recent race in the runner's history, b) the least recent (first) race in the runner's history, c) the runner's most varied; the race with the highest coefficient of variation of the segment pace, *d*) the least varied race in the runner's history; that is the race with the lowest pacing variation, *e*) the previous PB race; the runner's fastest race in their race, *f*) prior to the current PB, the personal worst race; the slowest race in the runner's history, and *g*) a pseudo race-record based on the mean of the runner's non-PB races.

The general conclusion in this paper is that their hypothesis is confirmed. The predictions will, in most cases, get better with a richer case representation, presented in Figure 3.3. We can also see that women tend to be slightly better to follow the race plan and achieve the best time according to the predictions [58]. They found out that not always a richer case design will give the best results. A case where they only used three landmark races scored as one of the best representations. So the key finding in this work is the difference in prediction error from simple to more complex case representations, and that representation does not always have to be the case with most parameters.

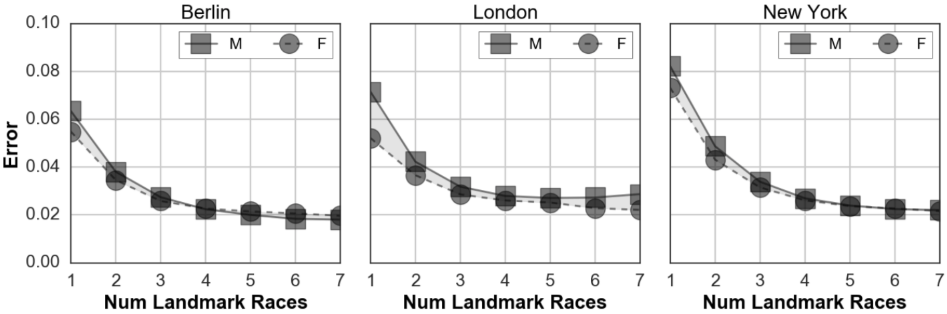


Figure 3.3: Figure shows the mean prediction error compared to the number of landmark races included in the case representation [53]

What strengthens Smyth and Cunningham's analysis is that they base the study out of weaknesses from earlier research, and improves it by using more realistic cases. Runners who only run in one city once a year may not be the perfect test runner for achieving a personal best, where the interests more or less would be in finishing in a reasonable time. In our work, it is essential to absorb as much research and experiences from similar cases and experiments. We will adapt and use a richer case representation. However, we will categorize the landmark races differently.

3.2 Related Speed Skating Knowledge

There are many similarities between runners and speed skaters, and also decisive factors that runners do not have to evaluate. The model will contain different case features which characterize a speed skater and World Cup race. To get a better understanding of the case features later explains will we go through the factors affecting speed skating and describe why they are essential.

To become a successful speed skater are several factors to overcome. The first is knowledge of pacing and race strategies. Pacing explains how fast a skaters skates to a certain moment for achieving the best possible time, and how to distribute the energy. Later in this chapter, we explain what the difference between female and male athletes are and how both groups adopt pacing. Another factor that is important to observe is the differences in performances in different altitude conditions, and how the barometric pressure affects the skater.

As the results of the prediction and whether it matches reality are dependent on the athlete's effort and day-to-day shape. It is important to look at how the athletes adhere to the various competitions and how the prioritization is about when the athlete is in their best shape. What strengthens speed skating in this context is that there is a minimal difference between championships and World Cups compared to other sports. Senior skaters perform in average 30% faster in the important competitions such as World Cup, Olympics and World Championships, which is important for prediction to see that the skaters overall perform on a high level each competition [31]. Compared to rowing where important competitions were 1.3% faster [50] than less important competitions, and swimmers where 0.9% faster as Olympics [36] than Pan Pacs, which is a less important competition in the eyes of Olympics.

3.2.1 Altitude in speed skating

The most important environmental race factor is the altitude of the rink, and also the relationship to the barometric pressure. The ice rinks in the World Cup is varying in altitude, where Salt Lake City and Calgary are the two highest above sea level. These two ice rinks are also where the majority of athletes have their records set. See Table 3.1 for an overview of ice rinks used in this work.

In an article regarding race factors affecting performances in speed skating, they stated that altitude resulted in average performance improvement for 3000m with $3.2\%(\pm 0.5)$ for senior skaters per 1000m increase in altitude [31]. Female skaters tend to have a bigger advantage of altitude than men. In [56], it is stated that women get a 4.6% faster finish-time in high altitude than men.

The reason why skaters have an advantage in high altitude rinks is that the barometric pressure decreases when the altitude increases. At any height, the air pressure represents the total weight of air molecules above, so in lower altitude, the air pressure is higher because of the number of air molecules above compared to high altitude where there are fewer molecules [14]. So in other words, since most of the particles are held close to the surface of the earth due to gravity, there are fewer molecules to move when skating in higher altitude conditions, so it is easier to achieve a higher pace.

By looking at where the world records in skates are set, one can see that for female skaters, about 63%, 7 out of 11 set in Salt Lake City (Utah) with an altitude of 1423 MASL, and the remaining four are in Calgary which is 1105 MASL. For male skaters are the conditions quite similar to 58% set in Salt Lake City and remaining in Calgary [54]. This shows that it is a great advantage for athletes to be in high altitude climate.

The altitude parameter, how high above the sea the competition unfolds, is the one parameter I will emphasize the most and performs the greatest variation to the prediction. Meters above sea level abbreviated to MASL in this task.

List of indoor speed skating rinks

Rink	**MASL**	**Location**
Calgary Olympic Oval	1105	Calgary, Canada
Utah Olympic Oval	1423	Salt Lake City, USA
Gunda-Niemann-Stirnemann-Halle	214	Erfurt, Germany
Vikingskipet	125	Hamar, Norway
Minsk Arena	209	Minsk, Belarus
Thialf	0	Heerenveen, Netherlands
Eisstadion Inzell	690	Inzell, Germany
Gangneung Oval	26	Gangneung, South Korea
Speed Skating Centre	120	Kolomna, Russia
Meiji Hokkaido-Tokachi Oval	79	Obihiro, Japan
M-Wave Nagano	346	Nagano, Japan
Ice Palace Krylatskoye Moscow	127	Moscow, Russia

Table 3.1: List of the rinks used in this study. Accessed from [61]

3.2.2 Division A and B

One certain factor, and especially in speed skating is how they compete. In most sports, there are similar conditions for everyone, regardless of ranking or division. In swimming have every swimmer an own lane with the same water conditions as the one next to you, and in track and field is it the same length and conditions in all lanes.

In individual races, skaters will compete in heats, also called Divisions based on the ranking [21]. In Division A the best-ranked skaters start, and the lower ranked skaters start in Division B. If there are no more than 20 entries all skaters start in division A. If there are 21-24 skaters signed, will Division A contain 12 skaters and if there are more than 24 skaters 16 of them will be in division A. What differs the two divisions is that Division A the skaters start in pairs and division B in quartets. Both divisions are scheduled on the same day with the same conditions. Divisions is an essential feature for the prediction because starting in pairs or quartets can have an impact on the race, pace, and finish-times.

3.2.3 Gender

As mentions earlier, women tend to keep their pace better than men [58]. According to an article published in 2016 about pacing strategies for 1500m speed skating focusing on gender and performance, the speed distribution profile is similar for both genders [11], where the races start with an accelerating part followed by a decrease in speed towards the end. However male skaters showed a higher performance than the female skaters, and also statistically more aggressively throughout the race which can have consequences on the finish-times and pace plan [31].

Several factors are confirming that male skater has a different strategy and different ways of pacing a race than female skaters [11]. One of the factors that will be focused on here and also on the basis that only 3000 meters for women are evaluated in the master thesis is the ability to pace themselves. In a study where they examined gender gaps for various Olympic sports, where speed skating has been involved since 1924, the conclusion was that the gender gap would remain in the future [57]. After a considerable inversion of differences for women and men, a little back in time, women and men are now developing at an equal pace, and therefore can the study on female skaters also transfer to male skaters. For the past 26 years, has there been observed a stable gap and appears to remain so. In speed skating, a gender gap is measured since 1989 at 6.95% [57].

Another factor in the difference between male and female skaters is how technique and biological differences play out. According to an article posted by Carlos Rafaell Correia-Oliveira1 [11], women, have biological disadvantages such as knee angle which makes them unable to have a position that is optimal for air friction and also causes loss of speed.

Chapter 4

Design

4.1 Intentional System Design

The following subchapter is a design specification of an intentional case-based reasoning system. It is important to emphasize that this is not the system implemented, but an illustration of how we determine a fully developed intentional system, see Figure 4.1. We have determined *retrieval* and *reuse* as the two most essential components for us to implement. *revise* and *retain* are equally essential, but since we are investigating how CBR solve prediction problems will retrieving and adapting case knowledge be a natural first step. We are focusing on and studying some components of the intentional CBR system design, see Figure 4.4. The following sub-chapters will elaborate the intentional phases and case representations. All figures are created with *draw.io*[1].

4.1.1 Case Representation

There are four different types of features necessary for a complete case representation, see Figure 4.2. *1*) Race results that include personal bests and other non-personal best times. *2*) Weather conditions for all races such as air temperature, ice temperature. *3*) Race conditions for all races in the case. Such as division, altitude, date of the race. *4*) Athlete info represents whether the athlete is long-distance or short-distance specialist, age, and other parameters that can be decisive in a similarity function.

To be able to revise and retain new knowledge and experience from present cases, it is important to have a rich case representation. In the article published by Smyth and Cunningham [53], they conclude that richer case representations tend to perform better with lower prediction error.

Different case bases can interact with the same system by implementing data collection requirements. We implement requirements in our work that deals with gender, age groups,

[1] https://www.draw.io

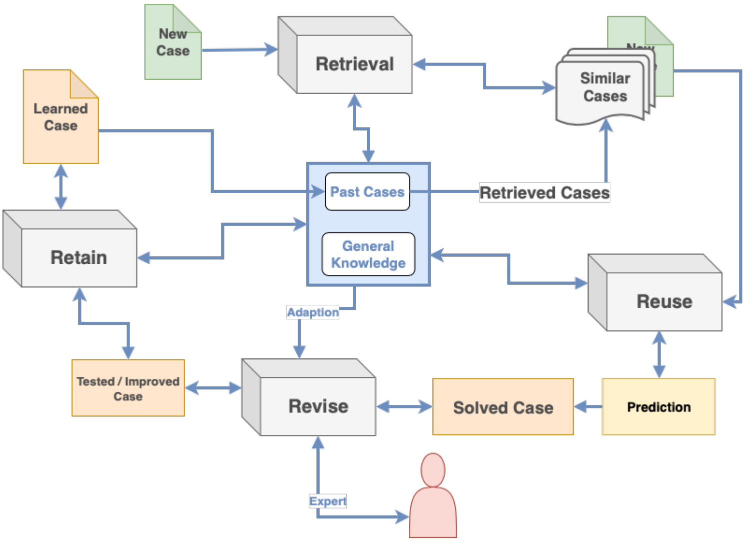


Figure 4.1: Intentional System Design

and type of competition (only world cup). Specifications are necessary in more comprehensive systems with extended case representations, including various type of cases. In our data collection phase we define requirements, and collected data that maintained the requirements. When reusing, some changes can occur in requirements following norms or stakeholders evolutions.

4.1.2 Retrieval and Reuse

The *retrieval* and *reuse* phase is the foundation of the CBR system, and where we collect and adapt the similar cases for the prediction. Figure 4.3 illustrates the full *retrieval* and *reuse* phase, including general *revise* and *retain*. When a new case comes in the system, the retrieval step looks for matching cases. If the distance between the new and old prob-

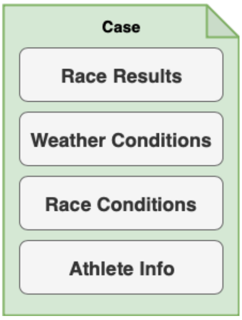


Figure 4.2: Initial Case Representation

lem is too distant, and no matching cases, a manually solving process occur to determine if the case can be solved and retrained to the case base. If there are matching cases, we reuse the retrieved cases and adapt the knowledge to calculate new predicted finish-times. We revise whether the result provides additional knowledge to the knowledge base and if yes, add to the case base. Typically will the revising phase be a manually extended process where coaches and athletes evaluate the actual results and the recommended finish-time. When there are additional knowledge the system retain the new revised case to the case base.

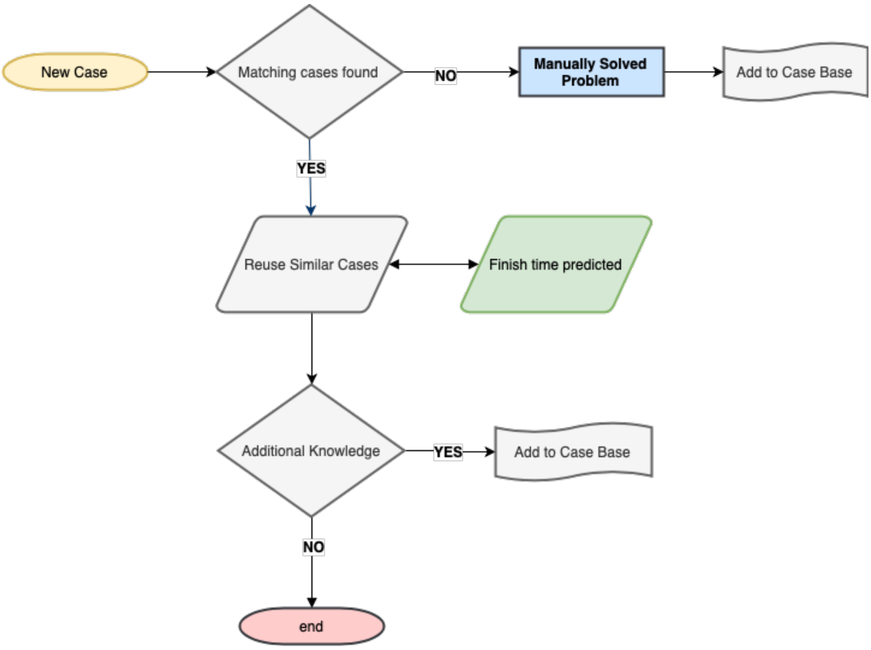


Figure 4.3: Flowchart of retrieval and reuse phase

4.1.3 Revise and Retain

There are various techniques of revising and retaining a solved case in a CBR system. In an extended six step CBR Cycle [43] will the retain phase include additional *review* and *restore* steps building an maintenance phase. Figure 4.7 shows the decomposition of CBR, where the revise and retain step has extended with inspiration from the six-step cycle and maintenance steps to fit our research problem. We are revising, in our intentional system, including measuring the solved cases with speed skating knowledge, such as statistics and actual results. The revise phase includes domain knowledge, interaction with experts, athletes, and traditional model evaluation. An additional step in the *retain* phase allows user and model to select which of the tested cases to retain. Our intentional CBR system improves by achieving richer case representations and experiences from several, including components add valuable knowledge.

4.2 Implemented System Design

After reviewing an intentional system will this chapter focus on the implementation and the components we have designed for the research purposes. Figure 4.4 explains the implemented system design. We focus on *retrieving* the most similar cases, and *reusing* them for prediction purposes.

When a new case enters the system, the retrieval phase based on similarity metrics compute the most similar cases. The retrieval phase is modeled with *myCBR Workbench* and uses the knowledge available in the model to find similar problems. The cases base is created before the retrieval phase and includes knowledge and past cases. Retrieval communicates with the case base to access the cases needed. The most similar cases and the new case are gathered and sent to the reuse phase, also mentioned as the adaptation process.

The adaptation process uses knowledge from the case base as well as similar cases to solve the new problem. The system implements two strategies (*median* and *mean*) for solving the new problem using the retrieved cases and domain knowledge. The case base is highly connected with all processes in the CBR system, and makes the case structure and knowledge model important.

All cases are solved, but in some cases, greater than others. There are systems where the new problem potentially will not find a similar problem, and the system is unable to solve the problem. This system store all solved cases but not retain them back to the case base. A retained solved case would represent a hypothetical result and therefore not a suitable solution to retain to the case base, before implementing a revise phase.

4.2.1 Case Structure

We are focusing on three types of features in this implementation. Figure 4.5 illustrates a categorized view of the features and an example from the case base. We include three race results and one external condition and race condition. The case features created are evaluated by the domain knowledge and modeled in **myCBR**.

1) Race results that include personal bests, season best and other non-personal best times. *2*) External conditions for all races where we focus on altitude. *3*) Race conditions for all races in the case, such as division.

We are explaining all case features used in the implementation chapter (Ch 5).

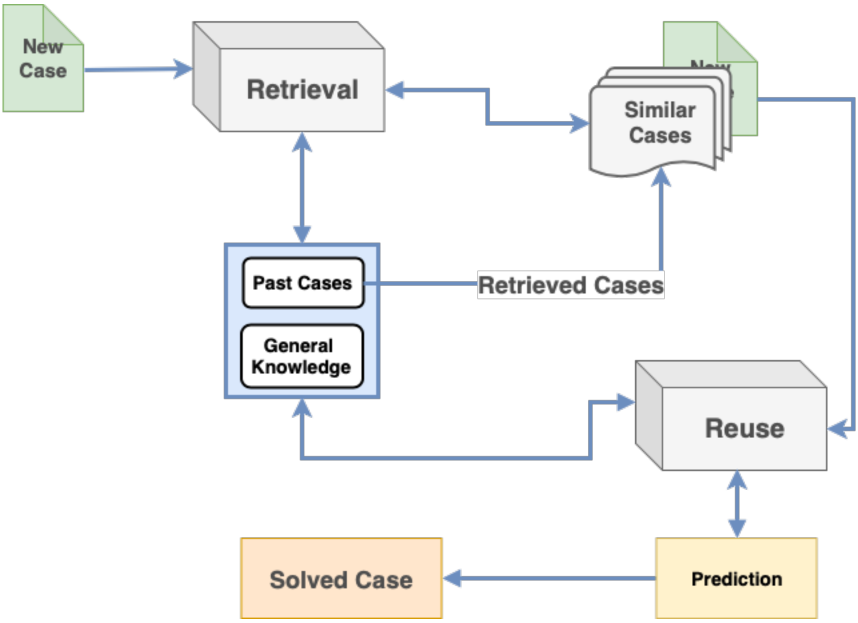


Figure 4.4: Implemented CBR System Design

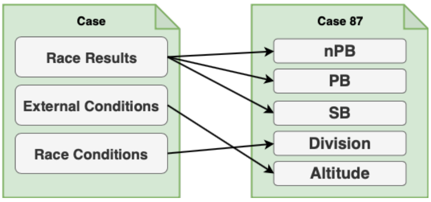


Figure 4.5: Implemented Case Representation with mapping to example case

4.2.2 Retrieval

Figure 4.8 shows the task decomposition for the implemented retrieval and reuse phases. In *retrieval*, the model goes through four tasks, identifies features, search, initial match, and select. First, the retrieval identifies features, then searches for domain knowledge and after finding necessary knowledge the initial match step calculate similarity for the features. Lastly, it selects the most similar cases based on predefined criteria.

We adopt the cases and improve the finish-time based on race conditions. However, do not revise or retain improved solutions. The main objective of further work is to implement the remaining steps in the CBR cycle. All cases retrieved includes a various number of races.

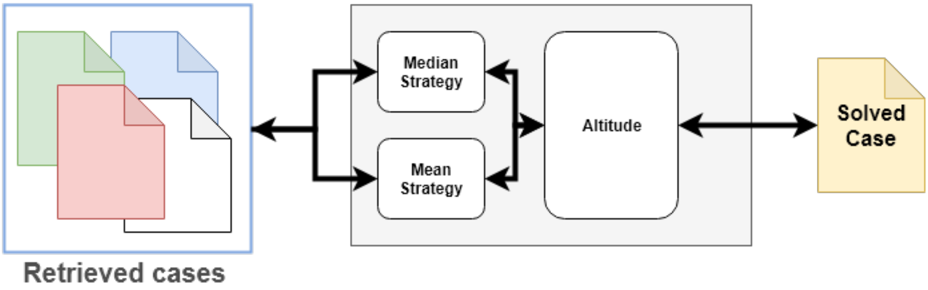


Figure 4.6: Adaptation Process

4.2.3 Reuse

This is the most critical process in the system, where the retrieval phase is necessary for the reuse process to exist. The goal is to modify cases (i.e., race) and predicting new best possible finish-times to fit the query case. The reuse step is an essential step for the prediction. The adaptation process includes domain knowledge (i.e., altitude correction and division), rules, and query case. In general will the process contain three steps, *1*) filter the retrieved cases based on similarity *2*) calculate new finish-time using *mean* and *median* strategy *3*) correct the results based on domain knowledge and query altitude. The full adaptation step is illustrated in Figure 4.6. The grey box illustrated the adaptation process where the output is a new case, including the new best possible finish-time, together with the query features.

We are implementing one dimension of the reuse phase. *Transformational* reuse, because we are using the knowledge related to the new cases for prediction. In an intentional CBR system would the second dimension, derivational reuse, supply ability to adapt past solved cases instead of only new cases, including results as the system does.

An extension of these processes will be presented in Chapter 5. The result of the predicted finish-times is used as a recommendation, and in future work be retained to the case base.

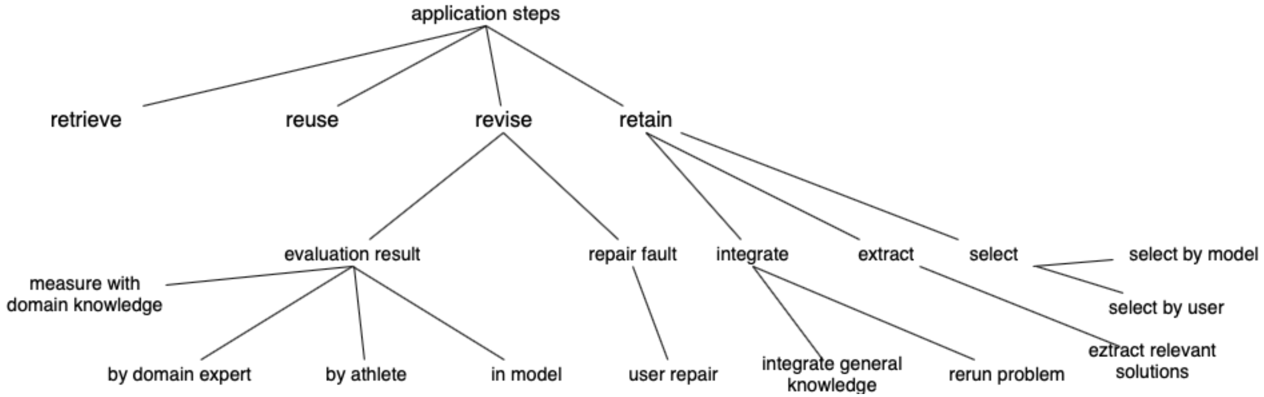


Figure 4.7: Task decomposition for revise and retain (adapted from [2; 43])

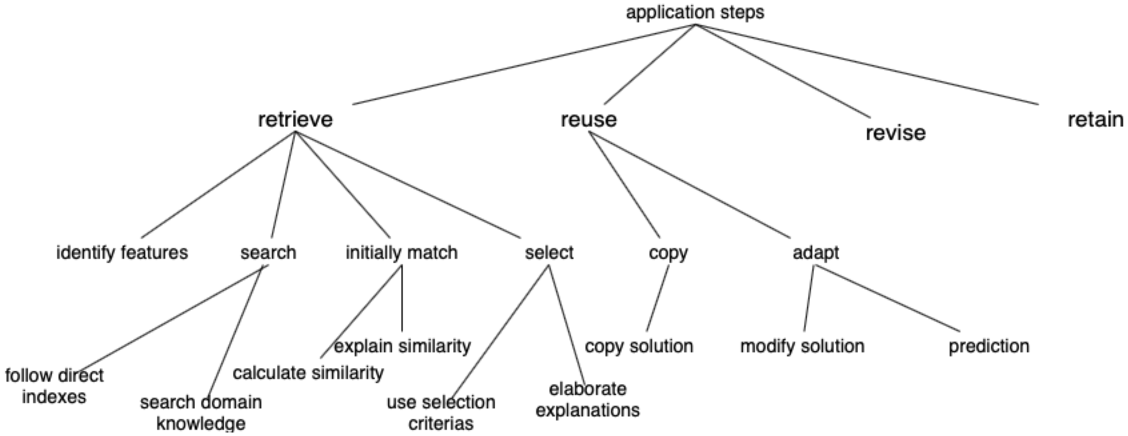


Figure 4.8: Task-method decomposition of Retrieve and Reuse in CBR (adapted from [2])

Chapter 5

Implementation

We will, in the following chapter, describe the implementation of the CBR system from Chapter 4. The implementation will contain data collection phase, modeling phase with *myCBR*, retrievals, and at last adaptation process.

5.1 Data Collection

5.1.1 Data set

To be able to achieve an accurate and representative prediction, a substantial component is to create a valid and correct data set. The data set contains features representing essential parameters in speed skating, as described in Chapter 3.2, and these features are later cases in the case base. We will focus on implementing the design from the previous chapter, where we scaled the intentional CBR system and prioritized the retrieval and reuse phase.

We collected all race results and race conditions from the official result site for speed skating, *www.speedskatingresults.com* [28]. This subsection will focus on collecting data, and an extended description of the cases and case representation is presented later in this chapter. We use Table 3.1 to visualize the altitude of the different ice rinks used in the data set. There are a couple of requirements that must be met for the data to be valid and as similar as possible in our data set. All results are from the same kind of race, women's 3000 m. Women tend to pace themselves better than men [11; 57], and a 3000 m race last for approximately four minutes, which makes it an attractive race to start our predicting with. Shorter distances cause the results to be concentrated over a minor difference in time and set complex accuracy requirements to the system. We also added a requirement where the race had to be a World Cup race to be valid in the data set. World Cup races are races where the athletes statistically perform well [31]. Athletes with more than three races in the case base were separated from the athletes with less than three races. This non-numerical parameter allows us to sort the cases were the ones with multiple races can give more accurate guideline about physical shape and variations in results from the past.

BASIC CASE REPRESENTATION	
float	nPB
float	PB
symbol	Division
int	nPBMasl
int	PBMasl

Table 5.1: Basic Case representation

The cases with few races gives us an incomplete insight into the athletes' physiological health and competition schedule. Specific data collection gives us better cases with seemingly few parameters.

The data set for this implementation contains 169 cases, where each case varies from five to eleven features. Table 5.1 and 5.2 illustrates the various case representations.

```
c1(float nPB, float PB, symbol division ,int
          nPBMasl ,int PBMasl)
```

Each case consists of at least one **nPB** race which stands for a 3000 m race that is not a personal best time. **PB** races stands for the athlete's best 3000 m time. **Division** represent which division the **nPB** race was performed in. Dissemination on divisions is written in Chapter 3.2.2. **Masl** in both **nPB**, **SB** and **PB** represents the altitude of the ice rink where the race was preformed.

For those athletes who have more races to analyze, we created an extended case representation that contains multiple features that can make a variation in the prediction results. In Smyth and Cunningham's marathon article they concluded richer representation did not always produce better results. We will also investigate the hypothesis if an extended case representation affects the results and gives us a more accurate prediction. Table 5.2 shows the features from initial case representation (Figure 4.4) we include in our implementation.

We converted all acquired times from *MM:SS: HH* to *SS: HH* to get a more straightforward numerical comparison, and more accessible for the *myCBR* API to handle correctly.

5.1.2 Case Representation

Next step in the implementation is to structure the cases efficiently. We analyzed essential factors in speed skating and translated them into case features. One can find background research on the case features in Chapter 2.2. The list below contains all case features with an explanation of how they contribute.

1. **nPB** The foundation of the prediction is to take a not personal best race and compare with similar speed skaters who have made a similar non-best race in the same race conditions. We use the weighted difference between nPB and PB to predict how close to personal best time the query skater possible can achieve with nPB race

FULL CASE REPRESENTATION	
float	nPB
float	PB
symbol	Division
int	nPBMasl
int	PBMasl
float	NewestRace
float	NewestRaceMasl
symbol	SkaterType
bool	AllRaces
float	SB
int	SBMasl

Table 5.2: Case Representation with all features included

conditions. We included nPB as a case feature because most speed skaters have several races that are not best-time but interesting as a solution for upcoming races in the same conditions. We want to many different cases (problems) as possible so that we can find a suitable solution (similar races).

2. **PB** To fulfill the comparison between nPB and PB, is it relevant to gather information on the speed skater personal best time and race conditions. Performing a PB will not happen often during a season, and express the best possible time in optimal conditions. Our work will not predict a new best-time except if the query runner has the same conditions as for the PB. We included PB because we want to find the difference between a regular race and the best possible race. We will also include PB in the prediction accuracy analysis.

3. **SB** As mention, tend old nPB races to give a not representative illustration of the speed skater. Seasonal Best stands for the best time from a specific season. We are using seasonal bests from 2017/2018. Including SB will supplement races to a more extended case representation, and also pinpoint what the speed skater achieved last season. In most cases will SB be close to PB, yet more representative since it is a reasonably new race.

4. **Division** As written in Chapter 3.2.2, is it a difference in the execution of the race when starting in division A or B. Division B starts in a quartet start, and the speed skaters can get aerodynamic benefits when lying close the speed skater in front. However, possibly disadvantageous in not having an open lane for a clean and incident free run. We wanted to incorporate as many races as possible, and therefore categorize the cases concerning division. This feature is one of the non-numerical features in the case representation.

5. **nPBMasl, PBMasl, NewestRaceMasl & SBMasl** (Meters above sea level of the rink) For the most accurate and well-defined prediction, it is essential to find out where the race took place and how the barometric pressure was at the current rink.

It was mention that altitude decreases the times by $3.2\%(\pm 0.5)$ for senior skaters per 1000m increase in altitude [31]. Excluding the altitude conditions will give an inconstant prediction. Since we require similar race conditions on the races compared, is altitude essential to include. The altitude is the most critical, which makes the most difference in the predictions. Chapter 3.2.1 explains the advantages of high-altitude racing.

6. **NewestRace** Athletes who have competed in multiple races have the opportunity to supplement the newest race as a case feature. The newest race signifies the result of the latest race, and thus also how the performance curve of that athlete develops. We will in our experiment only include this feature for the athletes competing in the World Allround Championships and later in the World Cups Finals. The two competitions are held in the same altitude conditions and will give good indications for the future results. We have to emphasize that this feature will be more accurate for speed skaters who frequently compete, and less essential for them too with few competitions during a season. We included this feature because we wanted to investigate the outcome of excluding the not representative race results, such as last years results.

7. **SkaterType** Characterize speed skater is an essential part of a system like this. We wanted to categorize the speed skaters into sprinters and long-distance for pacing purposes. Sprinters tend to race with an aggressive strategy where the speed decreases towards the end of the race and long-distance increase the pace towards the end after a more passive start. In long-distance is the aerobic endurance essential, but in a sprint is the anaerobic power more decisive. We categorize the speed skaters from the ratio between 500 m and 3000 m races. Sprinters will hold an average lower ratio then long-distance speed skaters. This feature is an non-numerical parameter.

8. **AllRaces** The prediction results can vary when including nPB races achieved in a short interval and races spread over a more extended period of time. We wanted to implement a boolean feature representing whether the speed skater has been competing in three world cup competitions (World Cup 3, 4 and 5) in 2017/2018. We can specify the retrieval for athletes with the corresponding competition schedule and exclude cases that can mislead the predicted finish-times.

In the experiment we will use some of the features. We will, as described in the thesis structure, only implement elements of the design and test elements of the implementation. See figure 1.1 for visualization of the structure. We will discuss the consequences of including case features in Chapter 7.

5.2 Modeling with *myCBR Workbench*

After defining the case features explained in the previous chapter, we began modeling the case base and similarity measures with *myCBR*[1]. *myCBR* is an open source Case-Base Reasoning developing tool distributed by NTNU [45]. We used two different components

[1] http://mycbr-project.net/index.html

of the software, *myCBR Workbench*, and a software development kit (SDK). We will analyze and discuss the possibilities and disadvantages of the software and address how the software helped us develop and retrieve. Our motive for using *myCBR* was to have a graphical user interface for modeling purposes where we wanted to focus on system design and research than developing new CBR application software. *myCBR* provides a straight forward approach for building small CBR applications, and a SDK for integrating with our code which suited our situation ideally.

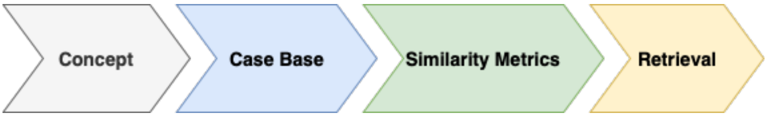


Figure 5.1: Modeling flow in myCBR

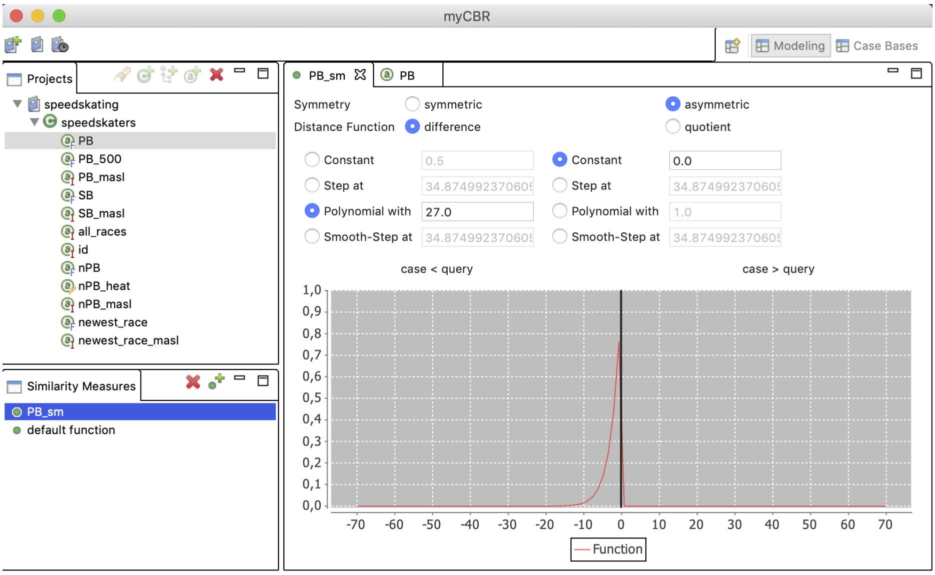
Figure 5.1 illustrate the key components in the workflow for modeling a CBR System in *myCBR workbench*. We developed the knowledge model in four steps before the retrieval of similar cases with the SDK. In short, first, create a concept for the knowledge model. For the case structure we imported the data set and *myCBR Workbench* defined the case features automatically. We then elaborated similarity measures on feature level (local similarity), and concept level (global similarity). We used Euclidean Distance for finding similar cases and weighted the features equally in the amalgamation function. The amalgamation function ties all features together based on the weights given [13]. Other opportunities for amalgamation functions are weighted sum, min and max. Finally, optimize the knowledge model using the retrieval module. Figure 5.3 describes the relation between the components in *myCBR* and one can see that the *myCBR Workbench* is the foundation in the CBR system. Simple testing by the Retrieval Engine and modelling the explanation knowledge helps us retrieve the cases with proper similarity.

5.3 Retrieving with *myCBR SDK*

We modeled the CBR system completely in *myCBR Workbench*, whereas we had to develop a Python environment for the adaptation process. The *myCBR* SDK allows extended applications to interact with the knowledge model, and continuously update the CBR system. Figure 5.3 describes the interaction between the applications distributed by *myCBR* and external applications [47]. The SDKs essential tasks are to load and control the project, handling the POST queries from the external application, retrieve cases from the project, and load the case bases. We cloned a RESTful API project from GitHub[2] created by Kerstin Bach that inserts *myCBR* into Spring.io and provides retrieval through HTTP REST calls and Swagger API.

The retrieval response from the API comes as a *json* object, shown in Figure 5.4, with the similar cases ordered by similarity, where the query case has a similarity of 1.0. We

[2] https://github.com/amardj/mycbr-rest-example



Symmetry ○ symmetric ● asymmetric

Distance Function ● difference ○ quotient

○ Constant	0.5	● Constant	0.0
○ Step at	34.87499237060	○ Step at	34.87499237060
● Polynomial with	27.0	○ Polynomial with	1.0
○ Smooth-Step at	34.87499237060	○ Smooth-Step at	34.87499237060

Figure 5.2: The modelling view *myCBR Workbench* with case features (left top), local similarity functions (bottom left) and the local similarity function for PB (middle)

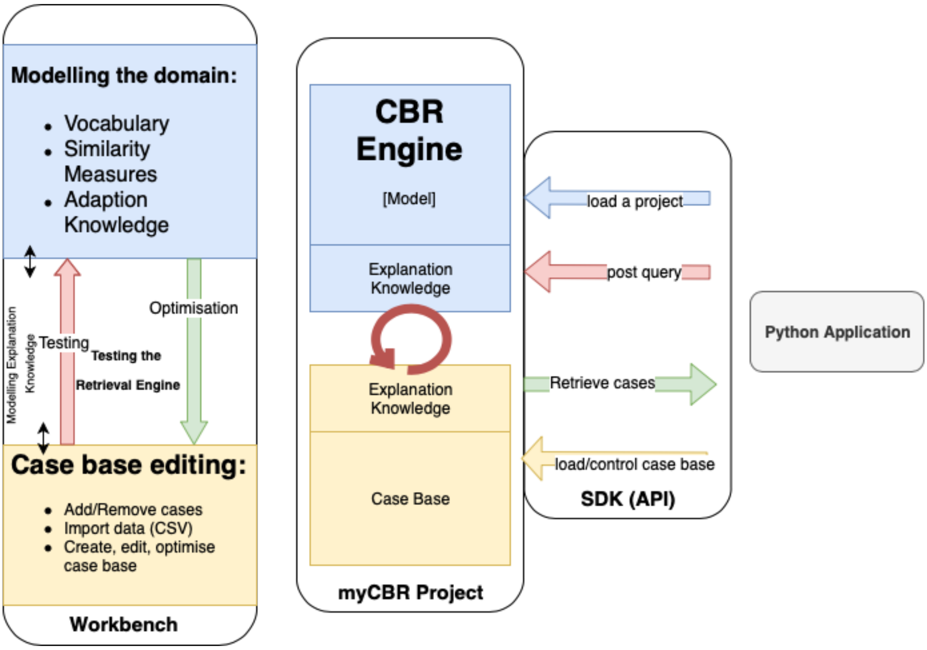
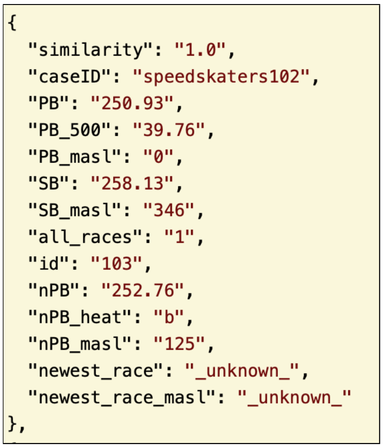


Figure 5.3: Interaction between *myCBR Workbench*, SDK and external applications [47]



```
{
  "similarity": "1.0",
  "caseID": "speedskaters102",
  "PB": "250.93",
  "PB_500": "39.76",
  "PB_masl": "0",
  "SB": "258.13",
  "SB_masl": "346",
  "all_races": "1",
  "id": "103",
  "nPB": "252.76",
  "nPB_heat": "b",
  "nPB_masl": "125",
  "newest_race": "_unknown_",
  "newest_race_masl": "_unknown_"
},
```

Figure 5.4: JSON response for query case

used the *pandas*[3] library for preparing the retrieved cases for prediction algorithms. Later in this chapter will a real-time example explain the complete process and handling of the retrieved cases. If a case does not include a specific race, the attribute is marked with the value **`_unknown_`** as seen in Figure 5.4.

5.4 Similarity Measures

Similarity functions define the similarity between two cases. Similarity measures are a necessary and decisive component of the CBR system. In this system, we are defining attribute-value pairs with the local-global principle. When defining a proper similarity measure for each feature, it is essential to analyze the distribution of data stored in the case for the specific feature.

We used a boxplot function on each feature and found the interquartile range (IQR) to see the data distribution. A boxplot function is a standardized procedure of visualizing the division of the data set categorized in five different steps. The minimum, first quartile, median, third quartile, and maximum [20]. The boxplot function used as inspiration to develop the similarity measures in this thesis is simple with a focus on the rectangle that covers the first quartile to the third quartile, called the IQR. It is favored to use the IQR instead of median and average when defining similarity measures because most values are allocated there. Equation 5.1 shows the IQR formula:

$$IQR = Q3 - Q1 \qquad (5.1)$$

Figure 5.5 illustrates the boxplot graph for nPB, PB and SB. The coloured squares represent the IQR where the mid 50% of the distribution is allocated. The small circles are outliers, and the lines are the top and bottom quartile. Figure 5.6 shows the similarity func-

[3] `https://pandas.pydata.org`

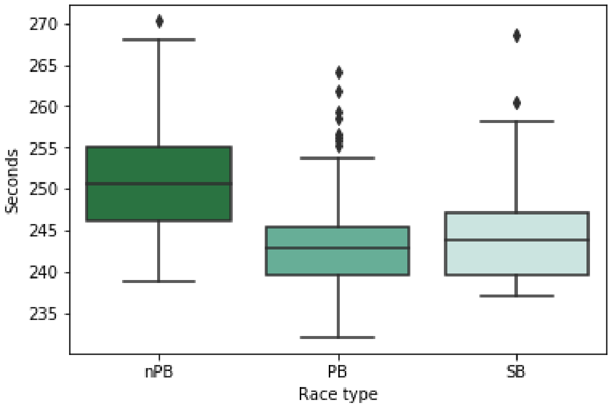


Figure 5.5: Boxplot for PB, nPB and SB

tion for the nPB, PB and SB case feature. We created an asymmetric similarity function for all races. Where $cases < query$ is a polynomial function, fitting the curve including the IQR rate within the 0.05 in similarity. Where $case > query$ is the similarity 0.0 because we only want to retrieve cases faster than the query for more optimistic predictions.

In this work, there are non-numeric features with two outcomes where a simple symbolic similarity function fits perfectly. Figure 5.7 explains the two outcomes, where the similarity for same division equals 1.0 and different divisions equals 0.0. An improvement is to tune this case feature so that we can improve similarity when comparing cases with a different division. We want to retrieve and reuse cases from similar divisions.

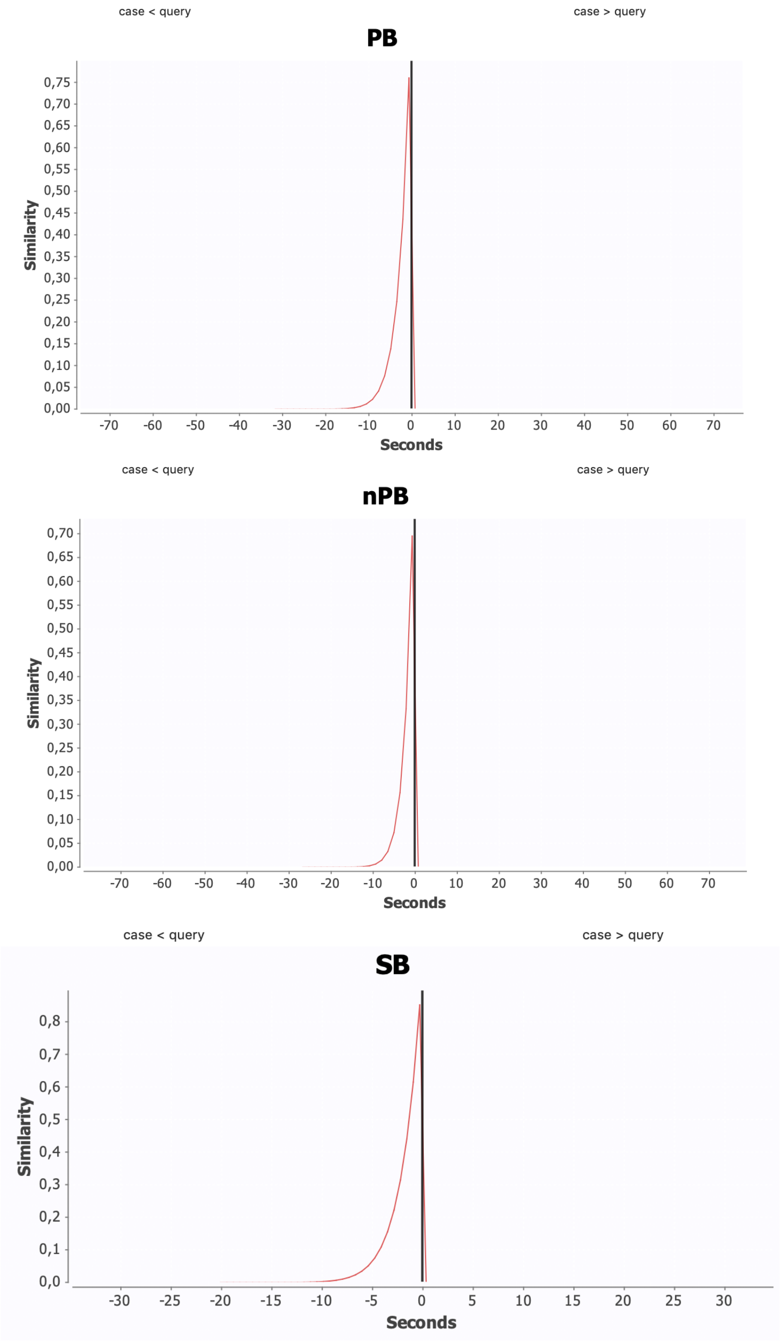
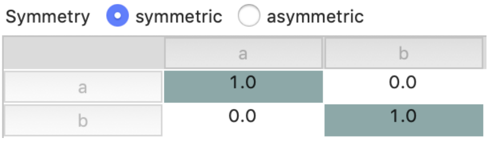


Figure 5.6: Similarity Measure function for PB, nPB and SB from textitmyCBR Workbench



Symmetry ● symmetric ○ asymmetric

	a	b
a	1.0	0.0
b	0.0	1.0

Figure 5.7: Symbolic local similarity function for Division in *myCBR Workbench*

5.5 Validation Methods

There will be two different validation strategies used for this work. The first one is a standard k-fold Cross Validation and is based on the principle one leave out the k number of cases from the training set. The second one will be based on the same k-fold principle and leave one "test case" out and iteratively train the data set on the rest of the cases in the case base.

5.5.1 k-Fold Cross-Validation

In this chapter, the basics of k-fold cross-validation will be explained. K-fold cross-validation is an improvement of the holdout method. The holdout method is the simplest cross-validation method where the data set is divided into one training set and one test set [48].

Cross-validation is used to evaluate machine learning models and based on the concept that one leaves k instances of the training set and use for testing [10], and in this thesis, k will stand for the number of cases left out for testing. Cross-Validation is mostly a technique for estimating the quality of a machine learning model. Cross-Validation testing is performed to determine how the model works on data not included in the data set used for training the model. See Figure 5.8 for graphic representation. There are many different variations of Cross-Validation, wherein this task, the type Train / Test Split will be applied. Train/Test split extracts only one k so that the data set splits into two parts, one for training and one for testing. 10% from the training set will be left out from the model and use for testing. Instead of an iterative process where one takes out a part, the model trains with remaining parts and tests with the left out part and repeats this process with several different parts in this thesis the model will only be tested with 10% of the data set, once.

k-fold Cross-Validation is a technique that is easy to use, and also easy to understand. Generally, this model ends up with less encouraging predictions of the model than other evaluation techniques. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

5.5.2 Leave-one-out Cross Validation

Leave-one-out Cross Validation (LOOCV) is a variant of k-fold cross-validation, which was explained later in Chapter 5.5.1. In k-fold is the data divided into k parts, but in

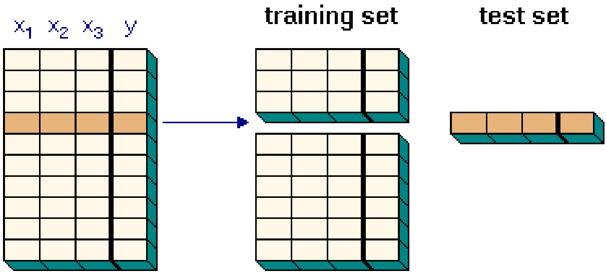


Figure 5.8: Model describing the Cross-Validation principle [19]

LOOCV is the data divided into $k = n$ parts. Which implies that the number of parts equals to the number of observations [12; 35; 19]. This type of cross-validation fits perfectly for small data sets and maximize the training of the data set to all cases instead of some pieces.

In LOOCV is the left out case used as validation data and query in the retrieval, and the remaining cases in the data set used as training data. LOOCV is an expensive process because one will have to do the retrieval k number of times.

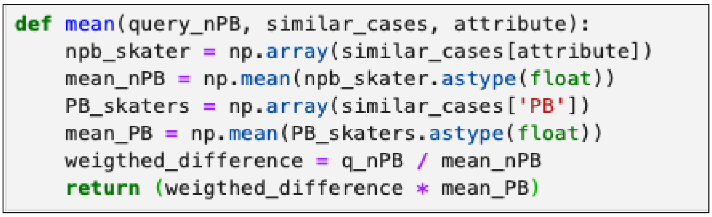
5.6 Adaptation process and prediction strategies

The prediction will try to find the best reasonable finish-time for a query speed skater based on the relationship between finish-times nPBs, SBs, and PBs. We will use the external race conditions from the query nPB as the race condition for the predicted race. In Equation 5.4 is a query case shown. In the process of calculating the predicted finish time will we use two different approaches, mean and median. Then an algorithm where we compensate for the differences in race conditions. The purpose of using two different approaches is to see which one that gives the most realistic result.

5.6.1 Mean Strategy

We base our first approach on an average of all the retrieved $nPBs$. As shown in Equation 5.2 will nPB from query q and *mean* from retrieved cases C_i constitute a weighted difference that is multiplied with the mean of the PBs of C_i. The mean approach will vary in quality. The result is not a real case from the data set, but a common result. However, a mean strategy will provide a more relatable result for a small data set than the median strategy. Figure 5.9 shows an implementation of the equation in Python.

$$P_{mean}(q, C) = \frac{q(nPB) \cdot \sum_{\forall i \in 1..k}(C_i(nPB)) \cdot C_i(PB)}{k} \tag{5.2}$$



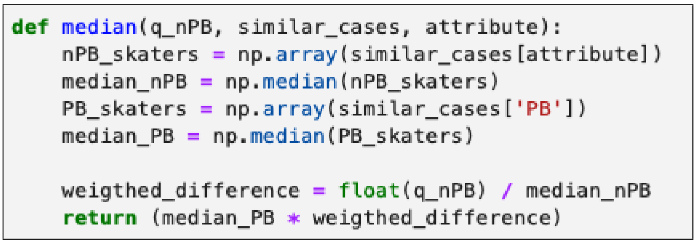
```python
def mean(query_nPB, similar_cases, attribute):
    npb_skater = np.array(similar_cases[attribute])
    mean_nPB = np.mean(npb_skater.astype(float))
    PB_skaters = np.array(similar_cases['PB'])
    mean_PB = np.mean(PB_skaters.astype(float))
    weigthed_difference = q_nPB / mean_nPB
    return (weigthed_difference * mean_PB)
```

Figure 5.9: Mean Strategy

5.6.2 Median Strategy

The *median* strategy, shown in Equation 5.3, focus on the median of the retrieved cases C_i and estimates the best possible time for q by multiplying a difference based on query nPB and the median of the nPB from the retrieved cases C_i. In consultations with a domain expert will this approach produce a more accurate representation of a speed skating race as it will automatically take a realistic race into accounts, which includes mistakes and accidents. What strengthens the approach in advance is that the median strategy always will use real cases, as the mean approach creates new race results based on retrieved cases. Figure 5.10 shows an implementation of the equation in Python.

$$P_{median}(q, C) = \frac{q(nPB) \cdot \forall i \in 1..k \; C_i(PB)}{\underset{\forall i \in 1..k}{\text{med}} \{C_i(nPB)\}} \tag{5.3}$$



```python
def median(q_nPB, similar_cases, attribute):
    nPB_skaters = np.array(similar_cases[attribute])
    median_nPB = np.median(nPB_skaters)
    PB_skaters = np.array(similar_cases['PB'])
    median_PB = np.median(PB_skaters)

    weigthed_difference = float(q_nPB) / median_nPB
    return (median_PB * weigthed_difference)
```

Figure 5.10: Median Strategy

5.6.3 Calculating with Altitude

There are two different ways of performing an accurate prediction, including altitude as a feature in the global similarity function or adjusting the predicted times with the accurate altitude and thus assuming that all races are on the same altitude in the retrieval. We will in our experiment, adjust the predicted finish times after retrieval and *PBMasl, SBMasl and nPBMasl* will be weighted as 0.0 in the amalgamation function. As mentioned earlier, there is a significant difference in results on races in rinks located in high altitude climate, and thus, barometric pressure is the most critical feature in the knowledge model. To be able to adjust the prediction to the exact height difference, we need to make some qualified assumptions.

First, we assume that the barometric pressure increases by *1 hPa per 8 meters in altitude gain*. These assumptions are essential for further computation, and we have discussed with a domain expert in the speed skating society. Nevertheless, will these assumptions evoke error but also help us experimenting.

The next qualified assumption is where we expect that barometric pressure makes a difference in speed skating up to 0.6 hundredths of a second per hPa per round (400 m). We need to apply this qualified assumption to be able to correct the predictions. Experts state a standard assumption is a conversion from races in high altitude to low altitude with approximately 0.8-time points. One time point equals one second each 500 m. 0.8-time points will then correspond to about 0.8 seconds of 500 m. Calgary Olympic Oval is located 1105 meters above sea level, while Vikingskipet (Hamar) is located 125 meters above sea level, where there is a difference of 980 m, corresponding to approximately 122 hPa. A calculation of how we reach approximately 0.8-time points shown below.

Altitude differences for Calgary and Hamar

$$1105m - 125m = 980m$$

Corresponding hPa (1 hPa = 8 meters)

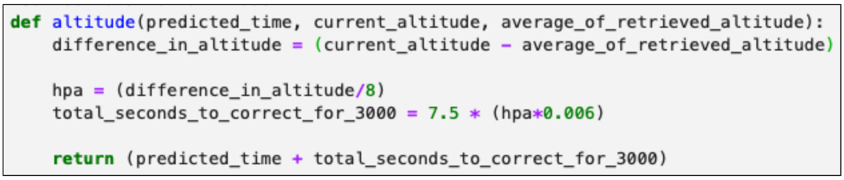
$$980m/8m = 122 \text{ hPa}$$

How many seconds per round does it correspond to?

$$122 \text{ hPa} * 0.006 \text{ seconds} = 0.732 \text{ seconds per round (400m)}$$

We expect about $+0.8$ time points per round when we calculate from rinks in the high altitude climate to the lower altitude climate. As in the example, finding the difference from Calgary to Hamar and notice that there will be a time supplement of 0.732 seconds per round. On a 3000 m race, the adjustment from the two rinks be 7.5 rounds $* 0.732 = 5.49$ seconds.

Looking at the data set, we can ascertain that most PBs come from high altitude, either Salt Lake City (1423m) or Calgary (1105m), and gives us an average altitude at 1214 MASL. In average is the difference between PB and nPB in our data set 7.09 seconds, and with a significant part of the PBs performed in high altitude will an additional 5.49 seconds comparing low altitude and high altitude be a reasonable assumption. Figure 5.11 shows an implementation of the assumption.



```
def altitude(predicted_time, current_altitude, average_of_retrieved_altitude):
    difference_in_altitude = (current_altitude - average_of_retrieved_altitude)

    hpa = (difference_in_altitude/8)
    total_seconds_to_correct_for_3000 = 7.5 * (hpa*0.006)

    return (predicted_time + total_seconds_to_correct_for_3000)
```

Figure 5.11: Altitude function for correction of predicted finish-time

5.7 Prediction Experiments

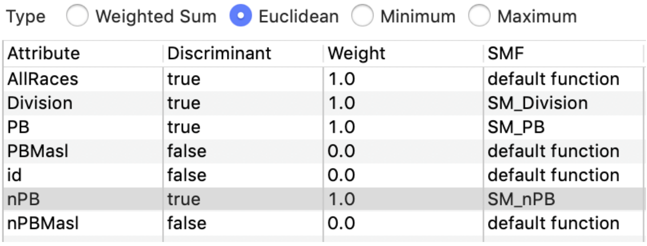
After describing the components to implement, the next step is to test essential parts of the CBR system. We divided the experiment into three parts, where each part focuses on new case features, which can affect the results and prediction error. In the next chapter will the results of the experiments be explained and analyzed.

5.7.1 Prediction based on nPB

In the article by Smyth and Cunningham [52] explained in related research (Ch 2.2), they use an approach with the runners nPB and PB as a foundation in the prediction. We will, in the first experiment, adapt the basic prediction approach and the use of multiple prediction strategies but change types of strategies and complement with non-numerical features.

$$q = (\texttt{nPB},\ \texttt{nPBMasl},\ \texttt{PB},\ \texttt{PBMasl},\ \texttt{Division}) \tag{5.4}$$

The features used in this retrieval are *nPB, PB, nPBMASL, PBMASL, Division*, as the query case show in Eq 5.4. All cases in the case base have a registered an nPB and PB race. Both races have an associated external race condition feature that represents the altitude at the current location. The global similarity in this step contains three local similarities. The local similarities are nPB, PB, and division for the nPB race. Figure 5.12 shows the global similarity function with Euclidean Distance, and amalgamation function sat with equal weights on all the included features.

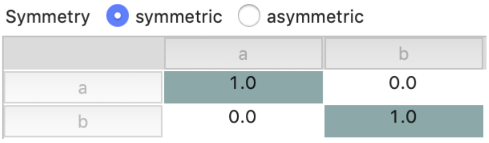


Type ○ Weighted Sum ● Euclidean ○ Minimum ○ Maximum

Attribute	Discriminant	Weight	SMF
AllRaces	true	1.0	default function
Division	true	1.0	SM_Division
PB	true	1.0	SM_PB
PBMasl	false	0.0	default function
id	false	0.0	default function
nPB	true	1.0	SM_nPB
nPBMasl	false	0.0	default function

Figure 5.12: Global Similarity for nPB based retrieval from *myCBR Workbench*

We want to find cases with an nPB similar and slightly faster because we predict ambiguous but realistic finish-times. PB is processed the same way, however hereabouts more essential to find cases with faster PB than the query cases because we need the difference between the two races from the retrieved cases, and apply the difference to the query case to predict a faster still possible finish-time. We created asymmetric similarities, shown in Figure 5.6, to handle the criteria of retrieving cases where $case < query$. The consequences of not modeling an asymmetric similarity measure could be to retrieve similar cases that are slower than the query case, and the prediction ends up to predict a slower finish time than already achieved. The third similarity function used in this retrieval is a symbolic similarity function describing the division for the query case at the nPB race. The results are different for speed skaters starting in Division A and B. Therefore, important only to retrieve cases with equal division, illustrated in Figure 5.13. The similarity function for the division is a simple function where the similarity equals 1.0 for corresponding divisions and 0.0 else.



Symmetry ● symmetric ○ asymmetric

	a	b
a	1.0	0.0
b	0.0	1.0

Figure 5.13: Similarity Measure function for Division in *myCBR Workbench*

After receiving the most similar cases using the *myCBR SDK*, we run through the two predictions strategies, *Median* (Eq. 5.3) and *mean* (Eq. 5.2). Chapter 3.2.1 explains why altitude causes variations in performance, and Chapter 5.6.3 explains how to compensate for the variations in the predictions. Since we do not include the race conditions in the retrieval, we need to correct the predicted times we receive from the *mean* and *median* strategy. In a future implementation of the CBR system is it essential to include the race conditions in the global similarity function, but to illustrate an example of the implemented system we only use the domain expert assumptions described in Chapter 5.6.3. An average of the altitude from the retrieved cases and the nPB altitude from the query case is used to find the difference that will be corrected the finish-time. The difference in altitude is then used to find the corresponding hectopascal (hPa). We find the total seconds by multiply the hPa found with 0.006 seconds and 7.5, which is the number of rounds on a 3000 m race. Figure 5.11 shows the calculation programmatically.

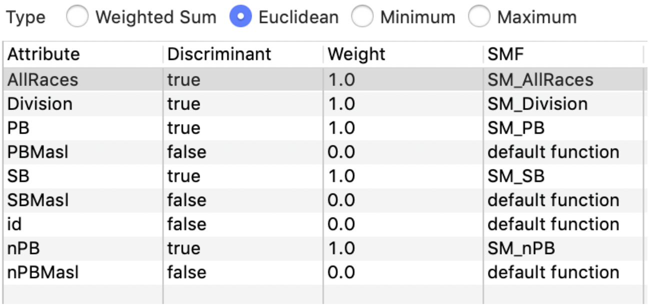
After reusing the retrieved cases in the two prediction strategies and adjust the altitude differences, the results represent the best possible finish-time in the race conditions of the query nPB race.

5.7.2 Prediction based on SB

The second prediction experiment is based on the speed skaters best time last seasons (SB). Equation 5.5 illustrates the query case representation.

$$q = (\texttt{nPB},\ \texttt{nPBMasl},\ \texttt{PB},\ \texttt{PBMasl},\ \texttt{SB},\ \texttt{SBMasl},\ \texttt{Division}) \tag{5.5}$$

This prediction plan appeared up when collecting the newest time for all speed skaters, which turned out to be a more complex task than expected. We want the cases to include results that are up to date and represents the athletes' physiological shape, which assumingly give a more accurate result. SB indicates how fast the speed skater performed last season and therefore a qualified indication of how the speed skater perform the next season. What differs SB and nPB is that nPB can provide a false image if the results are outdated, where SB provides a more current image of the speed skater. However, nPB contribute to the case base with numerous races, and essential for a detailed knowledge base. SB will also hypothetically provide a more stable prediction since the race is the optimal race result from a series of races the previous season where an nPB is simply one single race that could include mistakes and physiological conditions on the athlete which can affect negatively on the result. Figure 5.14 presents the global similarity function for the SB retrieval with the same structure as the first nPB experiment.



Type ○ Weighted Sum ● Euclidean ○ Minimum ○ Maximum

Attribute	Discriminant	Weight	SMF
AllRaces	true	1.0	SM_AllRaces
Division	true	1.0	SM_Division
PB	true	1.0	SM_PB
PBMasl	false	0.0	default function
SB	true	1.0	SM_SB
SBMasl	false	0.0	default function
id	false	0.0	default function
nPB	true	1.0	SM_nPB
nPBMasl	false	0.0	default function

Figure 5.14: Global Similarity for SB based retrieval from *myCBR Workbench*

The procedure is equivalent to the nPB experiment. However, the global similarity function is modified, see Figure 5.14. We add the case features SB and $SBMasl$ to the global similarity function. The case features are *SB, PB, SBMasl, PBMasl, nPB, nPBMasl, Division, AllRaces*. The case base is smaller with fewer cases available in this experiment since not all of the cases could be related to an SB race, and many cases evolve from the same speed skater only another nPB race. The retrieved cases that not include an SB race will be filtered away using *pandas* DataFrame[4]. The most similar cases are reused with the same *median* and *mean* strategies and altitude correction as the nPB experiment. In the nPB experiment, the average of nPBMasl is used in the altitude calculation, but in this experiment, the average of retrieved SBMasl and query SBMasl determine the difference.

[4] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

5.7.3 Prediction for World Cup Finals 2019 based on newest time

Associating this work with reality is a significant highlight in the process, whereas we investigate if the CBR system behaves correctly. The results from the two previous experiments will provide invaluable knowledge of how the similarity measures behave to improve the model in further work. In this experiment, we used the results from the World Allround Championship[5] and predicted the results for the World Cup Finals[6]. Typically, we do not hold a test set that can evaluate a prediction error because we are predicting improved finish-times and not the real finish-times and this experiment gives a hands-on evaluation of the accuracy of the CBR system.

We have collected the results from the 3000 m race in World Allround Championship and added PB and SB to the case representation in the case base. The case base will only include the 23 participants from World Allround Championship, which is a disadvantage for the accuracy of the prediction. Important to mention that the World Championships was held the weekend before the World Cups Finals with approximately likewise conditions. The World Cups Finals in Salt Lake City has an altitude of 1423 MASL and 1105 MASL at the World Championships in Calgary. Also to be mentioned is that not everyone that participated in the World Cup Finals participated in the World Championships, and vice versa. Therefore will the main focus be to calculate the error on the average predicted finish times compared to the actual results in the Finals, instead of comparing prediction and results for each athlete individually.

$$q = (\texttt{WAC, WACMasl, PB, PBMasl, SB, SBMasl, Division}) \tag{5.6}$$

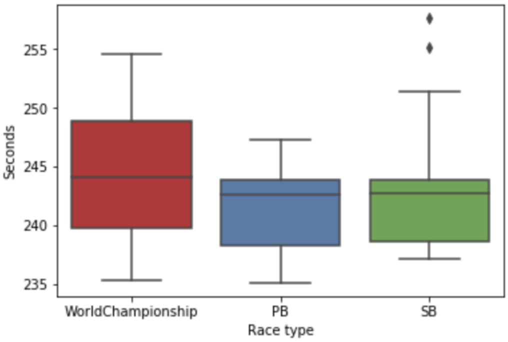


Figure 5.15: Boxplot for WAC, PB and SB

This experiment will use a similar approach as the nPB experiment (Chapter 5.7.1), where the World Allround Championship (WAC) race will represent the nPB feature. We included the SB race to extend the case representation and include the best time from the

[5] albertasport.ca/event/2019-isu-world-allround-speed-skating-championships

[6] https://live.isuresults.eu/events/2019_USA_0001/schedule

previous season. Equation 5.6 shows the query case representation. We use the same two strategies as the other experiments, and the correct the predicted times according to the altitude differences. In this experiment will the $nPBMasl$ always be 1105 as the World Allround Championship race condition and corrected to the World Cup Final at 1423 MASL.
We composed the similarity measures identical to for the other experiments. However, since we handle new data will the polynomial functions adjust to fit the 0.05 similarity rule used in the previous similarity functions, see Chapter 5.4. In the boxplot from Figure 5.15 we can observe the distribution of the races used in the experiment. The small case base will also indicate a smaller variation in distribution.

Chapter 6

Experiment

We structured our work to explain the system design, implementation of elements from the system, and test parts of the implementation. The following section will include a walk-thru example, and results of the three experiments explained in Chapter 5.7.1 - 5.7.3.

6.1 Example

We will, in this section, run through the entire process of a case retrieval and adaptation process. We experiment with nPB based case retrievals. Results come from real cases from the case base, as well as a query case. The test case is one of the cases excluded from the CBR system for experimentation. Query case has the following case representation:

$$q(\text{id}, \textit{nPB}, \textit{PB}, \textit{Division}, \textit{nPBMasl}, \textit{PBMasl})$$

And representing values:

$$q(71, 245.41, 238.39, a, 0, 1105, 242.76, 690)$$

Before starting the retrieval with *myCBR Workbench* and SDK, is it best practice to manually see which cases we determine as most similar, so we have an opinion about the retrieval result and accuracy of similarity measures. We evaluated the efficiency of the similarity measures this way. In the following example, expect cases within a few seconds from 245 in nPB be retrieved. Similarity for PB had less variety than nPB, and the similar cases are closer to the query case. We did retrieval using the SDK, and HTTP GET request through Swagger API and the python library *requests*[1]. Table 6.1 shows the ten most similar cases retrieved ordered in similar. We chose to retrieve ten cases because based on the results in Figure 6.2 reusing 10 cases present the most enduring prediction results.

[1] https://pypi.org/project/requests/

id	nPB	PB	div	nPBMasl	PBMasl	SB	SBMasl	Similarity
36	252.48	238.39	a	209	1105	-	-	0.8164
71	244.91	237.09	a	0	1105	237.09	1105	0.8098
12	245.36	242.75	a	1423	1105	-	-	0.8064
70	244.16	237.70	a	0	1423	239.47	1423	0.7890
69	244.00	237.70	a	0	1423	237.70	1423	0.7800
68	243.53	237.78	a	0	1105	237.58	1105	0.7683
73	245.54	238.01	a	0	1105	238.10	1105	0.7625
4	238.93	237.84	a	1105	1423	-	-	0.7423
118	245.91	237.84	a	214	1423	-	-	0.7421
34	248.85	237.78	a	209	1105	-	-	0.7353

Table 6.1: Retrieval result of 10 most similar cases. (div = division)

After having a look at the most similar cases, we start reusing and adapting without considering the external race conditions and consequences from altitude, which next step corrects. Before programming computations of *mean* and *median* prediction strategy, essential to exclude the query runner as the retrieval includes the query with similarity 1.0. Similarity 1.0 tells the retrieval results are correct and when comparing the two identical cases the global similarity output that they are equal.

We transformed the response from a JSON object to a two-dimensional size-mutable array with *pandas* DataFrame. Figure 5.9 displays code for *mean* strategy where we used *numpy*[2] library for mean nPB and PB variables. The method returns the relationship between query nPB and the mean PB. Multiplying the difference with the mean PB from the retrieved cases equals the predicted finish-time. Figure 5.10 displays code for *median* strategy, whereas the structure is the same as *mean strategy*. We used *numpy* to find the median for nPB and PB. We found the predicted time by multiply the difference with query nPB. After finding predicted finish-times without involving external race conditions, the altitude was corrections implemented (Figure 5.11) which corrects finish-times with the qualified assumption that altitude affects 0.6 hundredths of a second per increasing hPa per round (400m), from Chapter 5.6.3.

The results of both strategies shown in the listing below. First, the average similarity states that the ten cases retrieved are highly similar and therefore, reasonable representations from the case base. The two strategies are giving a nearly similar prediction. To analyze the differences in the two strategies needs many predictions with various conditions and data. We are studying the query altitude for the nPB race at $0MASL$ and average nPB altitude for retrieved cases at $316MASL$ and expect the corrected finish-times to be faster than the ones before the correction. This because the finish-times tend to be faster in higher altitude conditions. Based on the calculations from Chapter 5.6.3 a $316MASL$ difference equal 1.77 seconds, and subtracted from the predicted finish-time as average altitude is higher than query altitude.

[2] http://www.numpy.org/

```
Output:
    Average Similarity - 10 cases:
    0.7752661133506294
    ----
    Query nPB:  245.41 Query PB:  238.39
    ----
    Median Time without altitude: 238.07
    Mean Time without altitude:  238.32
    ----
    Query nPB altitude: 0.0
    Average nPB altitude:  316.0
    ----
    Median Corrected time:  236.29
    Mean Corrected Time:  236.55
```

This example presents a prediction of a query runner with an nPB at 245.41 and PB at 238.39 with race conditions of division A, and 0 MASL to be able to finish on 236.29 with the median approach and mean approach on 236.55. The average similarity for the ten retrieved and adapted cases is above average shown in Table 6.2 — the average output similarity at 0.7752 originating from the ten cases used in this example.

6.2 Results

In this chapter, we present the results from the experiments. There are three different retrieval approaches to explain, and in the next chapter, discuss the outcomes. As mentioned earlier, it is difficult to conclude whether the prediction is beneficial or incorrect based on the estimated finish-times. We do not have a definitive solution for the predictions since we want to achieve improvements for the query case. However, what gives an impression of whether the estimates are optimistic or realistic is how close to the nPB and PB finish-time the predicted results ends up. All results will be available in the Appendix.

6.2.1 Result Goals

As stated in the introduction, the preliminary objective is to predict the best possible finish-times for speed skaters given external conditions. It is necessary to provide results that visualize and test which strategy most suitable for prediction, as referred to in the research questions. We expect to present an implementation from a CBR system, including retrievals and reuse. We mainly focus on the performance of the two strategies and how they behave when increasing the retrieved cases. We are dealing with athletes and possible improved times, therefore, will prediction error not be our focus. However, more interesting to see how strategies behave in different conditions. We are showing the results for 5 and 10 retrieved cases because there are generally most differences between the two retrievals.

6.2.2 Feature Selection

Table 5.2 shows the case features implemented in the system, however our experiment tests *nPB, PB, Division, nPBMasl, PBMasl, SB, SBMasl, AllRaces*. The reason why we do not test all implemented case features is that we want to present results with high similarity, and we want statistically significant retrievals. Since our case base has a limitation in not including all case features for all cases, we are experimenting on the most distributed case features.

6.2.3 Similarity for retrievals

The essential criteria for an accurate prediction are to produce high similarity values on the retrieved cases. We achieve the desirable similarity from developing accurate similarity measures so that cases can be classified correctly and give an accurate reflection of the knowledge model. The following results concerning k number of cases retrieved illustrates similarity from 1.0 to 0.0 calculated by the similarity measures. Table 6.2 displays the average similarity for increasing number of retrieved cases (*k*) for 5 til 25 for nPB and SB retrieval. The average similarity is reasonable decreasing when the number of cases increases. We are retrieving the five most similar cases and then repetitively adding 5 more cases, whereas the first five cases always are included. What differs the different retrievals is that we include cases with lower similarity. For this specific purpose, an advantage with as few cases as possible. Later in this chapter will we visualize the behavior of the two strategies concerning the increasing number of cases retrieved.

	# of retrieved cases				
	5	**10**	**15**	**20**	**25**
Similarity nPB	0.7755	0.7388	0.7075	0.6858	0.6698
Similarity SB	0.6721	0.5895	0.5378	0.4893	0.4623

Table 6.2: Average similarity for nPB and SB based retrieval

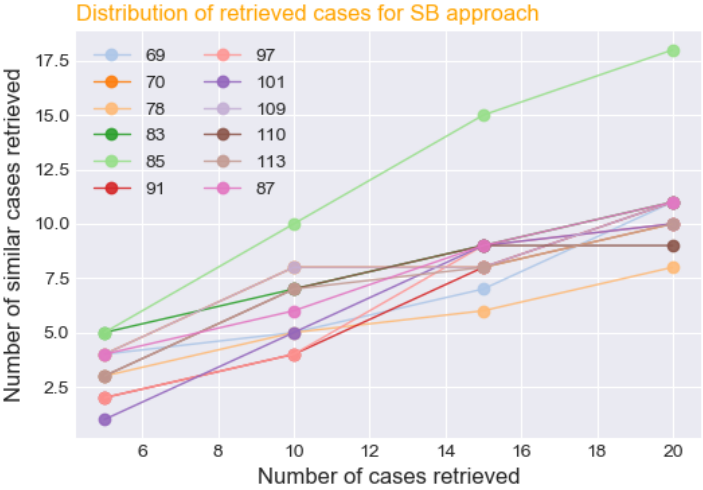
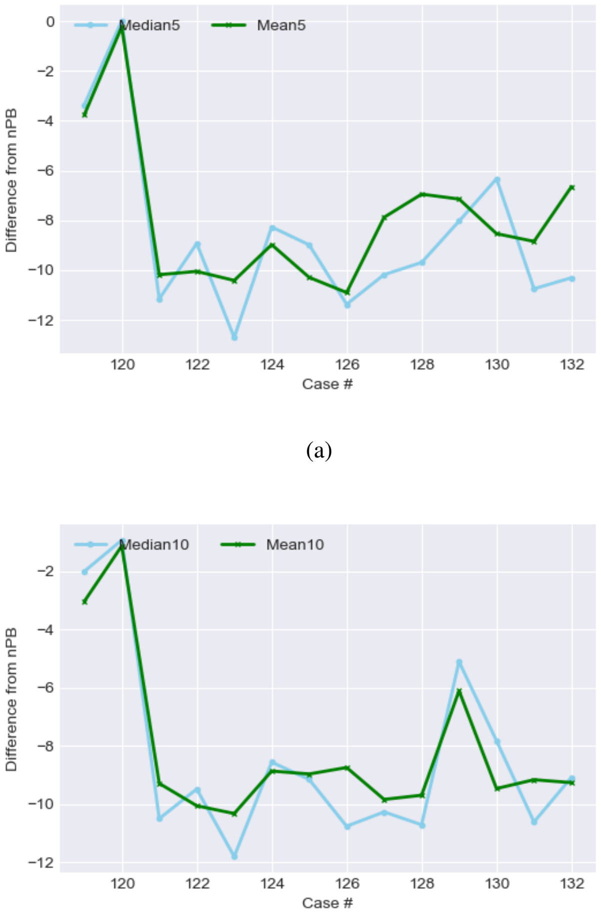


Figure 6.1: Line plot over retrieved similar cases

One can see that the nPB approach with a more extensive case base for retrievals gives better similarities than the SB approach where not all the cases are qualified with all case features. From 5 to 15 retrieved cases will the nPB approach only fall 0.07 in average similarity, but the SB approach ends up decreasing 0.12. Figure 6.1 shows a spaghetti-plot where *x-axis* is the number of cases retrieved from the case base, and *y-axis* the number of similar cases reused after filtering the not qualified cases. As we see, is only one of the test cases close to using the same amount of cases retrieved in the prediction. In the nPB retrieval are all the test cases equally in cases reused, which is a consequence of a more extended case base.

6.2.4 nPB finish time improvements

The first experiment, and also the main experiment for this work, is a prediction based on the speed skaters non-personal-best time. As explained in the implementation is nPB a randomly 3000 m race satisfying the requirements. There is no date interval for the nPB race, so the quality of the cases varies. Some cases can represent a fairly new race, where other cases can include races from early last season. We use nPB races because any race can be categorized as an nPB race, and we include a diversity of cases that makes the case base compatible with various problems.



(a)

(b)

Figure 6.2: Difference (in seconds) from nPB finish-time with (a) 5 and (b) 10 similar cases for *median* and *mean* strategy

Figure 6.2 shows us the difference between the predicted time and the nPB for both *median* and *mean* strategy at five and ten retrieved cases. In both graphs (a and b) is the *median* strategy more optimistic than the *mean* strategy. Looking at (a), we see that for the smallest difference between prediction and nPB are the two strategies almost equal, and for cases that differ more than 8 seconds do the two graphs behave differently. Also interesting is that the *median* strategy tends to vary more from five to ten case. Looking at the distribution for *median* strategy, Figure 6.4, are retrieval of 15 similar cases the one that stands out as the most inconsistent. The other three retrievals (5,9,10) are almost equal for most test cases. The inconsistent for increasing number of cases depends on calculating average turns out to be better for more cases, and median calculation perform better on more similar cases than additional cases. For *mean* distribution will more cases gain the average prediction and we can see that the retrieval for five similar cases vary more than for larger number of cases retrieved. $m9$, $m10$, $m15$ are almost identical for all test cases, when $m5$ have bigger variations.

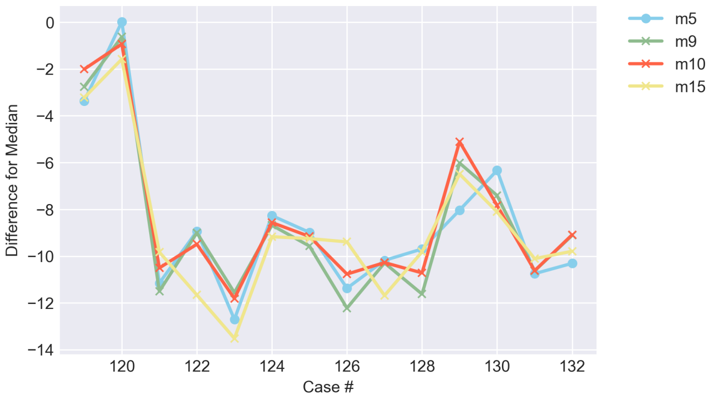


Figure 6.3: Median differences (in seconds) from nPB for 5,9,10,15 retrieved cases

Figure 6.5 illustrates the average difference for nPB time and the predicted finish time. We want to use this figure to illustrate that the median approach increase the difference, which can be interpreted as weaker prediction the more cases retrieved. It is opposite for the *mean* strategy where the difference are reasonable equal for all retrievals.

6.2.5 SB finish times improvements

Secondly, we implemented a retrieval based on the speed skaters seasonal best time from last season (2017/2018). Since not all cases had an SB time recorded, we filtered an reused the retrieved cases including all parameters. This procedure amputates the case base and also a variety of cases. Figure 6.1 illustrates the number of cases qualified compared to the number of cases retrieved. Figure 6.6 shows the difference between SB and predicted time for (c) five retrieved cases and (d) ten retrieved cases. By looking at Figure 6.6 (c), we see

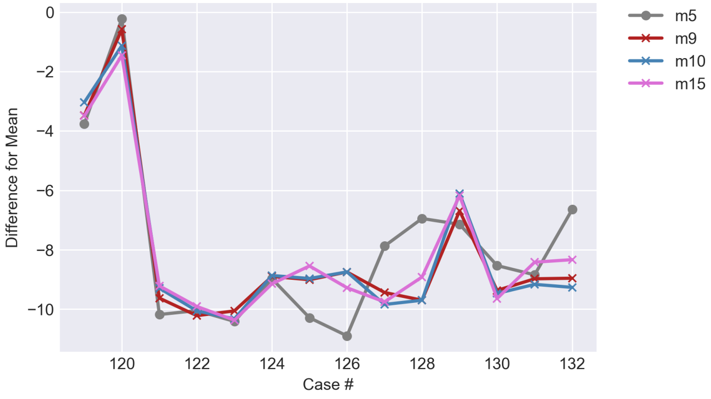


Figure 6.4: Mean differences (in seconds) from nPB for 5,9,10,15 retrieved cases

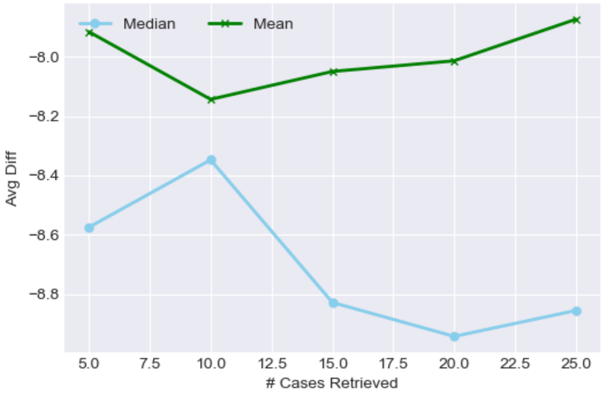
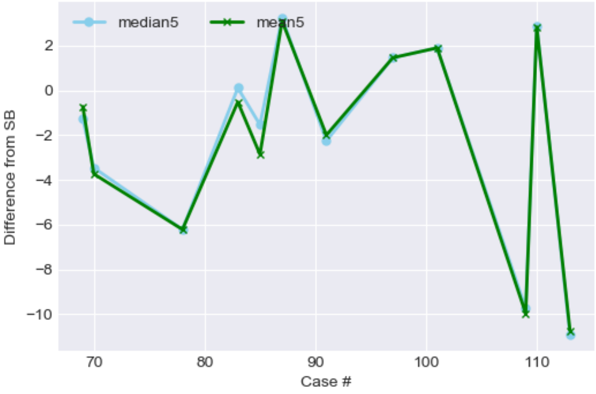
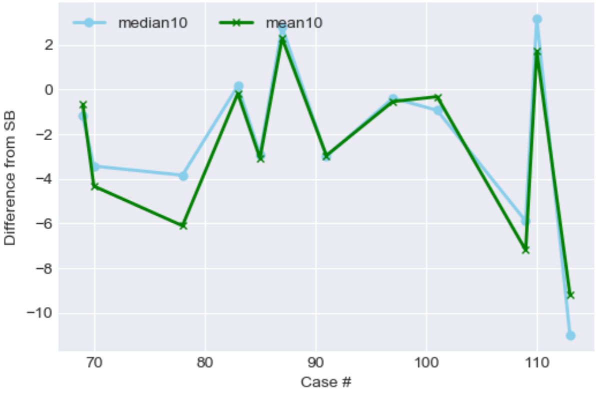


Figure 6.5: Average difference (in seconds) between nPB and predicted finish-time for # cases retrieved

that the two strategies perform equally for all cases tested. There are no remarkable differences worth commenting, however the differences more significant for (d). Compared to nPB retrieval are the SB curves more equal, and behave likewise. For (d) ten cases, we see the *mean* strategy tend to predict slightly faster times than *median*. This observation is the opposite of nPB results from the previous chapter. The SB predictions are more stable independent of strategy, whereas nPB approach has more variation between the two strategies.



(c)



(d)

Figure 6.6: Difference (in seconds) from predicted and SB finish time with (c) 5 and (d) 10 similar cases for *median* and *mean* approach

6.2.6 SB vs nPB

In this section, we compare the two approaches already presented. We want to study which prediction approach performing best (SB or nPB). Figure 6.7 and 6.8 compare the two approaches over five similar cases retrieved ((e) and (f)) and over ten retrieved cases ((g) and (h)). We ran the two approaches on the same test cases and compared the difference in predicted time compared with SB.

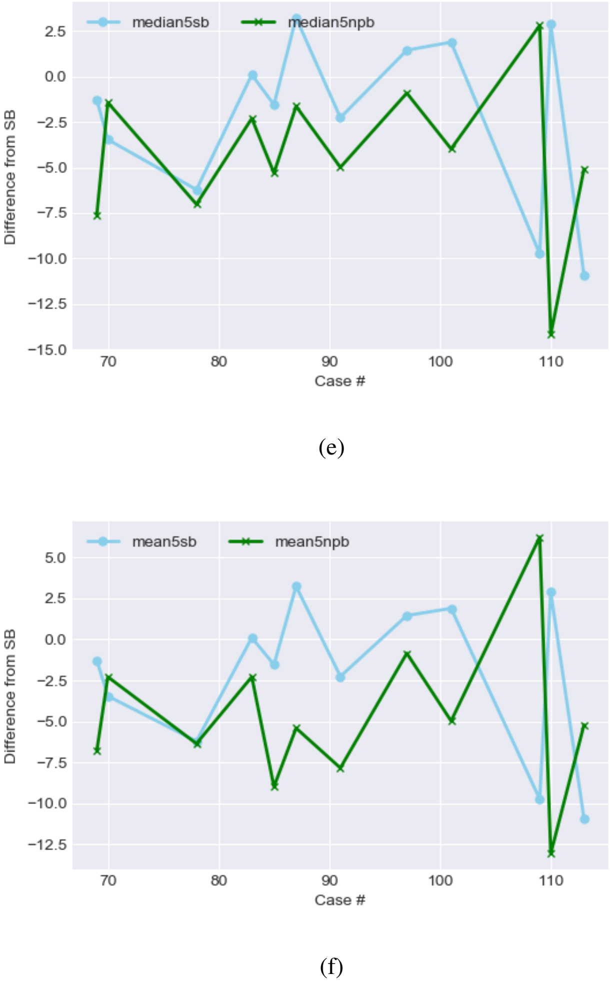
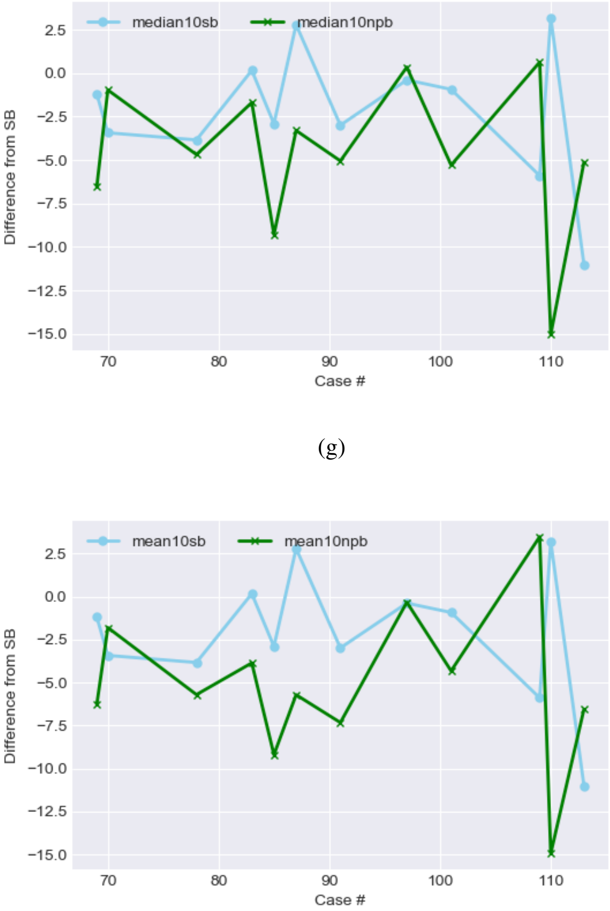


Figure 6.7: Difference (in seconds) for SB and nPB approach with 5 similar cases with (e) *median* and (f) *mean* strategy

Studying at the four graphs, we conclude that nPB approach predicts the most positive and faster results, and SB tends to be more stable and less variance. For all experiments are nPB predicting faster finish-times than SB approach. What limits the nPB approach is the variation for each case. As a example is Case 85 where *median* strategy predicts -5.30 from SB using five cases and $-9,28$ using ten cases. SB approach predicted $2,27$ and 0.90 for Case 85 with the same retrieval strategies. It is reasonable for nPB to vary more within the experiments because the number of cases retrieved is the number of cases used for adaptation. Figure 6.1 shows how many cases reused and adapted after excluding invalid cases for SB approach. A case is not usable if it not contains an SB race.

In this section, we focus on comparing the differences between approaches rather than between adaptation strategies. Table 6.2 supports Figure 6.7 and 6.8 as the similarity decreases and the prediction results for SB approach vary when increasing number of cases retrieved. We discuss the consequences of both approaches in the next chapter.



(g)

(h)

Figure 6.8: Difference (in seconds) for SB and nPB approach for 10 similar cases with (g) *median* and (h) *mean* strategy

6.2.7 World Cup Finals

Lastly, we predicted the outcome of the World Cup Finals in Salt Lake City based on the most recent race, which had similar race conditions. The complete implementation can be found in Chapter 5.7.3. The case base for this retrieval contained 23 cases, and all cases had nPB representing the World Allround Championship result, PB and SB from the 17/18 season. Table 6.3 shows the results from the experiment when retrieving 5 and 10 similar cases. Figure 6.9 is an comparison of the *median* and *mean* strategy and the actual result from World Cup Finals for 10 similar cases retrieved, which gives the best results (See Figure 6.5). *X-axis* is duration of a race in seconds, and *y-axis* the athlete competing. The green circle ($\bigcirc$) is the predicted *mean* finish time, red diamond ($\Diamond$) the predicted *median* finish time and the blue ($\times$) the actual world cup final result. The two prediction strategies calculate similar finish times for all athletes, but the *mean* strategy tends to predict slightly faster times than *median* strategy. In the time interval, $235 - 242$ seconds, both strategies perform well. This result comes from retrieving the ten most similar cases. Out of the eight test cases did 50%, predicted the *median* strategy a finish time within 0.5 seconds from the actual Word Cup Final result. $62,5\%$ of the cases were within one second from the result, and the most significant deviation was under 4 seconds. However, the PB best time for that specific case was only $+2.1$ seconds slower than the predicted time. The *mean* strategy does not perform as well as a *median* strategy. $37,5\%$ of the test cases is 0.5 seconds indifference. *Mean* strategy also tends to predict slightly faster times than the *median*.

World Cup Finals	5 (# cases) Median	5(# cases) Mean	10 (# cases) Median	10(# cases) Mean
03:52,02	03:57,71	03:54,61	03:50,44	03:50,50
03:54,06	03:58,05	03:57,79	03:56,14	03:55,12
03:55,58	03:58,25	03:58,41	03:55,25	03:55,25
03:55,73	03:58,02	03:57,10	03:56,12	03:55,97
03:59,00	03:59,66	03:58,89	03:58,08	03:58,13
03:59,05	04:02,38	04:00,52	03:58,68	03:58,32
04:00,76	04:01,92	04:02,34	04:00,72	04:01,04
04:07,05	04:04,78	04:03,78	04:03,41	04:02,75

Table 6.3: Results from World Cup Finals Prediction with Median Approach

# of retrieved cases

	5	10	15
Similarity WC	$0,4657$	$0,3202$	$0,2209$

Table 6.4: Average similarity for World Cup Finals retrieval

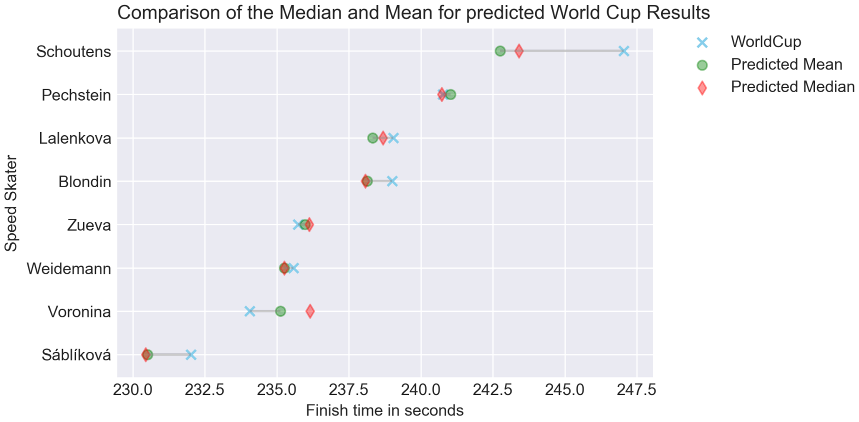


Figure 6.9: Comparison of *median* and *mean* approach for World Cup Finals

Average similarity decrease when retrieving more cases, shown in 6.4. We achieve the most accurate results when retrieving ten cases because the two strategies work well with a larger number of cases. However, in this experiment will the accuracy and similarity of cases used decrease if we retrieve more than half of 23 cases large the case base.

Chapter 7

Discussion & Obstacles

7.1 Discussion

In this work have we designed, implemented, and tested a case-based reasoning system for prediction of best possible finish times for speed skaters. We created three research questions representing the problem we investigated throughout this research.

First, we investigated how to use CBR for predicting results for speed skating, where we designed a system with several case features that characterize a speed skater. Case-based reasoning is a suitable approach for solving the research problem and recommended for similar studies. We implemented the retrieval and reuse phase, and we are not retaining solved problems. The reason is that our work is state of the art for speed skating contributing to machine learning, and the focus is to investigate if CBR could be applied to speed skating as it already does for running. Secondly, we research how a combination of numeric and non-numeric parameters can improve the prediction. We found multiple race factors affecting speed skating results that could transform into a non-numeric case feature; however, the results are mainly affected by numerical features. Non-numeric case features compose an extended case base with richer case representations, which in the long run, increase the similarity. We used division and consecutive races as non-numeric features in the experiment. Lastly, we developed strategies for predicting finish-times and tested which strategy was most proper and which attributes most valuable. The *median* and *mean* strategies were tested and evaluated in two different experiment approaches (nPB and SB). We concluded that *median* strategy is the most encouraging strategy — and *mean* the most stable for retrievals with a larger amount of cases. Most valuable attributes are the external race condition features, such as altitude. The season-best case feature created more equal predictions for both strategies and also less optimistic and probably more realistic finish times. From the three experiments performed, we observed several major findings.

7.1.1 Best prediction strategy

First, out of the two strategies implemented and tested did the *median* strategy deliver the most ambiguous finish times, and *median* strategy, the most stable predictions related to the number of cases retrieved. Figure 6.5 illustrates the average differences between nPB and predicted finish time for nPB retrieval. *Median* strategy is constantly predicting average faster finish-times independent of the number of cases used in the adaptation process. What is also making *median* to the most ambiguous and varying strategy is that the model operates with a small data set. The definition of median refers to the number separating the data set in lower and higher half [34]. Small data set are usually not statistically significant because it is more likely that the results occur by random chance that due to the experiment. The *median* strategy provide faster finish-times when we increase the number of cases included because the median of races retrieved most likely have a more significant difference to PB than the best-retrieved cases. The more cases retrieved, the less comparable will the median case be. *Mean* strategy behave differently, also visualized in Figure 6.5. The average difference fades for increasing cases retrieved. Studying Figure 6.6, we observe similar behavior for the SB approach. *Median* strategy tend to vary with an increasing number of cases, where the *mean* strategy reflects a more reasonable curve even when cases increase. A strategy predicting faster finish-times does not mean it is better than a strategy recommending slower finish-times. A slightly positive recommendation provides a psychological effect that benefits coaches and athletes. With such an argument in mind is it beneficial to predict finish-times that can improve and challenge the speed skater, rather than of demotivate. We could have included analyses from coaches and athletes and evaluate if they rather compete with ambiguous or achievable goals. *Median* strategy provides ambiguous predictions and *mean* strategy more achievable goals. We prefer more ambiguous and challenging as the motivation is to predict new best-possible finish-times.

Likewise conclusions was drawn by Smyth and Cunningham [52; 51; 53] in their marathon analysis. They used *best*, *mean*, and *even* strategy based on marathon pacing plans to help marathon-runners achieve a personal-best in their next race. Their findings pointed out that *mean* strategy gave the most accurate PB finish-time and pacing recommendation. The *mean* strategy predicts the most stable results and therefore, easy to conclude as the most accurate ones as well. We did not resonate over which strategy is most accurate because we receive the best possible finish-times, and there is no efficient way to validate the results. The focus is to convey how the two strategies behave in various experiments. What differs their research from our is that they include marathon pacing plans that influence the results. Smyth and Cunningham based the retrieval on numeric features, where the objective was to help to find a new personal-best time while we predict the best possible result given specific conditions. What is encouraging for future research in recommending systems is that we achieve similar conclusions on different sports and objectives.

7.1.2 Best retrieval approach

The second significant finding is related to the preferable feature selection. We experimented with two different retrieval approaches, nPB, and SB based retrieval. An essential precondition is that the SB approach operates with an amputated case base because several cases are missing the SB case feature. The nPB approach turned out to give the best predictions because $1)$ we use the knowledge base to a full extent, and $2)$ the similarity on the cases retrieved are significantly higher (Table 6.2). Moreover, Figure 6.6 demonstrates that nPB provides the most ambiguous prediction results, and SB presents the most consistent results. We can not exclude that the SB results due to random change instead of the CBR system we created. The similarity should be significantly higher for the SB results to be statistically significant. The SB approach is an approach performing great and has advantages future implementations should appreciate. To maintain the SB case feature is expensive and time-consuming since all race needs to be evaluated and extended with the correct season-best and also yearly update the parameter. Related research discusses issues related to the quality of races [8; 52]. An nPB can give a false impression on athlete expected performance, as well as the PB can be years old and not give the right relation to SB or nPB. In [51], one of the leading suggestions for further work is to continue to investigate other factors that can impact the marathon prediction performance, whereas the time between races and multiple races was likely to improve the system. In the follow-up article [53], they extend the case representation with more races describing specific pacing strategies and race outcomes. However, the results in Figure 6.6 tells that including multiple races increase the stability and reduce the variations that occur with merely using an nPB race.

There is an expected outcome that nPB performs better than SB overall because the implemented data for SB are narrowed down to a small number of cases. Even if the SB approach applies more case features, as concluded in [32] increase the accuracy, will the size of the case base be a limitation affecting the results. When reusing more cases, it implicates that the cases added for each retrieval is less similar than the cases already used. We are always retrieving the most similar cases, so the difference from retrieving 10 and 15 cases is that we append the next five similar cases. Many cases strengthen the *median*, and *mean* strategy because we achieve a more realistic prediction, but also increases the potential error.

Marathon runners vary within when they maximize their potential [8], and speed skaters vary in time achieving the seasonally best time. Accordingly is SB a necessary supplement to the knowledge base because and the CBR system recommends a finish time based on the speed skaters maximal potential during a year. In further work, including SB to more cases improves the prediction in stability and accuracy.

7.1.3 Combining non-numeric and numeric parameters

Differing our research from other studies of sports prediction did we investigate several non-numeric features. We aimed to evaluate if there are potential for non-numeric features as it often should be included to cover the domain fully.

The third finding, and also the explanation to the second research question, is that combining non-numeric and numeric features affects the prediction positively. Non-numeric parameters exist in every problem space and well needed for categorizing and sorting the cases efficiently. The system becomes scaleable because we manage the cases with local similarity measures instead of manually in the data set. Polynomial similarity function allows us to model the system highly detailed, whereas, in the implemented system, we prefer numeric features for retrieval and non-numeric features for categorizing and labeling approaches. Non-numeric features are used to include and exclude cases in retrieval

We aimed to investigate the consequences and usability for non-numeric feature due to the many factors affecting speed skating, which quickly transforms into case features. The intentional CBR system includes athlete records, biological characteristics, and extended race conditions and can easily develop as symbolic case features. Collecting symbolic parameters has been studied in previous research. Novatchkov and Baca's AI study [32] included *sex* and *experience* as biological characteristics in the supervised classification (ANN) model.

7.1.4 Case-Base reasoning as a suitable machine learning method

The main objective is to consider if case-based reasoning is suitable for speed skating problem-solving scenarios. After reviewing the facts related to speed skating and implementing a system describing the problem, we can conclude that CBR is a proper artificial intelligence method for recommending and predicting finish-times for speed skaters.

We base this finding on the conclusions from other research and the findings described above. We have reproduced the methodology from other studies with *myCBR* as an efficient modeling tool and achieved related results. Speed skating, as mentioned, is a sport that is suitable for feature extraction with several essential external parameters that can be processed to match a CBR system. One of the issues in the marathon recommendations [52; 53; 8] were the lack of different races in various conditions. In speed skating, it is more accessible to manage data from different situations and then achieve an extended knowledge base with diversity in problems and solutions.

Furthermore, the prediction error for a future live feedback system can have significant consequences. A CBR system must contain a large amount of data to be as accurate as necessary. Looking at the results from World Cup Finals predictions, where the CBR model provides reasonable recommendations, implies that fully developed systems have a future in speed skating prediction. 50% of the cases predicts finish times within 0.5 seconds from the actual results. This example was made to illustrate the power of CBR systems and to show real-time examples where we easily can compare the estimated fin-

ish times with results from competitions. However, the similarity from the experiment states that the cases used were not as similar as they should. nPB retrieval had an average similarity around 0.7388, whereas we only achieved 0.3202 for 10 cases retrieved. The case base contained only the 23 athletes competing at the World Allround Championship, which is why the similarity is remarkably low.
CBR is essential for the future of machine learning, and they are the driving power to keep developing the methodology [1]. We can apply CBR to various environments where it is difficult to formalize [16] and supports the thoughts and findings as speed skating includes exciting features needs to be processed differently than typical numeric analyses. Speed skating experts have, in a long time, studied different pacing strategies [56; 31] but never included external factors.

Even if working with CBR accommodate many limitations remains the goal to investigate the possibilities in using CBR as a predicting tool, and we imply that speed skating is a suitable sport, and CBR a proper method. No doubt that case-based reasoning can help athletes and coaches improving their best times.

We learned and explored a lot in both domains we touched upon during this work. We learned that CBR is an efficient and suitable methodology for almost any problems as long as one collects the foundation data and develop a case structure that fits the CBR cycle. In [53], [27] and [8], future work suggest to contribute to other sports than the one explored in the research. We contributed to the Artificial Intelligence and Case-Based Reasoning field when a prediction system to a new sport and studying the important factors affecting the results.

7.1.5 Modeling with *myCBR*

Lastly, it is essential to mention *myCBR* and evaluate how this tool helped us modeling the system. We created the knowledge model and defined similarity measures in the*myCBR Workbench* and later used the *myCBR SDK* for retrieval. *myCBR* is a easy-to-use open-source application for primarily retrievals. Another study evaluating *myCBR* implemented *COLIBRI Studio* that provides the visual builder tools required to generate CBR systems without dealing directly with the source code[45].

Bach et al. [5] evaluated the knowledge modeling with *myCBR*, where they state that in further work additional features for automatic extraction of vocabulary items and similarity measures be incorporated. After using *myCBR*, can not see any limits with the *Workbench* and *SDK*. An improvement could be to integrate adaption processes easily. However, the *SDK* allows us to implement customized processes for reusing the retrieved cases.

7.2 Obstacles & Limitations

In this study, several factors can affect the prediction. This section covers the challenging parts related to machine learning, case-based reasoning, sports science, and biological circumstance.

7.2.1 Limitations in Case Based Reasoning

All phases in the CBR cycle [2] have limitations, whereas we focus on the ones used in the implementation. It is essential to discuss is the knowledge modeling required for CBR systems [37]. Modeling of similarity measures are necessary for a sustainable and accurate system and requires expert domain knowledge, which is highly expensive. We have mentioned the importance of a well-organized data set. There are in high interest to study modeling features in noisy environments [27] and also limitations related to collecting features.

The difficulty in constructing systems describing the real world is thoroughly studied. Kofod-Petersen [23] observed the issue where cases describe a situation or event, instead of constructing cases based on user models and system. We proceeded the same way when designing our system, and building a case feature before designing the system creates discussions in how we should compare cases. Therefore essential to have domain knowledge, so the knowledge domain represents the problem and solution space clearly and detailed. Case base maintenance is an expensive process and requires automation systems for keeping the system updated. Many systems do not need maintenance as our system does, which limits the development.

After reusing the cases in the adaptation process, the cases regularly revised manually. Revising limits CBR systems because it requires domain experts and expansive workflows. Machine learning has challenges in explaining the results and decisions. The main goal for machine learning is to reduce the gap between digital modeling and real environments.

7.2.2 Obstacles working with sport

There are numerous obstacles when predicting sports results. The major assumption in this thesis and all scientific papers concerning sports is that athletes do their greatest to perform and achieve the best possible result. This is the motive for us to collect World Cup races as the main data source, and especially women, because they tend to stick to their race plan better than men [58], and speed skaters highly prioritize World Cup Competitions [31]. The human factor limits recommending systems. Humans make mistakes, and mistakes are crucial for prediction and error evaluation. The prediction can be highly accurate but perform poor because of small mistakes that have a consequence on the finish-time.

Another challenge when gathering data is when speed skaters compete for the first race of the season, which tend to represent a poor version of the athlete. We ignored the race order and dated to simplify the model and to reach out goals. In an intentional system, the data set should include races with related information of athletes current conditions and date of the competition. The external race conditions are parameters that can affect the results and should include in the case representation, but there are challenges in categorizing the disadvantages or advantages in a different situation. Some of these parameters can be hard to allocate from past races and limits data collecting for previous events.

It is difficult to control and observe all phases in a competition. A limitation stated by [32] explains the challenges in observing technical pre-conditions in weight lifting. Same challenges occur in speed skating, where technical adjustments and mistakes can influence the results. With no specific solution will this obstacle be a generalized problem when combining machine learning and sports.

Lastly, our experiment is limited by the assumptions for altitude consequences. We made a qualified assumption, where an accurate equation describing how altitude affects the results in speed skating is necessary for further work.

Chapter 8

Conclusion & Further Work

8.1 Further Work

We have in this thesis mentioned a couple improvements a long the way. This section will structure the plans, and discuss how we can improve the CBR system. Our plans include implementing more case features and enable experts to revise the solutions for further improvements. At the end of the tunnel is a feedback system where coaches and athletes can use a CBR system before, during, and after the race. Berndsen et al. [8] have similar plans with their research, making machine learning based recommendation systems an attractive implementation

8.1.1 Intentional System

First of all, we have described an intentional system design that implements all four “R’s” of the CBR cycle. The intentional CBR system is illustrated in Figure 4.1 and described in Chapter 4. We implemented the two first phases (*retain* and *reuse*), which we considered most critical. In the future, we will implement the missing processes, revise and retain. We have shown that CBR is a suitable method for predicting speed skating results, and expected expansion is to tune the system to predict accurate results usable for the athletes. We will develop a manual revision process where domain experts evaluate the predicted finish-times based on knowledge and actual results in the predicted conditions. Based on this revision process is the next implementation to retain the solved and tested cases so that new cases can learn from mistakes and successful predictions.

8.1.2 Case features

There are a couple of specific features we need to include in future work. Firstly, pacing plans can improve the recommendation and add a new dimension to the prediction. We started implementing this idea by creating a case feature *SkaterType* that categorized the speed skaters as sprinter or long-distance. Collecting data concerning pacing profiles would give new valuable input for the system, but also an expensive process. Case-based reasoning performs best with accurate similarity measures and a lot of knowledge and experience. By including pacing plans, the knowledge base expands and make the revision easier for the domain experts. Pacing plans imply split times for each round, where we also in the future separate the finish-times into segments. A prerequisite for pacing plans to work efficiently is only to include the longer distances, so that the model can cluster and categorize different pacing strategies.

Secondly, including more races in each case fixes the problem where the SB approach performed critically because of the small case base. We have shown that by including richer case representations and a large case base will the similarity increase, and the retrieved cases represent the problem better. We only included World Cup races, and there are several other competitions with equal status possible to include. Extend the knowledge base and create richer case representations are an expensive approach and depends on the resources available.

8.1.3 Live feedback system

For a system like this to have more than novelty interests, should evolve to a live feedback system. This master thesis started as a conversation between a computer scientist and speed skating experts about how to combine the two domains best possible. The most interesting suggestions, and also probably the most valuable for the athletes would be a system where the coaches give feedback during practice or competitions about the upcoming laps and how mistakes and race strategies affect the result. When a speed skater starts the race with a slow turn, the coach can predict the outcome of the race based on the mistake. Another approach is to learn from competitors during the competition. For an athlete will information and recommendations based on how the previous speed skater performed be highly valuable. A speed skater recently raced with a too ambiguous pacing strategy in the last Allround race and lost the first place. With pacing recommendations could the speed skater race with a safer approach and maybe won. This approach requires large data sets, domain knowledge, and an efficient retain process. Even if it is far stretched, it is essential to evaluate how to combine the two domains so that both can expand and grow.

There are no limits for what to include in a CBR system, and the machine learning domain needs more studies regarding sports science and artificial intelligence. All systems base their accuracy on experience, and so do researchers. We need to expand the view and include more sports, as well as the sports domain, need to investigate more about what is affecting the results and outcomes.

8.2 Conclusion

In this work, we have studied the use of case-based reasoning system when predicting the best possible finish-times for female speed skaters. It builds on related research and attempts to, by a working implementation, recommend a system for an other sport than running [8; 53].

The research goals for this project was to investigate how we can use CBR for predicting improved finish-times, how we can include numeric and non-numeric case features and lastly which prediction strategy suits best for the system. We present an intentional and implemented system design. Our implementation focused on the *retrieval* and *reuse* phase. In our further work, we are implementing the two remaining phases described in the intentional system, and strengthen the system with manually revision processes. To investigate our research objectives, we modeled a CBR system using *myCBR Workbench* and *SDK*. Parameters affecting speed skaters were collected and described as different case features. The *retrieval* includes developing similarity measures and *reuse* the most similar cases in an adaptation process using *median* and *mean* strategies for calculation the new predicted solution.

When evaluating our background research and results, we found that case-based reasoning is indeed a suitable machine learning methodology for predicting speed skating results due to easy adaptation process and many suitable features affecting a race. One step in improving the state of art feedback systems was to incorporate numeric and non-numeric case features. We found *median* strategy to be the most optimistic and satisfying strategy with the limited case base, also emphasized in our World Cup Finals example. However, the results are presented in Chapter 6.2 visualize both strategies as efficient but *median* strategy will challenge the athlete more.

We have contributed to the AI community with cutting edge transformation of sports scientific parameters and integrated them to a CBR system. Related research has predicted personal bests and recommended pacing plans, however we maintained external conditions related to a specific case and adapted the domain knowledge when calculation finish-times. Similar research has not previously been published, and future studies will elaborate more regarding the case representation. This study can contribute to a live feedback system where coaches and athletes can predict and analyze races simultaneously.

The system can help athletes pace themselves during competitions and challenge to perform faster times or skate smarter. For training purposes coaches can use this research as a base for how they can plan their training and the results they should aim for. We can expand this research by building a extended case base, manually revise and lastly retain the improved cases. Our results were limited due to a small case base and in the future more parameters affecting speed skating should be included.

Bibliography

[1] Aamodt, A. [n.d.], 'Case-based reasoning - an introduction'. [Online; accessed 02.05.2019].

[2] Aamodt, A. and Plaza, E. [1994], 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *AI Commun.* **7**, 39–59.

[3] Ahn, H. and Kim, K.-j. [2009], 'Bankruptcy prediction modeling with hybrid case-based and genetic algorithms approach', *Applied Soft Computing* **9**, 599–607.

[4] Bach, K., Althoff, K.-D., Newo, R. and Stahl, A. [2011], A case-based reasoning approach for providing machine diagnosis from service reports, Vol. 6880, pp. 363–377.

[5] Bach, K., Sauer, C., Althoff, K.-D. and Roth-Berghofer, T. [2014], Knowledge modeling with the open source tool mycbr, Vol. 1289.

[6] Bartlett, R. [2006], 'Artificial intelligence in sports biomechanics: New dawn or false hope?', *Journal of sports science medicine* **5**, 474–479.

[7] Begum, S., Ahmed, M., Funk, P., Xiong, N. and Folke, M. [2011], 'Case-based reasoning systems in the health sciences: A survey of recent trends and developments', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **41**, 421 – 434.

[8] Berndsen, J., Lawlor, A. and Smyth, B. [2017], 'Running with recommendations'.

[9] Bichindaritz, I. and Marling, C. [2010], *Case-Based Reasoning in the Health Sciences: Foundations and Research Directions*, Vol. 309, pp. 127–157.

[10] Brownlee, J. [2018], 'A gentle introduction to k-fold cross-validation'.
URL: *https://machinelearningmastery.com/k-fold-cross-validation/*

[11] Carlos Rafaell Correia-Oliveira1, Mayara Vieira Damasceno1, A. d. A. M. R. B. e. M. A. P. D. K. [2016], 'Pacing strategy in speed skating: Influence of the gender and race's performance level in 1500-m distance'.

[12] Cheng, H., Garrick, D. J. and Fernando, R. L. [2017], ‘Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction’, *Journal of Animal Science and Biotechnology* **8**(1), 38.

[13] Craw, S. and Preece, A. [2002], *Advances in Case-Based Reasoning: 6th European Conference, ECCBR 2002 Aberdeen, Scotland, UK, September 4–7, 2002 Proceedings*.

[14] DAS [2010], ‘Pressure with height’.
URL: *http://ww2010.atmos.uiuc.edu/(Gh)/guides/mtr/prs/hght.rxml*

[15] Díaz-Agudo, B., Plaza, E., Recio-García, J. and Arcos, J. L. [2008], Noticeably new: Case reuse in originality-driven tasks, pp. 165–179.

[16] El-Sappagh, S. H. and Elmogy, M. [2015], ‘Case based reasoning: Case representation methodologies’, *International Journal of Advanced Computer Science and Applications* **6**(11).
URL: *http://dx.doi.org/10.14569/IJACSA.2015.061126*

[17] Fong, K. [2018], ‘Olympic oval needs upgrades regardless of 2026 olympics’.
URL: *https://www.660citynews.com/2018/11/11/olympic-oval-needs-upgrades-regardless-of-2026-olympics/*

[18] Fuller, S. [n.d.], ‘Topic: Running jogging’.
URL: *https://www.statista.com/topics/1743/running-and-jogging/*

[19] H.Lohninger [2006], ‘Cross-validation’.
URL: *http://www.vias.org/tmdatanaleng/cc*$_{c}ross_{v}alidation.html$

[20] Hoffmann, R. [1981], ‘Box plot: Display of distribution’.
URL: *http://www.physics.csbsju.edu/stats/box2.html*

[21] IBU [2015], ‘Communication no. 1958’.
URL: *https://www.yumpu.com/en/document/view/55300537/international-skating-union-communication-no-1958*

[22] Kang, Y. B., Krishnaswamy, S. and Zaslavsky, A. [2011], Retrieval in cbr using a combination of similarity and association knowledge, pp. 1–14.

[23] Kofod-Petersen, A. [2006], Challenges in case-based reasoning for context awareness in ambient intelligent systems.

[24] Kwon, N., Lee, J., Park, M., Yoon, I. and Ahn, Y. H. [2019], ‘Performance evaluation of distance measurement methods for construction noise prediction using case-based reasoning’, *Sustainability* **11**, 871.

[25] Lapham, A. and M Bartlett, R. [1995], ‘The use of artificial intelligence in the analysis of sports performance: A review of applications in human gait analysis and future directions for sports biomechanics’, *Journal of sports sciences* **13**, 229–37.

[26] Lees, B. [2007], Cbr systems with smart revision algorithms.

[27] McCabe, A. and Trevathan, J. [2008], Artificial intelligence in sports prediction, *in* 'Fifth International Conference on Information Technology: New Generations (itng 2008)', pp. 1194–1197.

[28] McClennan, J. [2018], 'speedskatingresults.com'.
URL: *http://speedskatingresults.com/index.php?p=32*

[29] Muehlbauer, T., Schindler, C. and Panzer, S. [2010], 'Pacing and sprint performance in speed skating during a competitive season', *International journal of sports physiology and performance* **5**, 165–76.

[30] Mántaras, R., Mcsherry, D., G. Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., D. Forbus, K., Keane, M., Aamodt, A. and Watson, I. [2005], 'Retrieval, reuse, revision and retention in case-based reasoning', *Knowledge Eng. Review* **20**, 215–240.

[31] Noordhof, D., Mulder, R., de Koning, J. and G Hopkins, W. [2015], 'Race factors affecting performance times in elite long-track speed skating', *International journal of sports physiology and performance* **11**.

[32] Novatchkov, H. and Baca, A. [2013], 'Artificial intelligence in sports on the example of weight training', *Journal of sports science medicine* **12**, 27–37.

[33] Núñez, H., Sànchez-Marrè, M., Cortés, U., Comas, J., Martínez, M., Rodríguez-Roda, I. and Poch, M. [2004], 'A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations', *Environmental Modelling Software* **19**(9), 809 – 819. Environmental Sciences and Artificial Intelligence.

[34] Ori, J. [2019], 'The effect of sample size on mean median'.
URL: *https://sciencing.com/the-effect-of-sample-size-on-mean-median-12744118.html*

[35] Peter, L., Mateus, D., Chatelain, P., Schworm, N., Stangl, S., Multhoff, G. and Navab, N. [2014], Leveraging random forests for interactive exploration of large histological images, Vol. 17, pp. 1–8.

[36] Pyne, D. B., Trewin, C. B. and Hopkins, W. G. [2004], 'Progression and variability of competitive performance of olympic swimmers', *Journal of Sports Sciences* **22**(7), 613–620. PMID: 15370491.

[37] Reuss, P., Dick, M., Termath, W. and Althoff, K.-D. [2017], 'Case-based reasoning: potential benefits and limitations for documenting of stories in organizations', *Zeitschrift für Arbeitswissenschaft* .

[38] Ricci, F. and Avesani, P. [1998], 'Learning an asymmetric and anisotropic similarity metric for case-based reasoning'.

[39] Richter, M. [2003], Knowledge containers.

[40] Richter, M. [2008], *Similarity*, Vol. 73, pp. 25–90.

[41] Richter, M. and Weber, R. [2013], *Case-based reasoning: a textbook*.

[42] Riegel, P. S. [1981], 'Athletic records and human endurance: A time-vs.-distance equation describing world-record performances may be used to compare the relative endurance capabilities of various groups of people', *American Scientist* **69**(3), 285–290.

[43] Roth-Berghofer, T. [2002], 'Six steps in case-based reasoning: Towards a maintenance methodology for case-based reasoning systems'.

[44] Roth-Berghofer, T. [2004], Explanations and case-based reasoning: Foundational issues, Vol. 3155, pp. 389–403.

[45] Roth-Berghofer, T., Sauer, C., Recio-García, J., Bach, K., Althoff, K.-D. and Diaz Agudo, B. [2012], Building case-based reasoning applications with mycbr and colibri studio.

[46] Sae-Hyun Ji, Moonseo Park, H.-S. L. and Yoon, Y.-S. [2010], 'Similarity measurement method of case-based reasoning for conceptual cost estimation'.

[47] Sauer, C. [2016], Knowledge elicitation and formalisation for context and explanation-aware computing with case-based recommender systems, PhD thesis.

[48] Schneider, J. [1997], 'Cross validation'.
URL: *"https://www.cs.cmu.edu/ schneide/tut5/node42.htmlfigcv0"*

[49] Shekapure, S. [2015], 'Problem solving using case based reasoning methodology'.

[50] Smith, T. B. and Hopkins, W. G. [2011], 'Variability and predictability of finals times of elite rowers'.

[51] Smyth, B. and Cunningham, P. [2017*a*], A novel recommender system for helping marathoners to achieve a new personal-best, pp. 116–120.

[52] Smyth, B. and Cunningham, P. [2017*b*], 'Running with cases: A cbr approach to running your best marathon', *Lecture Notes in Computer Science, vol 10339* .

[53] Smyth, B. and Cunningham, P. [2018], 'An analysis of case representations for marathon race prediction planning'.

[54] SpeedSkatingStats.com [2018], 'Current world records'.
URL: *http://www.speedskatingstats.com/index.php?file=recordsg=m*

[55] Stahl, A. [2003], 'Using evolution programs to learn local similarity measures'.

[56] T Muehlbauer, S. P. and Schindler, C. [2010], 'Pacing pattern and speed skating performance in competitive long-distance events', *Journal of Strength and Conditioning Research* .

[57] Thibault, V. e. a. [2010], ‘Women and men in sport performance: The gender gap has not evolved since 1983’, *Journal of sports science medicine vol. 9,2 214-23.* .

[58] W Trubee, Nicholas M Vanderburgh, P. . D. W. . J. K. [2013], ‘Effects of heat stress and sex on pacing in marathon runners’, *Journal of strength and conditioning research / National Strength Conditioning Association* .

[59] Watson, I. [n.d.], Case-based reasoning. [Online; accessed 29.04.2019].

[60] Wikipedia [2018], ‘Speed skating rink’.
URL: *https://en.wikipedia.org/wiki/Speed*$_s kating_r ink$

[61] Wikipedia [2019a], ‘List of indoor speed skating rinks’.
URL: *https://en.wikipedia.org/wiki/List*$_o f_i ndoor_s peed_s kating_r inks$

[62] Wikipedia [2019b], ‘Long track speed skating’.
URL: *https://en.wikipedia.org/wiki/Long*$_t rack_s peed_s katingCompetition_f ormat$

[63] Wikipedia [2019c], ‘Speed skating’.
URL: *https://en.wikipedia.org/wiki/Speed*$_s kating$

[64] Yeow, D. [n.d.], ‘Sochi special: The difference between short track and long track’.
URL: *https://www.danielyeow.com/2014/sochi-special-short-track-long-track/*

Appendices

Appendix A

Code

A.1 boxplot.py

```
import numpy as np
import matplotlib.pyplot as pltb
import pandas as pd

df = pd.read_csv( ’casebase.csv’, delimiter=’;’)
df.head(1)

def boxplot():
    df[["PB"]].boxplot()
    df[["nPB"]].boxplot()
    df[["SB"]].boxplot()

def main():
    boxplot();

main()
```

A.2 strategies.py

```
import numpy as np

def median(q_nPB, similar_cases, attribute):
    nPB_skaters = np.array(similar_cases[attribute])
    median_nPB = np.median(nPB_skaters)
    PB_skaters = np.array(similar_cases['PB'])
    median_PB = np.median(PB_skaters)

    weigthed_difference = float(q_nPB) / median_nPB
    return (median_PB * weigthed_difference)

def mean(query_nPB, similar_cases, attribute):
    npb_skater = np.array(similar_cases[attribute])
    mean_nPB = np.mean(npb_skater.astype(float))
    PB_skaters = np.array(similar_cases['PB'])
    mean_PB = np.mean(PB_skaters.astype(float))
    weigthed_difference = q_nPB / mean_nPB
    return (weigthed_difference * mean_PB)

def altitude(predicted_time, current_altitude,
   median_of_retrieved_altitude):
    difference_in_altitude = (current_altitude -
       median_of_retrieved_altitude)
    hpa = (difference_in_altitude/8)
    total_seconds_to_correct_for_3000 = 7.5 * (hpa*0.006)
    return (predicted_time +
       total_seconds_to_correct_for_3000)
```

A.3 nPBPredictions.py

```
import requests
import json
import pandas as pd
import numpy as np
import matplotlib.pyplot as pltb

#Complete calculation for nPB approach
test_cases = [119, 120, 121, 122, 123, 124, 125, 126, 127,
   128, 129, 130, 131, 132]
for x in test_cases:
    print("Case:␣", x)
    response = requests.get( url= "http://localhost:8080/
       retrievalByIDWithContent.json?casebase=
       cb_speedskaters&concept=speedskaters&amalgamation%20
       function=global&caseID=speedskaters"+str(x)+"&no%20
       of%20returned%20cases=26")

    response_json = response.json()
    df_response = pd.DataFrame(response_json)
    updated_response = df_response.iloc[1:26, : ]

    query = df_response.head(1)[’nPB’][0]
    retrieved_masl = float((df_response.head(1)[’nPB_masl’
       ][0]))
    npb_altitude = np.array(updated_response[’nPB_masl’])
    average_nPB_altitude = np.average(npb_altitude.astype(
       int))


    #Mean
    mean_prediction = mean(float(query), updated_response,
       ’nPB’)
    corrected_mean_prediction = altitude(mean_prediction,
       retrieved_masl, average_nPB_altitude.astype(int))

    #Median
    median_prediction = median(float(query),
       updated_response, ’nPB’)
    corrected_median_prediction = altitude(
       median_prediction, retrieved_masl,
       average_nPB_altitude.astype(int))
```

A.4 SBPredictioins.py

```
import requests
import json
import pandas as pd
import numpy as np
import matplotlib.pyplot as pltb

#Complete calculation for SB approach
SB_cases = [69,70,78,83,85,91,97,101,109,110,113,87]
for x in SB_cases:
    print("Case:␣", x)
    response = requests.get( url= "http://localhost:8080/
        retrievalByIDWithContent.json?casebase=
        cb_speedskaters&concept=speedskaters&amalgamation%20
        function=sb&caseID=speedskaters"+str(x)+"&no%20of%20
        returned%20cases=26")
    response_json = response.json()
    df_response = pd.DataFrame(response_json)
    updated_response = df_response.iloc[1:26, : ]

    updated_response.replace('_unknown_', np.nan,  inplace=
        True)
    filtered = updated_response.dropna(subset=['SB'], how='
        all')

    query = df_response.head(1)['SB'][0]
    retrieved_masl = float((df_response.head(1)['SB_masl'
        ][0]))
    sb_altitude = np.array(filtered['SB_masl'])
    average_SB_altitude = np.average(sb_altitude.astype(int
        ))

    #Mean
    mean_prediction = mean(float(query), updated_response,
        'SB')
    corrected_mean_prediction = altitude(mean_prediction,
        retrieved_masl, average_SB_altitude.astype(int))

    #Median
    median_prediction = median(float(query),
        updated_response, 'SB')
    corrected_median_prediction = altitude(
        median_prediction, retrieved_masl,
        average_SB_altitude.astype(int))
```

Appendix B

Results

B.1 Result for nPB Approach

B.2 Results for SB Approach

Table B.1: nPB Results in seconds

Case	5	5	9	9	10	10	15	15	20	20	25	25
	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean
119	240,12	239,71	240,72	240,00	241,47	240,44	240,26	239,98	239,52	240,41	238,10	241,29
120	246,90	246,66	246,25	246,30	245,95	245,75	245,30	245,43	245,59	245,91	245,50	245,67
121	245,44	246,41	245,09	246,96	246,09	247,30	246,76	247,37	247,52	247,69	247,79	248,07
122	242,70	241,58	242,63	241,41	242,14	241,57	239,97	241,72	241,24	242,03	241,73	242,47
123	244,00	246,29	245,14	246,64	244,89	246,37	243,18	246,33	243,04	246,29	242,32	246,05
124	238,46	237,75	238,05	237,83	238,17	237,86	237,56	237,58	238,51	237,27	239,24	237,96
125	240,31	239,00	239,72	240,28	240,13	240,32	240,04	240,74	240,52	240,15	241,41	240,88
126	244,08	244,55	243,23	246,70	244,68	246,70	246,06	246,17	243,93	245,78	245,53	245,90
127	247,11	249,41	247,00	247,85	247,02	247,45	245,61	247,53	245,62	247,73	246,35	248,18
128	253,38	256,12	251,45	253,37	252,35	253,37	253,25	254,15	252,97	254,89	252,53	254,63
129	262,27	263,16	264,28	263,61	265,19	264,19	263,81	264,12	261,88	263,19	262,68	262,95
130	244,24	242,04	243,16	241,20	242,74	241,10	242,46	240,93	242,33	241,34	240,77	240,90
131	242,32	244,22	242,44	244,09	242,45	243,90	242,96	244,65	242,88	244,54	242,80	244,52
132	249,60	253,26	250,80	250,95	250,83	250,64	250,12	251,57	250,21	251,56	250,23	251,27

Table B.2: SB Results in seconds

Case	PB	SB	5	5	10	10	15	15	20	20
			Median	Mean	Median	Mean	Median	Mean	Median	Mean
69	237,7	239,47	238,20	238,75	238,28	238,80	238,64	239,50	239,37	239,23
70	237,09	237,09	233,63	233,35	233,66	232,76	233,66	232,76	233,62	233,73
78	243,79	245,09	238,87	238,87	241,25	238,99	240,78	238,97	239,64	238,17
83	232,02	237,84	237,95	237,32	238,00	237,66	238,23	236,93	238,62	236,62
85	235,73	239,54	238,00	236,68	236,63	236,47	237,88	236,57	237,75	236,82
91	244,35	245,66	243,40	243,66	242,67	242,70	242,28	242,79	242,28	242,79
97	245,35	245,35	246,81	246,81	244,96	244,81	245,51	244,41	245,68	244,66
101	247,54	247,54	249,44	249,44	246,61	247,22	250,77	247,70	250,48	247,65
109	247,31	251,63	241,88	241,66	245,73	244,44	245,73	244,44	247,32	244,93
110	244,89	245,1	247,99	247,93	248,29	246,82	248,19	246,84	248,19	246,84
113	247,53	253,79	242,86	243,05	242,78	244,59	245,38	244,96	245,65	244,57
87	241,25	243,89	247,14	246,99	246,68	246,19	245,73	245,53	243,48	244,78

Appendix C

Data Set

The data set will be attached as a CSV file

Olof Andreas Bergman

NTNU
Norwegian University of
Science and Technology