Evaluating Basic Machine Learning Models for Detecting Fine-scale Water Streamlines

Sreya Aluri (Mentor: Dr. Zhe Jiang)

Abstract

Having an accurate representation of surface water sources is very important in various fields ranging from predicting climate changes to understanding agricultural landscape. One surface water feature which is not very easy to detect is fine-scale water streamlines. In my project I explore ways of automating this process using basic machine learning models. My findings show that this is a very complex problem that cannot fully be solved using basic models and my results hint at future directions of exploration.

1. Introduction

Detection of fine-scale water streamlines is important because they are surface water resources which affect the geography and climatology of an area. This plays a big role in how scientists understand a wide range of things from environmental monitoring to flood mapping (Maidment, 2016). However, it is really difficult to detect these streamlines without directly seeing them on site. Hence, finding a way of easily doing so with a high success rate would drastically improve the surface water data available to scientists. In my project, I try to automate this detection process using basic machine learning algorithms. I used some of the most commonly used binary classification algorithms to train various models and evaluated them individually and against each other.

2. Dataset Description

The data used in this project is based on the data used by Xu et al. (2021). It is taken from the National Hydrography Dataset which stores geospatial datasets representing the surface water features in the United States. The study area is a watershed in Rowan County, North Carolina and the corresponding geospatial data is stored in five bands: red, green, blue, infrared, and elevation. Each pixel in our data covers 1 m^2 of area. The number of training and testing pixels are listed in Table 1.

Table 1: Dataset Breakdown

Training Set		Testing Set	
Land	Stream (2.68%)	Land	Stream (4.41%)
18323774	504462	6376353	294415

3. Methods

Different models were trained and tested on the same training and validation datasets. The datasets consist of information pertaining to the red, green, blue, values corresponding to a

pixel in a true color imagery, along with infrared values and a digital elevation map. I ran this data through some common binary classification algorithms and got various models: a naive bayes classifier, support vector machine, and an artificial neural network.

3.1. Naive Bayes Classifier

Naive Bayes is an algorithm which is based on applying the Bayes' theorem from probability theory (Price, 1763). While a lot of algorithms try to understand how various features might relate to each other, this one assumes a conditional independence between every pair of features (Berrar, 2019). There are three main types of naive bayes models based on the distribution of the data: gaussian, multinomial, and bernoulli. For the data used in this project, I picked the gaussian distribution model as it assumes that the features follow a normal distribution which makes sense for color values, infrared values, and elevation values.

3.2. Support Vector Machine

The support vector machine algorithm is based on finding a hyperplane in the feature space which distinctly classifies data points (Cortes & Vapnik, 1995). This hyperplane will be the decision boundary. The crucial hyperparameters that govern this decision boundary are the regularization parameter and gamma. The regularization parameter defines how much you want to avoid miss classifying the date, i.e. how strict the decision boundary should be. Gamma defines how far the influence of a data point goes, i.e. how far can a data point be from the decision boundary before it stops influencing it. For my data, the positive labels are very sparse so I got good results for a relatively high gamma value (120), and a small C value (20).

3.3. Artificial Neural Network

An artificial neural network is based on collections of nodes (neurons) that are interconnected in a structured manner and that work together to understand a problem and solve it (Maind & Wankar, 2014). The structure that is given to these neurons is in the form of layers of nodes which can take information, interpret it, and pass it on to the next layer. The way that information is passed through these nodes is what contributes to the final prediction. An activation function is the function used to define how information is passed on from one layer to another. I built a basic artificial neural network that uses supervised learning with two hidden layers and a sigmoid activation function since this is a binary classification problem.

4. Results

We are tackling a binary classification problem for identifying stream and non-stream pixels. From the data it is clear that there are much fewer stream pixels than there are non-stream pixels because we are trying to detect finescale water streamlines. This means that for a model to be tagged as a good model, it needs to minimize false negatives as much as possible, and maximize positive predictions. Hence, I chose precision, recall, and F1 score as the main metrics to evaluate the models on. The results of each model's performance are shown in Table 2 below.

Table 2: Precision, Recall, and F1 of the models

Model	Precision (in %)	Recall (in %)	F1 (in %)
Naive Bayes	0.23	0.05	0.08
Support Vector Machine	0.87	0.31	0.43
Artificial Neural Network	0.74	0.28	0.41

We see that Naive Bayes is just a bad model to use because it is, as the name suggests, naive and does not make good predictions for such complex problems. The Artificial Neural Network has a pretty good precision score, but the recall is bad. The Support Vector Machine is somewhat more promising as it does pretty well with precision and better with recall, although still low.

5. Conclusion and Future Work

The models I developed do not have good scores because no hyperparameter optimization algorithms were used. However, they do hint at some promising future directions. The Support Vector Machine seems to be doing the best from my data, and it can be made even better by implementing hyperparameter tuning algorithms like grid search (Shekar & Dagnew, 2019) or random search (Bergstra & Bengio, 2012), and by adding kernel optimizations. In a similar vein, even though the neural network I implemented did not perform very well, the performance could be improved using hyperparameter tuning algorithms, adding more hidden layers, or by using more complicated neural networks. Even if these performance improvements were to be implemented, there is a fundamental problem that still needs to be addressed. My current models take in tabulated data where each row is treated as independent from the other, i.e. they implement pixel-based classification. This type of classification does not take the relative positioning of the pixel into account which is a vital factor because if most neighbouring pixels are stream pixels, then the likelihood of the current one to be a stream pixel should be higher as streams are mostly continuous. Therefore, future work would include delving into convoluted neural networks that can work with images or other such deep learning algorithms.

References

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*, 281–305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

Berrar, D. (2019). Bayes' theorem and Naive Bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology*, 403–412. https://doi.org/10.1016/b978-0-12-809633-8.20473-1

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/bf00994018

Maidment, D. R. (2016). Conceptual framework for the National flood Interoperability Experiment. *JAWRA Journal of the American Water Resources Association*, *53*(2), 245–257. https://doi.org/10.1111/1752-1688.12474

Maind, S. B., & Wankar, P. (2014). Research Paper on Basic of Artificial Neural Network. *International Journal on Recent and Innovation Trends in Computing and Communication* , *2*(1), 96–100.

Price. (1763). LII. an Essay towards solving a problem in the doctrine of chances. by the LATE Rev. mr. Bayes, F. R. S. communicated by Mr. price, in a letter to John Canton, A. M. F. r. S. *Philosophical Transactions of the Royal Society of London*, *53*, 370–418. https://doi.org/10.1098/rstl.1763.0053

Shekar, B. H., & Dagnew, G. (2019). Grid search-based hyperparameter tuning and classification of microarray cancer data. *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*. https://doi.org/10.1109/icaccp.2019.8882943

Xu, Z., Wang, S., Stanislawski, L. V., Jiang, Z., Jaroenchai, N., Sainju, A. M., Shavers, E., Usery, E. L., Chen, L., Li, Z., & Su, B. (2021). An attention U-Net model for detection Of fine-scale Hydrologic streamlines. *Environmental Modelling & Software*, *140*, 104992. https://doi.org/10.1016/j.envsoft.2021.104992