Forest Cover Classification

David Nicolay
Department of Computer Science
Stellenbosch University
Stellenbosch, South Africa
26296918@sun.ac.za

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

introooo TODO remove "Fig. 1", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

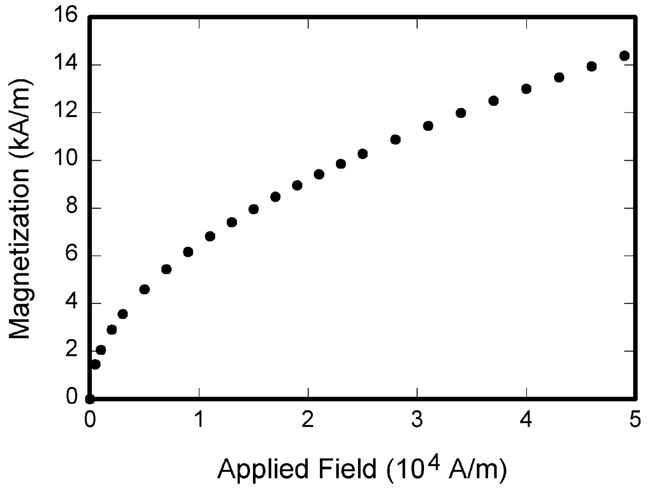

Fig. 1. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

II. Introduction

III. Background

A. k-nearest Neighbours (kNN)

The nearest neighbour algorithm is an example of instance-based learning where the prediction for an instance is made by comparing the instance to similar training instances. This is a lazy learning approach since we delay processing training data until prediction is needed. Since kNN is also similarity-based, it is non-parametric, which means that it does not learn a mapping from the input space to the output space.

The k-nearest neighbour algorithm is presented in Algorithm 1 which explains that we calculate the distance from each data point in our training dataset to the distance. From there, we choose the closed k observations and perform inference. In a classification context we may perform majority voting to determine the class, in a regression context we use the mean or median of the k nearest observations.

In order to use the algorithm we need to define a value of k which decides the number of nearest neighbour instances to consider when predicting. Choosing a value a low value for k results in a high variance, but low bias leading to an unstable model which tends to overfit. Choosing a large value for k results in a smaller variance, but high bias, leading to a more stable model which tending to underfit. Measures such as k-fold cross validation assist in the selection of a value of k.

Algorithm 1 k-Nearest Neighbors Algorithm

```
function KNN($D, x, k$)
    for all $x' \in D$ do
        $d = \text{distance}(x, x')$
    end for
    sort($d$)
    $S$ = set of $k$ patterns in $D$ closest to $x$
    return class as majority class in $S$
end function
```

Notation: D denotes the training dataset, x is the query instance, k is the number of nearest neighbours, and distance$(\cdot)$ is a distance metric (e.g., Euclidean distance).

B. Classification Trees

Classification trees represent a fundamental approach in supervised machine learning for predictive modelling, where the learned model forms a hierarchical tree structure with non-terminal nodes representing decisions on descriptive features and terminal leaf nodes representing target feature predictions [1]. A classification tree is a type of decision tree where leaf nodes represent different discrete classes.

Classification trees employ recursive partitioning where a training dataset is systematically divided into increasingly homogeneous subsets based on feature values. It uses a greedy strategy to select the feature to partition upon at each split that maximises information gain [2]. The information gain criterion, derived from Shannon's entropy measure, quantifies the reduction in uncertainty achieved by partitioning the data according to a specific feature test. Formally, given a dataset D with M different classes, the entropy is calculated as:

$$H(D) = -\sum_{m=1}^{M} p(y_m) \log_M p(y_m)$$

where $p(y_m)$ represents the probability of class y_m occurring in D. When the dataset is partitioned on feature α into O outcomes, the information gain is computed as:

$$\text{gain}(\alpha) = H(D) - H_\alpha(D)$$

where

$$H_\alpha(D) = \sum_{o=1}^{O} p_o\, H(D_o)$$

represents the weighted entropy after the split.

Since we recursively partition until either a stopping condition is satisfied or all subsets are homogeneous, classification trees inherently lead to overfitting, as the algorithm attempts to perfectly classify all training instances, often creating overly complex trees that capture noise rather than underlying patterns [1]. To remedy this, post-pruning techniques are used to remove branches that do not improve generalisation performance.

Classification trees can naturally handle both categorical and numerical features. Classification tree induction algorithms can cope with missing values. Unfortunately, classification trees exhibit an axis-aligned bias which restricts decision boundaries to be parallel to feature axes.

C. Expectations With Respect to Data Quality Issues

Since kNN relies heavily on distance metrics, it will be severely impacted by missing values since it cannot compute meaningful distances between points. This can however be handled by ignoring the missing values in the distance calculation and scaling up the weight of the non-missing descriptive features. The kNN algorithm can actually be used as an imputer itself as well. If nothing is done the algorithm will either fail or give biased distances. Classification trees on the other hand handle missing values more gracefully. Classification trees can adjust the gain ratio calculation

$$\text{gainRatio}(x) = (1 - F) \times \big(H(D) - H_x(D)\big)$$

where F is the fraction of missing patterns.

With regards to the Facet feature being correlated with the Aspect feature, kNN will be moderately impacted since the distance calculation used will overweight certain areas of similarity. There is no built in mechanism to handle feature correlation as a data issue. Classification trees will be minimally impacted since trees naturally handle correlated features through the feature selection process involving the gain ratio/information gain. This means that the classification tree induction will likely select one feature and ignore the correlated redundant one.

Concerning the Inclination feature only containing noisy values, the impact on the kNN algorithm will depend on the value of *k* since higher values are more sensitive to noise and lower values are more robust to noise. However, since there are many predictive features the value won't dominate the distance calculation. Classification trees are robust to noise and will handle it through pruning. The noisy patterns will end up in small leaf nodes as the model overfits and post-pruning removes these leaves. The noise becomes a minority and does not affect majority class prediction.

Since kNN employs distance measures which are often sensitive to outliers since the distance can be seriously skewed even if only one feature is outlying. For small values of *k* outliers are unlikely to be nearest neighbours and will therefore have limited influence. For large values of *k* the outliers will also have limited influence due to majority voting. Cumulatively, outliers will have a moderate impact on the kNN algorithm. Classification trees are robust to outliers since they handle them in a similar way to noise. Outliers are first isolated in small leaf nodes during induction. Pruning removes leaves which contain outliers which results in outliers becoming a minority in combined larger subsets. This makes decision trees robust to outliers.

The kNN algorithm will be severely impacted from features with numeric ranges that differ significantly from one another since features with large ranges dominate distance calculations. For example, if one feature takes on the value of 0 or 1 and another takes on the value of 0 or 100 000, the second will overwhelm the distance calculation. Without normalisation, the kNN algorithm will be unreliable. Classification trees have a natural robustness to different feature scales and therefore will be minimally impacted. This stems from trees using individual feature values for splitting thresholds.

The mixed data types will affect the kNN algorithm since it requires special handling for mixed types such as using the Gower's similarity with different similarity measures for different types. However, in this dataset since there are two categorical variables with the first encoded over four columns and the second over 40 and 13 numeric features there can be a distance bias toward the numeric features. Euclidean distance assumes features are continuous, but the categorical features are encoded as zero or one. Classification trees naturally handle mixed data types since they directly compare categorical features against each other for splitting, and the splitting is simply decided off information gain.

The impact of the Water Level feature having a cardinality of one will be none at all for kNN. It will only contribute the the distance calculation unnecessarily and be a computational waste. Since all observations have the same value, all the distance measures will be equally skewed, resulting in no impact at all. The classification tree will never select this feature during tree indication since there is zero information gain because all values are identical. Therefore, classification trees naturally handle this issue and it will have no impact.

The Observation ID unique feature will have a sever impact on kNN since each instance appears maximally different from all others on this feature. The problem is that this dominates distance calculations making all instances seem dissimilar. It breaks the fundamental assumption of similarity-based learning. Classification trees will overfit severely since each unique value would create a separate branch. The information gain maximisation favours features with many outcomes so the result will be an extremely busy tree with poor generalisation.

The skewed class distribution impact on kNN depends on the value of k, since a smaller k is more robust to class imbalance whilst a large k results in the majority class dominating the predictions. Methods such as the Synthetic Minority Oversampling Technique (SMOTE) must be considered. The impact of classification trees is moderate since they are sensitive to skewed distributions. The minority class instances result in small leaves which are likely to be pruned, then the combined instances will be classification as a majority class. This results in poor minority class recognition.

IV. Methodology

V. Empirical Procedure

A. Data Pre-processing

In this section we discuss the data pre-processing steps applied to the dataset prior to implementing the k-nearest neighbour and classification tree algorithms.

1) k-Nearest Neighbour Data Pre-processing: First, the unique identifier feature `Observation_ID` is removed because unique values make all instances appear maximally different. This breaks the similarity-based learning assumptions and leads to the feature dominating distance calculations. The feature has no signal that can provide the model with any benefit.

Second, we removed the constant feature `Water_Level` since it provides zero discriminative information but adds computational overhead to all the distance calculations. We can be certain that the model performance will not change based on the fact that the feature has a cardinality of one.

Third, we normalised the numeric features by applying linear scaling to ensure the numeric features are in the same range being between zero and one. Features with large ranges will simply dominate distance calculations over smaller range features. This would make the model unreliable and heavily weight features with higher values. Specifically, `Elevation`, `Aspect`, `Slope`, `Horizontal_Distance_To_Hydrology`, `Vertical_Distance_To_Hydrology`, `Horizontal_Distance_To_Roadways`, `Hillshade_9am`, `Hillshade_Noon`, `Hillshade_3pm` are all normalised.

Fourth, we handled the missing values by using kNN imputation with a k of five. There are only 298 observations that are missing the `Slope` feature. For each of these we computed the Euclidean distance between the observation and all others based on available features, then identified the k closest rows (neighbours) that have the missing feature value. The mean of these values were calculated and imputed to the missing observation.

Fifth, we addressed the feature correlation issue by removing the `Facet` feature since it is just providing redundant information. Correlated features can overweight certain aspects in distance calculations resulting in worse model performance. Essentially, the `Aspect` feature would have double the influence on the model as the other features if `Facet` was not removed.

Sixth, the mixed data types issue was handled by converting the one categorical variable `Soil_Type1` to numeric. This was achieved by simply replacing all the instances with a *positive* value with the number one, and the *negative* instances with zero. This approach was chosen since it aligned the feature with the other soil type features which all take on the values of either one or zero.

Seventh, the features containing outliers were left untouched. The kNN algorithm has moderate sensitivity to feature-based outliers, but a small k value provides natural protection.

Eighth, SMOTE was used to improve the class imbalance. Since the five minority classes had significantly fewer observations than the majority two classes, we employed SMOTE to increase the number of minority class observations to 20% of the majority class. This partial balancing approach was used since the dataset only increased in size by 34%, as opposed to the dataset exploding in size if they were all were made equal. For example, if we over-sample class 4 to 283,301 we'd create 280 000 synthetic samples for one class. This is massive and risky for overfitting. On top of this Tomek links for majority class under-sampling was also used to reduce the number of observations in the majority classes. Using this under-sampling technique alone would not be sufficient since only a small fraction of the majority samples are removed. The Tomek link technique resulted in the majority classes reducing by roughly 60 000 observations in each majority class. That is, Class 1 now has 158,427 samples, and Class 2 now has 225,918 samples after reduction.

Ninth, the noisy feature `Inclination` was removed since it was confirmed to contain only noise. Noisy features affect distance calculations, although larger k values provide some robustness it is safer to remove it since we know it contains no signal.

2) Classification Trees Data Pre-processing:

B. Control Parameter Tuning

C. Performance Metrics

D. Analysis Process

VI. Research Results

VII. Conclusion

References

References

[1] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[2] J. Kelleher, B. Namee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics, second edition: Algorithms, Worked Examples, and Case Studies*. MIT Press, 2020. [Online]. Available: https://books.google.co.za/books?id=wtGMEAAAQBAJ