NTNU
Norwegian University of
Science and Technology

CBR for road conditions on Norwegian alpine roads

Mikael Kolkinn

Master of Science in Computer Science
Submission date: January 2017
Supervisor: Jo Skjermo, IDI

Norwegian University of Science and Technology
Department of Computer Science

Problem description

The aim of this project is to study the use of methods such as KNN and case-based reasoning as part of a decision-support system for winter road operation. The project will use findings from earlier work in CBR for winter road operation at Dovrefjell, that will serve as the foundation of continued study on the subject. The project will focus on improvements with regards to case representation, similarity measure, classification, case pruning, refinement of case base or case initialization. New approaches and algorithmic changes should be investigated by building a demo system, or extend earlier systems.

Sammendrag

Formålet med dette prosjektet var å undersøke mulige forbedringer til og generaliseringen av et eksisterende CBR-system for veiforhold. Resultatene indikerer en forbedring i ytelsen ved bruk av k-nærmeste nabo sammen med et eksisterende beslutningsavhengig similaritetsmål. Den generaliserte versjonen av CBR-systemt virker å yte godt, og testene viser potensiale for bruk av systemet på andre fjelloverganger. En overvekt av tilfeller der veien er åpen, i motsetning til stengt, ser ut til å påvirke systemet slik at det oftere foreslår å holde veien åpen. Dette gir også utfordringer for de testede metodene for vedlikehold av kunnskapsbasen. Det gjenstår fortsatt arbeid for å lage et fullstendig CBR-system.

CBR for road conditions on Norwegian alpine roads

Mikael Kolkinn

NTNU
mikaelkolkinn@gmail.com

Abstract. This study's objective was to investigate various improvements to and the generalization of an existing CBR system for road conditions. Results indicate a slight performance improvement from using k-nearest neighbors together with a previously designed decision dependent similarity measure. The generalized version of the CBR system appears to perform well, and tests show potential for using the system on different alpine roads. An overweight of cases where the road is open as opposed to closed, seems to introduce a bias towards predicting that the road should be open. This is also a challenge for the methods for tuning and maintenance tested in the study. Further work remains with respect to creating a full blown CBR system.

1 Introduction

The winter operation of alpine roads is very dependent on the weather and its effect on the road conditions. Multiple factors like wind direction and wind speed, snow, ice and water on the road surface, snow fall, drifting snow, visibility and temperature, all affect the road conditions. Determining when it is necessary to close the road or to use convoy driving is a difficult problem. Operators of such road segments need to maintain safe operation of these roads, while maximizing the time a road is operational. In a study on the stretch of road used in this project [1], predicting the closing of the road was found to be challenging, and it was concluded that no single parameter alone give grounds for closing the road. According to another study [2] expert systems can help road maintenance managers by reducing stress, anxiety and guilt when there is doubt about which measures should be made and where the wrong decisions lead to accidents.

Previous work on this project [3, 4], resulted in a system using machine-learning that used data from multiple weather sensors, to predict whether the road should be closed, open or if convoy driving should be used. The system was created for use on the Norwegian alpine road E6 going over the Dovrefjell mountain range. It was based on knowledge from local contractors and historical data from previous closings of the road. In this paper, we continue the work on this system and divide this work into three parts.

First, we look at the measures of similarity used in the system, and assess the performance impact of small modifications to these similarity measures. The existing system used a special way of dealing with similarity that was believed to perform well, but was never really tested against a simpler version of the similarity measure. We therefore test the special similarity measure and a simpler version of it to see how they compare. We also test the performance impact from having the machine learning system base it's prediction on multiple of the closest matching historical events, as opposed the existing scheme of using just the single closest matching event to generate a prediction.

Secondly, we work on generalizing the system so that it is more suited for use on different alpine roads. The existing system was developed specifically for the Dovrefjell alpine road and it's three roadside weather stations. It was designed in such a way that the knowledge inside the system could not easily be transferred to other alpine roads. Nor did the design allow immediate use of new weather stations at Dovrefjell in the future. To make use of a new weather station, the system would require historical readings from that specific weather station, correlated with logs of road closings or convoy driving at the alpine road. The same thing would apply to another alpine road, meaning that knowledge from the Dovrefjell alpine road could not be used on a different alpine road. We propose a new, more general design that hopefully will allow immediate benefit from adding new weather stations along the Dovrefjell alpine road and make it possible to reuse the knowledge from Dovrefjell at other alpine roads.

Thirdly, we look at methods for tuning and maintenance of the machine learning system's knowledge base. Here we address a potential error introduced by the generalization of the system. We also test a method for removing redundancy in the knowledge base, reducing the size of the knowledge base, potentially shortening the system's execution time.

2 Related work

With regards to winter road operation, there have been a few studies on the use of artificial intelligence to estimate road surface conditions. [5, 6] focused on the prediction of road temperatures, frost and ice on the road and slippery road conditions. In [7, 8] this concept was taken further by developing a Maintenance Decision Support System (MDSS) that was also able to suggest maintenance plans with actions like plowing and salting of the road at specific times in the future, including the amount of salt that should be used. Here the users would get prognoses for the effect of the maintenance plan, and could even make their own maintenance plan and see it's predicted effect.

Another weather-related field where artificial intelligence has been used to increase safety is snow avalanche prediction [9, 10]. Knowledge about the avalanche risk in an area could be used to close exposed roads, ski slopes etc., and to plan controlled triggering of avalanches. Weather data has also been used to find recommended speed limits on roads [11] and to improve traffic light regulation [12]. In [11] Fahmy used neural networks to recommend speed limits for cars and trucks, based on parameters like precipitation, wind, fog and visibility. The system was trained using measurements of the average speed of cars and trucks in different weather conditions.

For the road E6 over the Dovrefjell mountain range, our project started with a mobile application that gave recommended speed limits based on values that the user put in [1]. These values were visibility, wind speed and friction. The application based its calculations in requirements for the safe stopping length and grip of a vehicle, and used various formulas to calculate a recommended speed limit. Later a Decision Support System using Case-Based Reasoning (CBR) was developed [3, 4]. The system was modeled with expert knowledge from local contractors, and used historical weather data correlated with logs of road closings and convoy driving to determine if the road should be closed or not.

3 Method

The artificial intelligence system used our project is a Case-Based Reasoning system (CBR). Case-Based Reasoning is based on the hypothesis that similar problems have similar solutions. Unlike many other artificial intelligence systems, CBR is memory driven, reflecting the way humans use and adapt their previously encountered problems to solve new problems. It stores previous problems and their solutions together, so called cases, in a case base. When solving a new problem, the CBR system retrieves one or more similar cases, then attempts to reuse their solution, adapting it if necessary. The proposed solution is then evaluated, either by applying it to the problem or by assessment from a domain expert. The solution can then be revised if required to solve the problem, after which the problem description and its solution can be retained as a new case in the case base, effectively letting the CBR system learn how to solve the new problem. Aamodt and Plaza [13] introduced the classic model of the CBR cycle. It consists of four steps, Retrieve, Reuse, Revise and Retain, as seen in **Fig. 1**.

One of the advantages of CBR over most other artificial intelligence methods, is its ability to explain the underlying reason behind its proposed solution. This is done through presenting a similar problem and the solution to that problem. This helps the user to better understand and relate to the proposed solution.

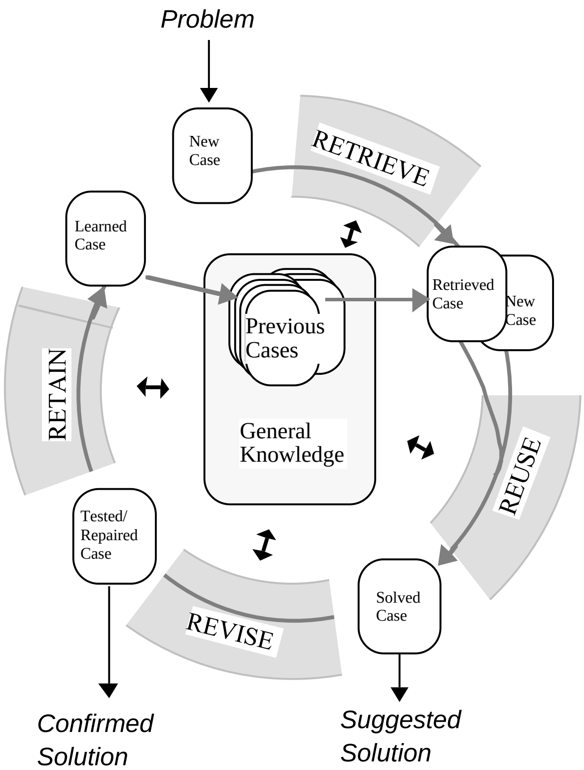

Fig. 1. The CBR cycle, from [13] with permission

A popular technique used in the retrieval phase of the CBR cycle is the k-Nearest Neighbor (kNN) algorithm [14]. It involves finding the k cases that are most similar to a new problem, and using the majority class of the k cases to classify the new problem. To find the similarity of two cases, we calculate the similarity of each attribute separately using specially designed similarity measures for these attributes. These attribute similarities are given individual weights and are summed up into a single similarity value for the two cases. The similarity value is normally a value between 0 and 1, where 0 means no similarity and 1 means completely similar.

$$Similarity(C,Q) = \sum_{i=1}^{n} fi(Ci,Qi) \times Wi \qquad (1)$$

Equation 1 shows how the similarity of two cases is calculated, where C is the existing case, Q is the query, n is the number of attributes from 1 to n, fi is a similarity function for attribute i in case C and Q and Wi is the weight for attribute i.

In this project, we use similarity weighted voting [15] instead of the majority class of the k cases. Similarity weighted voting works by weighting neighbors with a higher similarity more heavily than neighbors that are less similar. The class with the highest total similarity is chosen as the solution. This method has been shown to work well in [16].

3.1 Comparing similarity measures and k-values

As part of the continuation of the existing project at the Dovrefjell mountain road, we first wanted to draw attention to two small modifications that could potentially improve the performance of the CBR system. One of these modifications is to keep the similarity measure simple. As mentioned in chapter 1 Introduction, the existing CBR system [4], used a special similarity measure. With this similarity measure, the similarity of a query and an existing case depended on the decision in the existing case. If the decision of the existing case was "closed", then the similarity of attributes that were perceived as "worse" in the query than in the case would be 1. Likewise, if the decision was "open", then the similarity of attributes that were perceived as "better" in the query than in the case would be 1. Otherwise the specific similarity functions for each attribute would be used. Here "worse" and "better" were defined for a subset of the attributes used in the CBR system. An example of such an attribute is friction. A higher friction value is considered "better" and a lower friction value is considered "worse". For a more detailed description of the similarity measures used, see CBR-model 2 in [4].

The idea behind the decision dependent similarity measure was to cover a wider range of case variations with the limited amount of existing cases. With the narrow border between open and closed cases, another point of the similarity measure was for the similarity to only be lowered by attribute value changes that are likely to lead to a decision change. In other words, when the road is open and a weather attribute becomes "worse" or when the road is closed and a weather attribute gets "better".

The decision dependent similarity measure was thought to perform well, but was not really tested against the simpler, non-decision dependent similarity functions. We therefore test a simpler version of the similarity measure to check the validity of the decision dependent similarity measure and to see how the decision dependent version performs compared to the simple version. The simpler similarity measure uses the specific similarity measures for each attribute, defined for CBR-model 2 in [4], and does not depend on the decision in the existing case.

A second modification that could lead to better performance is the use of multiple of the most similar cases to generate a solution. The existing CBR system used the single nearest neighbor (1-NN) to find a solution to the query. To see if the nearest neighbor search performs better with other k-values, we performed tests with k-values 1, 3, 5, 7 and 9. Here the predicted decision is found through similarity weighted voting with the nearest cases, where each vote is based on the similarity of the case to the query. The decision with the highest total similarity is chosen as the solution.

3.2 Generalizing the case representation

In the existing case representation, detailed in [4], a case consisted of attributes from all the weather stations and a collective decision. This worked well enough, but was very specific to the alpine road E6 going over the Dovrefjell mountain range, with its three weather stations, at Avsjøen, Fokstugu and Hjerkinn. Due to this specificity, the existing approach had two main drawbacks.

Firstly, convoy driving or closing of the road often happens due to local conditions on parts of the stretch [1]. This implies that if the conditions at Fokstugu caused the road to be closed, then the readings from Avsjøen and Hjerkinn are uninteresting. With the existing case representation, two cases with similar conditions at Fokstugu being cause for closing, could have dissimilar conditions at the other weather stations, and therefore have a low overall similarity score. In this situation, it would be more interesting to see if the readings from a specific weather station are similar to a case where the conditions at that place caused the road to be closed or have convoy driving.

Secondly, knowledge from one weather station could not be used on another weather station, as the system used the combination of attributes from all the weather stations to inform its decision. Knowledge transfer across of weather stations is a feature that would be desirable when adding a new weather station to an existing system or when deploying the system to another alpine road. In the event of adding a new weather station to an existing system, the transfer of knowledge would enable the immediate use of the readings from the new weather station, in the CBR system. With the existing case representation, one would need to wait until enough new cases with readings from the new weather station were stored, before being able to utilize the new weather station. Because closing of the road is a rare event, it could take years until one could fully benefit from the new weather station. When deploying the system to another alpine road, it could make it unnecessary to process historical weather data for that road and logs of when the road was open, closed or had convoy driving. The system could therefore also be useful for an alpine road for which there exists no such historical data.

To address these issues, we propose a new, more general case representation where a case consists of attributes from only one weather station and a decision for the surrounding area of that weather station. We still use the same similarity measures as before, as described in CBR-model 2 in [4]. When converting from the existing case representation to the new one, we use the same decision for each case with the same timestamp. **Fig. 2** shows the case base after converting to the generalized case representation. The similarity of each case to another is reflected in the distance between them. Similar cases are close together and dissimilar cases are further apart. In the figure, different shapes represent different weather stations. An interesting observation here is that many cases from different weather stations are placed close together, meaning that they are quite similar. This strengthens the hypothesis that data transfer across of weather stations is possible.

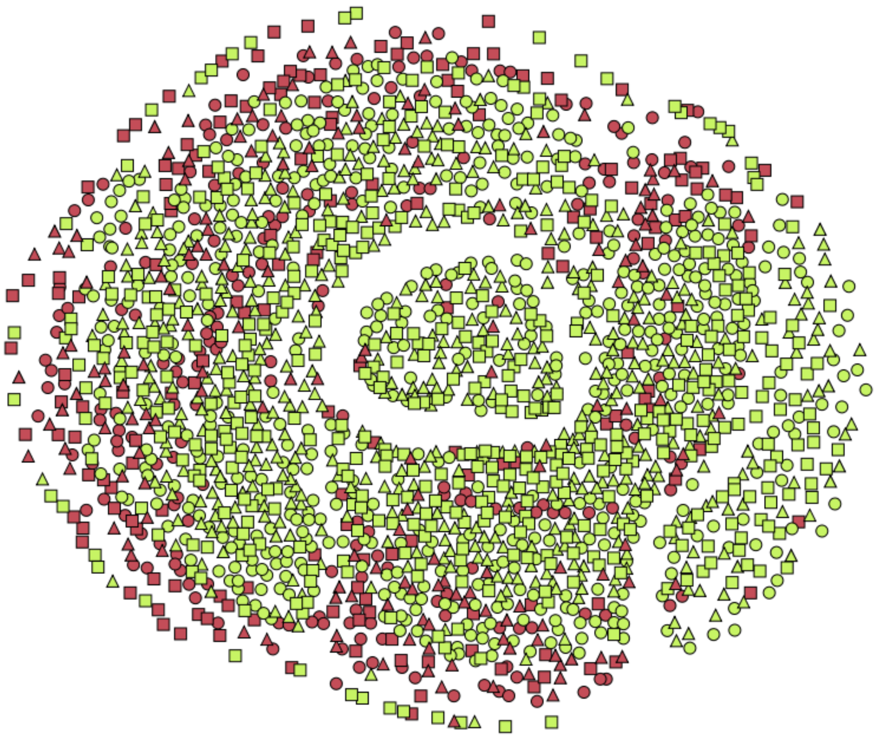

Fig. 2. Case visualization showing cases from 2011 to 2014 and their relative similarity. A case's similarity to another case is represented through the distance between the cases. Cases that are similar attract each other, and cases that are dissimilar repel each other. Red/dark grey points represent closed cases and green/light gray points represent open cases. The shape of the points represents their weather station: Avsjøen = circle, Fokstugu = triangle and Hjerkinn = square.

3.3 Tuning

In this chapter, we focus on ways to tune the CBR system, both in accuracy and reduction of the case base. We describe a potential error resulting from the conversion from the old to the generalized case representation, and a custom algorithm that we hoped would reduce this error. We also test a method for removing redundant cases from the case base. This is not an issue now, but redundant cases may lead to performance issues with regards to execution time in the future if the case base keeps growing.

As explained above the generalized case representation consists of attributes from only one weather station and a decision for the surrounding area of that weather station. A potential issue with this representation, is that for the road E6 over the Dovrefjell mountain range, there exist no historical data telling which parts of the road caused it to be closed at a certain point in time. The only data available is whether the road as a whole was open, closed or had convoy driving. As previously explained, we converted cases from the existing representation to the generalized representation by splitting the attributes by their originating weather station and using the same decision for each case with the same timestamp. This may however introduce erroneous decisions for cases where the local weather conditions of a case did not cause the road to be closed, but the road was closed because of weather conditions in another area or for different reasons. This is the situation only when the decision is closed or convoy. If the decision is open, then all locations are correct to indicate open. To fix these possible errors we propose a custom algorithm for removing erroneous cases from the case base. This algorithm works by first grouping cases together based on their location and time. A group consists of cases from one location, where each case in the group is closer than 24 hours to at least one other case in the group. After grouping the cases, each convoy or closed case in a group is tested against a CBR system with the cases from the other groups as its case base. If the predicted decision is open, the case is marked for removal. This process is repeated for all groups, and afterwards all the cases that are marked for removal get removed. The reason why the cases are grouped together based on time and location is to avoid a bias towards the timewise surrounding cases from the same location. These cases are in most circumstances only one minute or one hour apart, and do not always differ much from each other. Because they are close together in time, they are also likely to have the same decision. Without the grouping, the similarity-wise nearest neighbors of a case, would therefore often be the cases from the same location and time period. Since these cases are also likely to have the same decision as the querying case, the predicted decision would almost never differ from the decision of the querying case. The result would then be that most erroneous cases would not be removed. In **Fig. 2** we see single or minor groups of cases, closely surrounded by cases with a different decision, shown by their different colors. These cases may be erroneous cases as described above.

As mentioned in the beginning of this chapter, we also wanted to look at reducing the case base. While the case base in its current form is not large enough to generate performance issues with regards to retrieval time and memory consumption, this may become a problem in the future when more cases are added. We therefore test an algorithm, called Iterative Case Filtering (ICF) [17], for removing redundant cases from the case base. It is a commonly used algorithm and was chosen in this project because was easy to implement. ICF is a composite algorithm, featuring a noise reduction phase followed by a redundancy reduction phase. For noise reduction, ICF uses the Repeated Edited Nearest Neighbor algorithm (RENN) [18]. RENN considers a case as being noisy if it differs from the majority class of its k nearest neighbors. After the noise reduction phase, ICF aims to retain the border cases by removing only cases that are solved by more cases than they themselves solve.

4 Results

4.1 Experimental setup

Throughout the experiments described in this paper, we use datasets from the Norwegian alpine road E6 going over the Dovrefjell mountain range. These datasets come from weather data collected at the Norwegian Public Road Administration’s (NPRA) weather stations along the stretch correlated with logged occurrences of convoy driving or road closings. The NPRA has three weather stations along the stretch, placed at Avsjøen, Fokstugu and Hjerkinn. The data spans from November 2011 to March 2016, except from the winter of 2014/2015. During this time, the road has been closed 19 times. The data used are from the periods of time around occurrences of road closings, usually starting a couple of days before the road was closed and ending a couple of days after it was opened. From the winter of 2015/2016 this data is spaced at one minute intervals, whereas for the earlier data sets the data is spaced intervals of one hour. Overall the dataset contains 1,775 data points (cases).

We split the dataset in two, with roughly 70% as training set and 30% as test set. We created ten different splits of the data and we report all results as averages over the ten splits. Since the cases follow each other closely in time, the differences between one case and a case from the same location one hour later is sometimes small. To make sure that the test set is different from the training set, the cases are grouped together by their location and timestamp A group consists of cases from one location, where each case in the group is closer than 24 hours to at least one other case in the group. Each group are therefore at least 24 hours apart. These groups were then randomly placed in either the training set or the test set. 70% of the groups go in the training set and 30% go in the test set.

The performance of the CBR system is measured using **Equation 2**. This equation gives a balanced score from the system's ability to correctly predict both open and closed decisions. Unlike in the previous work on this project, we do not predict convoy driving. Convoy driving was found to be difficult to distinguish from open and closed cases, and neither of the previous systems performed well trying to predict it. Since the road technically is closed when convoy driving is employed, all cases of convoy driving will be regarded as closed cases in this paper.

For the new, generalized case representation, cases in the test with identical time stamps are grouped together and tested individually in sequence. If the predicted decision for one or more of these cases is closed, then the overall prediction for the group is closed. If the predictions for all the cases with the same time stamp is open, then the overall prediction for the group is open. This simulates the real world, where the whole road would have closed even if one sub segment of the road necessitated closing of the road. This is also the only way to get accurate results as we only have data showing when the whole road was closed, not which parts were cause for the closing of the road.

$$Performance = \frac{\frac{\#correct\_open}{\#total\_open} + \frac{\#correct\_closed}{\#total\_closed}}{2} \tag{2}$$

When using kNN, weights are found using evolutionary algorithms [19]. The evolutionary algorithm uses 70% of the training set as the case base and the other 30% of the training set to query against the case base. Two new splits are created and used on each generation of the evolutionary loop, and each individual's fitness value is its average performance on the two splits.

4.2 Comparing similarity measures and k-values

In the tests, the average score of the decision dependent similarity measure was 0.669 out of 1.0, and for the simple similarity measure the average score was 0.602, both using a k-value of 1. Looking at the accuracy for open and closed cases separately, the difference between the two similarity measures is slightly higher. From **Table 1** and **Table 2** we see that the decision dependent similarity measure correctly predicts 82 % of the open cases, and the simple similarity measure 84 %, with a difference of 2 % in favor of the simple version, also with a k-value of 1. For closed cases, the decision dependent version correctly predicts 51 % of the cases and the simple version 36 %, resulting in a 15 % difference between two similarity measures. **Table 3** and **Table 4** show the results for the decision dependent and simple similarity measure respectively. The highest scoring k-value for the two similarity measures varies with the case split they are being tested on. On average the decision dependent similarity measure scores highest using k=5, with an average score of 0.701 compared to 0.669 using k=1. The average score for the simple similarity measure is also highest using k=5, with an average score of 0.62 compared to 0.602 using k=1. Compared to using k=1, k=5 gives a score increase of approximately 4.8 % for the decision dependent similarity measure and approximately 3.0 % for the simple similarity measure. Although there was a slight gain in performance using k=5, it seems that the use of k-NN does not make that much of a difference for either the decision dependent or simple similarity measure. The difference between the worst and best performing k-value, is 0.032 for the decision dependent version and 0.018 for the simple version.

Table 1. Shows the prediction accuracy of open and closed cases, for the decision dependent similarity measure. The percentages are averages from the 10 case splits.

K-value	Correct Open	Correct Closed
1	82 %	51 %
3	82 %	56 %
5	82 %	58 %
7	80 %	59 %
9	81 %	55 %

Table 2. Shows the prediction accuracy of open and closed cases, for the simple similarity measure. The percentages are averages from the 10 case splits.

K-value	Correct Open	Correct Closed
1	84 %	36 %
3	84 %	39 %
5	85 %	39 %
7	87 %	35 %
9	83 %	40 %

Table 3. Test results for the decision dependent similarity measure, using the existing case representation

K-value	1	3	5	7	9
Average	0.669	0.691	0.701	0.696	0.692

Table 4. Test results for the simple similarity measure, using the existing case representation

K-value	1	3	5	7	9
Average	0.602	0.617	0.620	0.611	0.607

4.3 Case generalization

For the generalized case representation, the tests returned a highest score of 0.709, as shown in **Table 5** using a k-value of 3. This score is only slightly higher than the best score measured using the existing case representation using the decision dependent similarity measure, but there is a shift when looking at the percentage of correctly predicted open and closed cases, as seen in **Table 6**. There the generalized case representation with k=3 correctly predicts 5 % less open cases and 7 % more closed cases than the existing case representation with k=5. The generalized case representation seems to be more sensitive to differing k-values, compared to the existing case representation. From the results in **Table 5** and **Table 6**, we see large fluctuations in the system's ability to correctly predict open and closed cases, with the correct prediction of 84 % of the closed cases using k=1 and 28 % using k=9. With regards to the overall scores taking the correct prediction of both open and closed cases into account, the fluctuation is smaller with a difference of 0.098. This is probably due to the binary nature of the solution space (open and closed), where a decrease in the correct prediction of open cases lead to an increase in the prediction of closed cases and vice versa.

Table 5. Test results using the generalized case representation

K-value	1	3	5	7	9
Average	0.674	0.709	0.693	0.639	0.611

Table 6. Shows the prediction accuracy of open and closed cases, using the generalized case representation. The percentages are averages from the 10 case splits.

K-value	Correct Open	Correct Closed
1	51 %	84 %
3	77 %	65 %
5	85 %	54 %
7	91 %	37 %
9	94 %	28 %

We also tested the system's ability to use cases from different weather stations than that of the querying case to predict the decision at the time of the query. This was done by ignoring cases from the same weather station as that of the query when generating the prediction. From **Table 7** and **Table 8** we see that the results from this test are surprisingly similar the results from the previous test, and the best score here was also 0.709 like in the previous test.

Table 7. Test results using the generalized case representation, where cases from the same weather station as that of the query have been ignored during testing

K-value	1	3	5	7	9
Average	0,651	0,709	0,676	0,649	0,623

Table 8. Shows the prediction accuracy of open and closed cases, using the generalized case representation, where cases from the same weather station as that of the query have been ignored during testing. The percentages are averages from the 10 case splits.

K-value	Correct Open	Correct Closed
1	51 %	79 %
3	78 %	64 %
5	86 %	49 %
7	91 %	38 %
9	94 %	31 %

4.4 Tuning

While testing the custom algorithm for removing "false positives", several strategies were explored for weighting the k-NN retrieval. The best results were gained from running the custom “false positive” removal algorithm without weighting attributes differently, and then using evolutionary algorithms to generate weights for the CBR retrieval. On average the scores were however lower when using the algorithm than without using it. The algorithm tended to remove too many closed cases, decreasing the system’s ability to correctly predict closed cases to around 24 % for all k-values. In **Fig. 3** we visualize the case base after removing “false positives”, using k=3 and equal weights for all attributes. By comparing it to the original case base, visualized in **Fig. 2**, we see that the closed cases now lie almost exclusively in the outer region of the graph and that most of the closed cases that previously lied amidst open cases inside the outer circle have been removed. The intention of the algorithm was to remove only the single or minor groups of closed cases closely surrounded by open cases. From looking at the visualization it does however seem that larger groups of closed cases have also been removed. This is likely because they were from the same location and time period. As we recall, the algorithm ignored cases from the same location and time period when marking cases for removal. When doing this, the larger groups of closed cases would seem smaller to the algorithm due to the exclusion of cases from the same location and time period.

The ICF algorithm for redundancy reduction also seemed to struggle with the overweight of open cases in the case base. For all k-values except k=1, the noise reduction phase using RENN caused a bias in the CBR system towards open cases, effectively making it unable to predict closed cases. For k=1 however, the average score was 0.676, which is 0.002 higher than the original score of 0.674 using k=1. The RENN algorithm works by removing cases that do not agree with the majority of the nearest neighbors, and keeps doing so until this condition is no longer satisfied. Since the majority of the cases in the case base are open, it is likely that the algorithm becomes biased and end up removing too many closed cases. Due to time constraints, the ICF algorithm has not been tested without a noise removal phase, but this could be an interesting test to do in the future, as it could potentially balance the amount of open and closed cases in the case base.

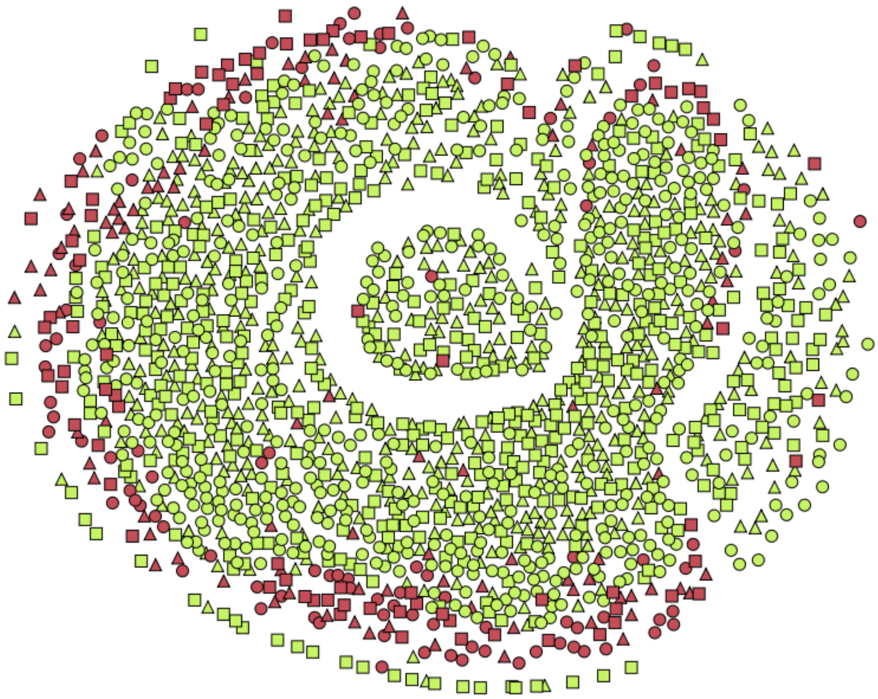

Fig. 3. Case visualization showing cases from 2011 to 2014, after using the custom algorithm intended to remove “false positives”, and their relative similarity. A case’s similarity to another case is represented through the distance between the cases. Cases that are similar attract each other, and cases that are dissimilar repel each other. Red/dark grey points represent closed cases and green/light gray points represent open cases. The shape of the points represents their weather station: Avsjøen = circle, Fokstugu = triangle and Hjerkinn = square.

5 Discussion and Future Work

For the machine learning part of the CBR system, we rely on logs of road closings that indicate when the road operators closed and opened the road. This means that the system is taught to predict when the operators would open or close the road. One issue with this approach, is that an operator’s decision may differ from time to time, even if theoretically given the exact same road conditions. There may be delays from when the road should have been closed to when it was closed, because the operator needed to drive up to the mountain to confirm their suspicion. As indicated described in [1], the time indicated in the logs may also be slightly off for various reasons. All this can lead to conflicting cases, making it difficult for the CBR system to learn correctly.

In the present and previous implementations, evolutionary algorithms were used to generate k-NN weights. For the existing case representation, these weights varied greatly with the case split and k-value that was being tested. For the new and more general case representation however, these weights remained more similar. Weights that worked well with one case split or k-value tended to work well with the other case splits and k-values as well. The weights also worked well across of different weather stations. This may indicate that the generalized case representation does not need weights that are tailored to local differences in topology, which is promising with regards to using the system at another alpine road.

As noted in chapter 4.4 Tuning, the case base has an overweight of open cases, which may lead to a bias towards predicting “open” more often than it otherwise would. This can also be seen in the other test results, where the system usually is better at predicting open cases rather than closed cases. The maintenance methods tested in this paper also seemed to struggle with the ratio of open to closed cases. It could be useful to try different means of balancing the amount of open to closed cases, to see if that makes the methods tested here perform better. It could also be that other perhaps more conservative methods for tuning system work better with our case base.

At the current stage, the CBR-system is designed to use real time data from the three weather stations along the road and give predictions based on this data. It would be interesting to attempt forecasting the values for the different attributes used in the system, and through that predict when the road should be closed or opened, minutes or possibly hours into the future.

Up until now the focus of this project has been on the retrieval phase of the CBR cycle. It would be natural to invest further work in other parts of the CBR cycle. Especially the retain phase, as the system currently has no functionality for adding new cases.

6 Bibliography

1. Engen T, Skjermo J, Opland R (2015) Vind- og friksjonsvarsling på E6 Dovrefjell.
2. Ljungberg M (2002) Expert System for Preventive Salting Operations on Winter Roads. Licentiate Thesis. Trita-Vt Fr xviii, 152.
3. Skjermo J, Dahl E, Opland R, et al (2016) Case-based Reasoning for Alpine Road Operation Support. ITS World Congr 23:10–14.
4. Gustafsson E (2016) CBR for Winter Road Operation at Dovrefjell. NTNU
5. Shao J (1998) Application of an artificial neural network to improve short-term road ice forecasts. Expert Syst Appl 14:471–482. doi: 10.1016/S0957-4174(98)00006-2
6. Gustavsson T, Bogren J (2007) Information not data: Future development of road weather information systems. Geogr Ann Ser A Phys Geogr 89 A:263–271. doi: 10.1111/j.1468-0459.2007.00325.x
7. Mahoney W, Myers W (2003) Predicting Weather and Road Conditions: Integrated Decision-Support Tool for Winter Road-Maintenance Operations. Transp Res Rec J Transp Res Board 1824:98–105. doi: 10.3141/1824-11
8. Mahoney III W, Bernstein B, Wolff J, et al (2005) FHWA's Maintenance Decision Support System Project: Results and Recommendations. Transp Res Rec 1911:133–142. doi: 10.3141/1911-13
9. Gassner M, Brabec B (2002) Nearest neighbour models for local and regional avalanche forecasting. Nat Hazards Earth Syst Sci 2:247–253. doi: 10.5194/nhess-2-247-2002
10. Möhle S, Bründl M, Beierle C (2014) Modeling a system for decision support in snow avalanche warning using balanced random forest and weighted random forest. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 8722:80–91.
11. Fahmy MMM (2008) Neural Network Approach to variable vehicle speed limitation upon weather conditions. In: 2008 Int. Conf. Comput. Eng. Syst. IEEE, pp 179–184
12. Kim J, Mahmassani HS, Hou T, Alfelor RM (2014) Development of real-time simulation-based decision support system for weather responsive traffic signal operations. In: 17th Int. IEEE Conf. Intell. Transp. Syst. IEEE, pp 810–815
13. Aamodt A, Plaza E (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun 7:39–59. doi: 10.1.1.56.4481
14. Fix E, Hodges JL (1989) Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. Int Stat Rev / Rev Int Stat 57:238. doi: 10.2307/1403797

15. Dudani SA (1976) The Distance-Weighted k-Nearest-Neighbor Rule. IEEE Trans Syst Man Cybern SMC-6:325–327. doi: 10.1109/TSMC.1976.5408784
16. Zavrel J (1997) An empirical re-examination of weighted voting for k-nn. Proc 7th Belgian-Dutch Conf Mach Learn 139–148.
17. Brighton H, Mellish C (1999) On the consistency of information filters for lazy learning algorithms. Princ Data Min Knowl Discov 283–288. doi: 10.1007/978-3-540-48247-5
18. Tomek I (1976) An Experiment with the Edited Nearest-Neighbor Rule. IEEE Trans Syst Man Cybern 6:448–452. doi: 10.1109/TSMC.1976.4309523
19. Bäck T (1996) Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York