UNIVERSITY OF VICTORIA

Department of Electrical and Computer Engineering

ECE 503 Optimization for Machine Learning

TECHNICAL REPORT

Jharna Kumari

V01046956

Abstract – Era of massive data has begun. There is a requirement for making use of data around us to gain insightful solutions to problems that can be solved using machines. A field in Artificial Intelligence, Machine Learning is the study of data and algorithms to make a machine learn to imitate human behaviour and make decisions on its own and gradually increase accuracy. An integral part of Machine Learning includes Optimisation. The concept includes machine learning models training on a certain amount of data to learn the relationship between features, eventually managing to predict new labels or output for an unseen input. Iterations of the training aim to improve the model's accuracy and reduce its margin of error as each iteration proceeds thus achieving optimisation. An important aspect of machine learning is mathematical optimization, which seeks to find the optimal parameters for a decision-making model based on available data. ML applications deal with a wide variety of data sizes, types and structures, as well as a range of performance evaluation criteria, but ML optimization problems generally fall into two general categories, unconstrained optimization and constrained optimization. In this paper, we shall deep dive into the realm of Machine Learning describing Optimisation problems. Then, discuss a problem that can be solved using commonly used optimisation methods. Next, we apply different methods to solve one problem and present the technical findings. Finally, address the challenges faced and offer recommendations for optimisation in Machine Learning.

Keywords – Machine Learning, optimisation, prediction model, optimal parameters, hyperparameter, tuning, maxima, classifier, hyperplane.

1. Introduction

Recently, popularity of Machine learning has grown tremendously. Data as a base as well as a by-product, keeps pouring in in different forms from various fields. Data collection is the process of gathering data for use in business decision-making, strategic planning, research, and other purposes[1]. For any machine learning research work, data collection is the very first step followed by making use of it effectively. One of the crucial parts of data analytics, effective data collection provides a foundation for deriving hidden information from the existing framework which later is used to predict or classify unknown future trends. In science, medicine, climatic, production, and other fields, typically, researchers create and implement measures to collect specific sets of data

as part of their data collection processes. Primarily there are two types of data – quantitative and qualitative data. Quantitative data is any data that can be counted or measured whereas qualitative data is descriptive mostly data that can be observed but not measured. In this report, we would be focusing on quantitative data which is numerical data. Data is a common component of ML techniques. There is no exaggeration when it is said that machine learning wouldn't exist without large scale data, since ML at its core is all about learning from data. As a logical first step, we should examine the specific types of data encountered in most ML applications as well as the associated learning methods known as supervised and unsupervised learning.

Training Data:

- **Supervised Learning –** Use of labeled data, where each data point has an associated label or target value. "Supervision" assists the model in understanding the relationship between inputs and outputs.
- **Unsupervised Learning:** Use of unlabeled data, where data points lack pre-defined labels or categories. The model is trained to discover hidden patterns and structures within the data itself.

Learning Goals:

- **Supervised Learning:** Main goal to predict or classify new data points based on the learned relationship between inputs and outputs from the labeled training data. Regression (predicting continuous values) and classification (assigning labels/categories) can be considered as some examples.
- **Unsupervised Learning**: Aims to discover insights from the data, such as deriving hidden patterns, clustering, or outliers. Examples include assemble (grouping similar data points), dimensionality reduction (compressing data while preserving key information), and anomaly detection (identifying exceptional data points).

Optimization Techniques:

- **Supervised Learning:** Optimizes the model to minimize the loss function, which measures the discrepancy between the model's predictions and the true labels. Common loss functions include mean squared error for regression and cross-entropy/softmax cost function for classification.
- **Unsupervised Learning:** Optimizes the model based on various criteria depending on the specific task. For example, clustering might optimize for maximizing inter-cluster distance and minimizing intra-cluster distance. Dimensionality reduction might optimize for preserving data variance while reducing dimensions.

Applications:

- **Supervised Learning:** Widely used in areas like spam detection, image recognition, recommendation systems, and financial forecasting.
- **Unsupervised Learning:** Commonly applied in market segmentation, customer churn prediction, fraud detection, and scientific data analysis.

Mathematical optimisation involves either minimisation or maximisation of a quantified objective function that is characterised by a set of model parameters. The big idea is to find an optimally tuned set of model parameters at which these objective functions achieve its lowest point (minimum) or highest peak (maximum). Based on the application of constraints to optimisation problems, it can be classified into two categories, namely, *constrained,* and *unconstrained* optimisation problems.

Unconstrained Optimisation: An unconstrained optimization problem deals with finding the local minimiser or maximiser x^* of a real valued and smooth objective function (cost or loss) *f(x)* of *n* variables, given by,

$$\text{minimise}_{\,x}\ f(x)$$

Constrained optimization problem seeks a decision variable ***x*** that minimizes an objective function *subject to* certain conditions that the minimizer must meet, typically these conditions are imposed in terms of algebraic equations and inequalities, so general constrained problems assume the form,

$$\begin{aligned} &\text{minimise}_{\,x}\ f(x) \\ &\text{subject to: } a_i(x) = 0 \text{ for } i = 1,2\ldots.M \\ &\qquad\qquad c_j(x) < 0 \text{ for } j = 1,2\ldots.L \end{aligned}$$

Example: Unconstrained Optimization

Let's start with an example. Suppose the function $P(x,y)$ represents the profit (in millions) for selling x thousands of product A and y thousands of product B. We want to find out how many of each product to sell in order to maximize our profit[2]. How can we identify points in the domain of $P(x,y)$ which might be maxima or minima? For one variable calculus, the maxima occurred anywhere that the function was flat. The function is neither increasing nor decreasing at this point, so its derivative is zero. Now we have more than one direction in which the function can change. We'd really like for the function $P(x,y)$ to be flat in all directions. To be flat in every direction means that the direction derivative of the function must be zero in every direction. Thus, for every vector u which has magnitude 1, we want,

$$f_u(x,y) = grad\, f(x,y) \cdot u = 0$$

This implies that norm($grad\, f(x,y)$) = 0. But this second condition can only occur if the gradient of f is, in fact, zero. So, a necessary condition for the point (x_0, y_0) to be a maxima or minima of $f(x,y)$ is that,

$$grad\, f(x_0,y_0) = 0$$

Also note that these conditions are merely necessary, but not sufficient. Just as in one variable calculus, where inflection points creep in, here we have saddle points. At these points, the function is concave up in one direction and concave down in another. After the calculation of stationary

points, it is tested whether the point is a minima or a maxima or a saddle point. The contour diagram shows –

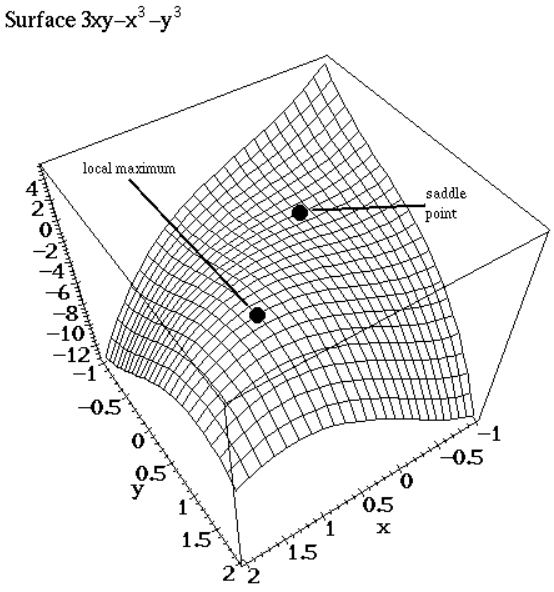

Fig 1. Contour Diagram for an Unconstrained Optimisation Problem.

Based on the hessian value at these stationary points, it can be compared if the points are local minima/maxima or saddle points. Hence, to maximize profit we can calculate the amount of Product A and Product B to be sold.

In this report, we would be focusing on an issue related to supervised learning and unconstrained optimisation. Constrained and unconstrained optimization problems are mathematical problems focused on finding the optimal values of variables that minimize or maximize a given objective function, subject to certain constraints. On the other hand, for a supervised learning we would explore, regression and classification.

Two fundamental types of supervised learning are regression and classification, where the goal is to learn a mapping from input features to an output variable based on labelled training data.

Regression: It is a type of supervised learning task, in which for a given input regression predicts a continuous output value (probability of an event's occurrence). Regression analysis is the "go-to method in analytics," says Redman[3]. Corporations are using it to make smart-decisions. Most companies use regression analysis to explain a phenomenon they want to understand (for example, Why did customer service calls drop last month?); predict things about the future (for example, What will sales look like over the next six months?); or to decide what to do (for example, Should we go with this promotion or a different one?).

Classification: Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. The model is training on training data and then it is evaluated on test data before being used to perform prediction on an unseen input. Examples of

classification problems include – classifying email as spam or not, classifying different species of a flower or classifying whether a method/service/product is good or bad, etc. There are four main types of classification tasks namely, Binary classification, Multi-class classification, Multi-Label classification and Imbalanced classification. We would be focusing on binary classification through an experiment in this report.

Binary Classification: This supervised learning task involves two-class labels. For example, medical tests to determine if a patient has a certain type of disease or not, quality control in industry, whether a specification has been met or not, credit card fraud detection, etc.

2. Problem Definition

The motive of this project is to address an appropriate constrained/unconstrained optimisation problem. There are certain guidelines for solving Optimisation problems –

- Identifying what is to be maximised or minimised and what the constraints are (if any).
- Define the objective function that is to be maximised or minimised depending on the goal. For example, in a classification task, we can minimise the error rate while in a regression task, we can maximise the correlation between predictions and actual values.
- Choose an appropriate optimisation algorithm depending on the different types objective functions. Usually in an optimisation task for classification we can use the following algorithms –
    - Logistic Regression
    - k-Nearest Neighbours
    - Decision Trees
    - Support Vector Machine
    - Naïve Bayes

    Some algorithms are specifically designed for binary classification and do not natively support more than two classes; examples include **Logistic Regression** and **Support Vector Machines**.

- Regularise your model to help prevent overfitting, which occurs when the model memorises the training data but doesn't generalises well to unseen examples. Common regularisation technique is to add a regularisation term to the objective function that penalises the complexity of the model.
- Evaluate and tune the model. The feedback can be used to tune the hyperparameters of optimisation algorithm and model architecture to improve performance.

In this project, we would be implementing binary classification on a dataset taken from UCI Machine Learning Repository called ***Ionosphere***. The Ionosphere dataset is a classic binary classification dataset that is often used for machine learning tasks. This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free

electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal[4]. It has 34 features (real) and 1 class (label). The task is typically framed as a binary classification problem where the goal is to predict whether the ionosphere is good or bad given the radar signal features.

3. Proposed Solutions

We would be focusing on three different algorithms to classify this dataset as follows –

k-Nearest Neighbors

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point[5]. Although it can be used for both regression as well as classification tasks, it is typically used as a classification algorithm. In a classification problem, a class label is allocated on the basis of a majority vote; the label that is most frequently represented around a given data point is used.

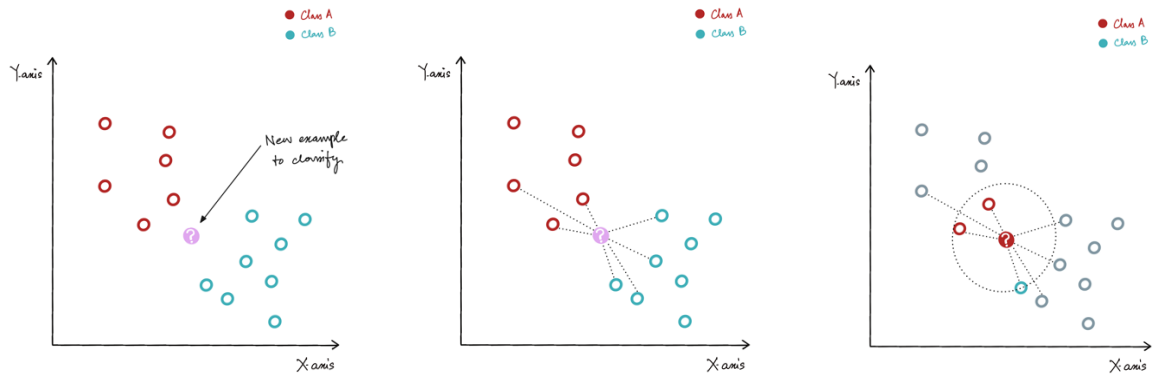

Fig 2. KNN Diagram

The K-NN working can be explained on the basis of the below algorithm:

1. Select the number ***k*** of the neighbors.
2. Calculate the Euclidean distance of ***k*** number of neighbors.
3. Take the ***k*** nearest neighbors as per the calculated Euclidean distance.
4. Among these ***k*** neighbors, count the number of the data points in each category.
5. Assign the new data points to that category/class for which the number of the neighbours is maximum.

How to select the value of ***k*** in the K-NN Algorithm?

- No particular way to determine the best value of ***k***. Trial and error applicable. The most preferred value for ***k*** is 5.
- A very low value for ***k*** such as ***k*** =1 or ***k*** =2, can be noisy and lead to the effects of outliers in the model.
- Large values for ***k*** are good, but it may find some difficulties.

The k-NN algorithm has been applied to many different fields, most notably classification. Among these use cases are:

- **Preprocessing of the data:** Missing values are common in datasets, but the KNN algorithm can estimate for those values through a procedure called "missing data imputation."
- **Recommendation engines:** The KNN algorithm has been used to automatically suggest more content to viewers based on clickstream data from websites.
- **Pattern Recognition:** KNN has also helped in pattern recognition, for example in the classification of text and numbers. This has been very useful for figuring out handwritten numbers on paperwork and envelopes that you may receive.

Logistic Regression

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on[6]. Logistic regression, despite its name, is a classification model rather than regression model. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. It is an extensively employed algorithm for classification in industry. The logistic regression model, like the Adaline and perceptron, is a statistical method for binary classification that can be generalized to multiclass classification.

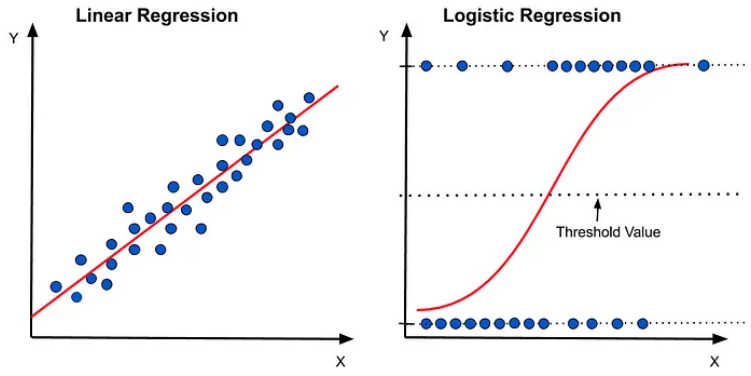

Fig 3. Logistic vs Linear Regression

Logistic regression uses a logistic function called a sigmoid function to map predictions and their probabilities. The sigmoid function refers to an S-shaped curve that converts any real value to a range between 0 and 1. If the estimated probability or output of the sigmoid function is greater than the predefined threshold, the model predicts that the data point belongs to that class. If the estimated probability is less than that predefined threshold, then the model predicts the instance does not belong to the class.
The sigmoid function is also known as an activation function for logistic regression, given as –

$$f(x) = 1 / (1 + e^{-x})$$

The following equation represents logistic regression:

$$y = \frac{e^{(b_0 + b_1 X)}}{1 + e^{(b_0 + b_1 X)}}$$

where, x = input value, y = predicted output, b0 = bias, b1 = weight

Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection[7]. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. The main objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

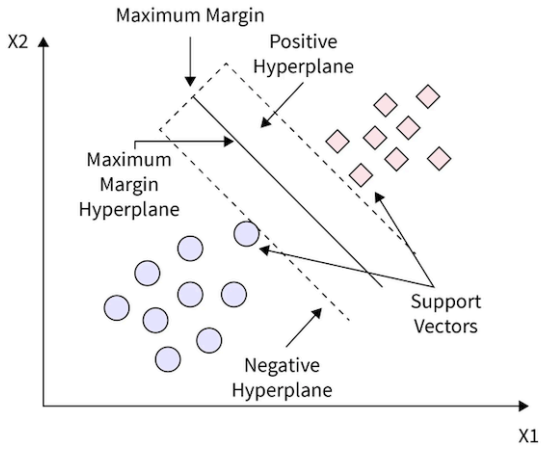

Fig 4. SVM

SVM algorithm is of two types:-

- **Linear SVM:** When the data points are linearly separable into two classes, the data is called linearly separable data. We use the linear SVM classifier to classify such data.

- **Non-linear SVM:** When the data is not linearly separable, we use the non-linear SVM classifier to separate the data points.

SVM working can be explained as below –

1. Gather a labeled dataset with binary labels. Scale/normalise the features of the input.
2. Kernel selection is used to transform the input features into a higher-dimensional space.
3. SVM aims to find the hyperplane that best separates the data into two classes while maximizing the margin between the classes. The training process involves solving a convex optimization problem to find the optimal hyperplane parameters (weights and bias) that maximize the margin and minimize classification errors.
4. A hyperparameter, commonly referred to as C is used to control the trade-off between achieving a smooth decision boundary and classifying training points correctly.
5. A decision boundary, hyperplane is calculated to separate the two classes.
6. New input is fed to the SVM to predict their classes by evaluating the decision function.

4. Results

The following result was obtained for ***Ionosphere*** dataset for different algorithms described above (ascending order of percentage of accuracy) –

```
KNN Confusion Matrix:
[[17 11]
 [ 0 43]]
KNN Accuracy: 84.51%

LogisticRegression Confusion Matrix:
[[18 10]
 [ 0 43]]
Accuracy: 85.92%

SVM Confusion Matrix:
[[24  4]
 [ 0 43]]
SVM Accuracy: 94.37%
```

In general, we can say that,

- SVM performed the best in terms of accuracy with 94.37%.

- kNN and logistic regression has comparable results with accuracy by kNN being only 1% lower than that of logistic regression.

These results suggest that the SVM model might be the best choice for this dataset, as it provides both high accuracy and high precision (low false positive rate). However, the actual choice of algorithm may depend on factors such as the complexity of the dataset, the availability of additional data, and the computational resources available for training the model.

5. Analysis & Interpretation

Factors that influence these results –

- ***Model Complexity:*** It is difficult for different models to capture underlying patterns in the data. Simple algorithms like logistic regression and kNN algorithm may not be able to capture all the complexities of the data hence result in lower percentage of accuracy.
- ***Model Parameters:*** Each model's parameters can be tuned to achieve greater accuracy.
- ***Training Data:*** Choice of training data and the versatility of the same in terms of the balance between positive and negative examples impacts the performance of a model.
- ***Data Quality:*** Variety, versatility, completeness, etc. are some qualities of data that significantly improve model training.

Disadvantages of k-NN Algorithm –

Since KNN is a lazy algorithm, it requires more memory and data storage compared to other classifiers. This can be costly both in time and money. More memory and storage increases business costs, and computing more data can take longer. Various data structures, such as Ball-Tree, have been created to address computational inefficiencies, but depending on business problem, another classifier may be ideal. The KNN algorithm leads to the curse of dimensionality, which means that it doesn't perform well with high-dimensional data inputs. Curse of dimensionality indirectly incurs overfitting. The value of k impacts the model's behaviour. Low value of k leads to overfitting whereas higher values tend to smooth out the prediction values.

Disadvantages of Logistic Regression –

Logistic regression might not be accurate for small sized sample dataset. The model produced by logistic regression is based on a smaller number of actual observations. This can result in overfitting.

6. Conclusion & Recommendations

The result suggests that the performance of the models varies based on the input features. While some models may perform well on a subset of the input features, others may not. This indicates that the optimal model may depend on the specific task at hand and the data available for training. The chosen models may not be the best-suited for the specific task. In some cases, it may be more appropriate to use different algorithms, such as decision trees or Forest Random, to achieve better performance. The dataset ***Ionosphere*** had 34 features whereas around 351 samples. Small number of samples compared to the large number of features, may lead to the curse of dimensionality. This can also lead to overfitting, where the model learns noise in the data rather than meaningful patterns. Hence explaining the low accuracy in two algorithms employed. With a small dataset, the quality and representativeness of the data become crucial. If the dataset doesn't adequately capture the diversity of the underlying population, the model's performance may be limited.

Ways to improve model's accuracy –

- Data Expansion: Presence of more data results in better and more accurate machine learning models.
- Outliers & Missing values: To improve accuracy, imputation of missing data or addressing outliers leads to better models.
- Feature Engineering: New information can be extracted from existing features; these features may have a higher ability to explain the variance in the training data. Thus, giving improved model accuracy.
- Feature Transformation: Data normalisation to change the scale of a variable.

References

[1] techtarget

[2] citadel.sjfc

[3] medium

[4] Sigillito,V., Wing,S., Hutton,L., and Baker,K.. (1989). Ionosphere. UCI Machine Learning Repository. https://doi.org/10.24432/C5W01B.

[5] IBM

[6] sciencedirect

[7] scikit-learn