Sujet:

"Comparative Analysis of Multiclass SVM Classification Using One-vs-One and One-vs-All Strategies

Group:

Diani Achraf
Daoudi Mohammed Oualid

Supervised by:

Mr El Afia Abdellatif
Mr El Houssaine Hssayni

Année Universitaire 2023-2024

Abstract

This report evaluates the effectiveness of multiclass classification using Support Vector Machines (SVM) with the Sequential Minimal Optimization (SMO) method, exploring both One-vs-One (OvO) and One-vs-All (OvA) strategies. The study is conducted using the "drug200" dataset, which consists of 200 instances and 6 attributes, aimed at predicting drug prescription based on patient characteristics. The comparative analysis focuses on evaluating and contrasting the performance metrics such as accuracy, precision, recall, and F1-score for each classification strategy.

Contents

1 Description of the dataset and the preprocessing steps : 4
1.1 Methods . . . 4
1.1.1 Dataset Description . . . 4
1.1.2 Preprocessing Steps . . . 4

2 SVM with SMO : 6
2.1 Support Vector Machines and Sequential Minimal Optimization . . . 6
2.1.1 Fundamentals of SVM . . . 6
2.1.2 Sequential Minimal Optimization . . . 6
2.1.3 Classification with SVM . . . 7
2.2 Multiclass Classification with SVM . . . 7
2.2.1 One-vs-One (OvO) Approach . . . 7
2.2.2 One-vs-All (OvA) Approach . . . 7
2.2.3 Pseudocode for SVM with SMO . . . 8
2.2.4 One-vs-One (OvO) SVM Classification Algorithm . . . 9
2.2.5 One-vs-All (OvA) SVM Classification Algorithm . . . 9

3 Results and Visualization 10
3.1 Comparison of SVM Multiclass Classification Strategies . . . 10
3.1.1 Overall Performance Analysis . . . 10
3.1.2 Class-wise Performance and Misclassifications . . . 10
3.1.3 Visualization of Results . . . 10
3.1.4 Insights and Recommendations . . . 10
3.1.5 Conclusion . . . 11

Chapter 1

Description of the dataset and the preprocessing steps :

1.1 Methods

1.1.1 Dataset Description

The "drug200" dataset used in this study consists of 200 instances, each representing a patient's medical profile. There are 6 attributes per instance:

- **Age**: The age of the patient.
- **Sex**: The gender of the patient (Male or Female).
- **BP**: Blood pressure levels (HIGH, NORMAL, or LOW).
- **Cholesterol**: Cholesterol levels (NORMAL or HIGH).
- **Na_to_K**: Sodium to potassium ratio in the blood.
- **Drug**: The type of drug prescribed (DrugA, DrugB, DrugC, DrugX, and DrugY).

This dataset is aimed at predicting the type of drug that should be prescribed to a patient based on their medical parameters.

1.1.2 Preprocessing Steps

Preprocessing is crucial for preparing the dataset for SVM classification. The steps undertaken include:

1. **Handling Missing Values**: The dataset was checked for missing values. In cases of missing data, strategies such as imputation or deletion would be applied, depending on the context.
2. **Encoding Categorical Variables**: Attributes like Sex, BP, and Cholesterol were encoded using one-hot encoding to convert them into a format suitable for the SVM models.
3. **Feature Scaling**: The features 'Age' and 'Na_to_K' were scaled using standardization to ensure that they do not unduly influence the model due to their range.

	Age	Na_to_K	Sex_F	Sex_M	BP_HIGH	BP_LOW	BP_NORMAL	Cholesterol_HIGH	Cholesterol_NORMAL	Drug_drugA	Drug_drugB	Drug_drugC	Drug_drugX	Drug_drugY
0	-1.291591	1.286522	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
1	0.162699	-0.415145	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
2	0.162699	-0.828558	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
3	-0.988614	-1.149963	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
4	1.011034	0.271794	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0

Figure 1.1: Caption

4. **Data Splitting**: The dataset was split into training and testing sets with a ratio of 80:20 to evaluate the performance of the models effectively.

Chapter 2

SVM with SMO :

2.1 Support Vector Machines and Sequential Minimal Optimization

Support Vector Machines (SVM) are a set of supervised learning methods used for classification, regression, and outliers detection. The effectiveness of SVMs in high-dimensional spaces makes them particularly powerful for a wide range of classification tasks.

2.1.1 Fundamentals of SVM

SVM works by finding a hyperplane in an N-dimensional space (where N is 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 goal is to find a plane that has the maximum margin, i.e., the maximum distance between data points of both classes.

- **Maximal Margin Classifier:** It is the hyperplane that separates the classes with the largest minimum distance (or margin) to the nearest training data points of any class.
- **Support Vectors:** Data points that are closest to the hyperplane and influence its position and orientation are called support vectors. These are the critical elements of the training data.
- **Kernel Trick:** For non-linearly separable data, SVM can be equipped with a kernel function allowing them to operate in a transformed feature space.

2.1.2 Sequential Minimal Optimization

The Sequential Minimal Optimization (SMO) algorithm is a method for solving the quadratic programming (QP) problem that arises during the training of support vector machines. It breaks the large QP problem into a series of smallest possible QP problems, which are then solved analytically.

- **Decomposition Technique:** SMO decomposes the overall QP problem into 2-dimensional QP problems that can be solved by an analytical solution which simplifies the computations.
- **Working Set Selection:** At every step, SMO chooses two alpha coefficients to optimize and finds the optimal values for these coefficients.
- **Efficiency:** This method is particularly favored for its efficiency in handling large datasets without the need for additional memory for auxiliary data structures, unlike other gradient descent-based algorithms.

This focus on the fundamentals and optimization method sets the stage for discussing the multiclass classification strategies involving SVM in the subsequent sections.

2.1.3 Classification with SVM

Once the optimal hyperplane is established using the SVM algorithm, it serves as the decision function for classifying new data points. The decision function defines whether a new point falls on one side of the hyperplane or the other, determining its class membership.

- **Decision Function:** The SVM decision function is expressed as $f(x) = \text{sign}(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b)$, where x_i are support vectors, y_i are their labels, α_i are the Lagrange multipliers, K is the kernel function, and b is the bias.
- **Binary Classification:** In its simplest form, SVM is a binary classifier that assigns data points to one of two classes based on the sign of the decision function.
- **Score and Margin:** The absolute value of the decision function represents the distance of a data point from the hyperplane, which correlates with the confidence of the classification. Points closer to the hyperplane are less confidently classified than those farther away.

This decision-making process highlights the effectiveness of SVM in distinguishing between classes with a clear margin of separation, ensuring robustness and accuracy in predictions.

2.2 Multiclass Classification with SVM

In scenarios where there are more than two classes, SVM can be extended to perform multiclass classification using strategies like One-vs-One (OvO) and One-vs-All (OvA). These methods adapt the inherently binary SVM to handle multiple classes effectively.

2.2.1 One-vs-One (OvO) Approach

The One-vs-One approach involves training a separate SVM classifier for each pair of classes. If there are n classes, this results in $\frac{n(n-1)}{2}$ classifiers. Each classifier is trained on data from two classes that it must distinguish between.

- **Classifier Training:** For each pair of classes, the SVM is trained only on the data pertaining to those two classes, ignoring all other data.
- **Decision Strategy:** During prediction, a voting strategy is used where each classifier votes for a class. The class with the most votes determines the label of the test instance.
- **Scalability and Complexity:** While this method is more computationally intensive due to the larger number of classifiers, it is less sensitive to imbalances among classes.

2.2.2 One-vs-All (OvA) Approach

Alternatively, the One-vs-All approach trains one SVM classifier per class, where each classifier distinguishes one class from all other classes combined. This strategy requires training n classifiers for n classes.

- **Classifier Training:** Each classifier is trained to identify whether a datapoint belongs to its respective class or not, making it a binary classification problem against all other classes.
- **Decision Strategy:** During prediction, each classifier provides a confidence score that the datapoint belongs to its class. The class whose classifier gives the highest confidence score is the predicted label.
- **Simplicity and Efficiency:** Although potentially subject to class imbalance, this method is computationally less demanding compared to OvO and is straightforward to implement.

Both methods have their strengths and weaknesses, and the choice between OvO and OvA may depend on the specific characteristics of the dataset and the problem at hand.

2.2.3 Pseudocode for SVM with SMO

Algorithm 1 SVM Training Using Sequential Minimal Optimization (SMO)

```
procedure SVMwithSMO(X, y, C, eps, max_iter, kernel)
    n_samples, n_features ← size(X)
    α ← array of zeros of length n_samples
    b ← 0
    K ← pre-compute kernel matrix between all pairs in X
    k ← 0
    while k < max_iter do
        num_alpha_changed ← 0
        for i ← 0 to n_samples − 1 do
            Ei ← b + ∑(α · y · K[i]) − y[i]
            if ((y[i] · Ei < −eps and α[i] < C) or (y[i] · Ei > eps and α[i] > 0) then
                j ← random integer not equal to i
                Ej ← b + ∑(α · y · K[j]) − y[j]
                alpha_old_i, alpha_old_j ← α[i], α[j]
                Compute L, H based on y[i] and y[j]
                eta ← 2 · K[i, j] − K[i, i] − K[j, j]
                if eta < 0 then
                    α[j] ← α[j] − (y[j] · (Ei − Ej)/eta)
                    α[j] ← clip(α[j], L, H)
                    if |α[j] − alpha_old_j| > 0.00001 then
                        α[i] ← α[i] + y[i] · y[j] · (alpha_old_j − α[j])
                        Update b
                        num_alpha_changed ← num_alpha_changed + 1
                    end if
                end if
            end if
        end for
        if num_alpha_changed = 0 then
            k ← k + 1
        else
            k ← 0
        end if
    end while
    w ← ∑(α · y · X.T)
    return w, b
end procedure
```

These pseudocode segments provide a high-level view of the steps involved in training SVM classifiers using the One-vs-One and One-vs-All strategies. They can be included in your LaTeX document to illustrate the algorithmic process underlying each multiclass classification approach.

2.2.4 One-vs-One (OvO) SVM Classification Algorithm

Algorithm 2 One-vs-One SVM Classification

```
procedure OneVsOneSVM(X, y, C, eps, max_iter, kernel)
    Initialize list of classifiers W, b
    for each unique class i do
        for each unique class j > i do
            Create binary labels for classes i and j
            Select data for classes i and j
            Train SVM using SMO on selected data
            Save the trained model's weights and bias
        end for
    end for
    return W, b
end procedure
```

2.2.5 One-vs-All (OvA) SVM Classification Algorithm

Algorithm 3 One-vs-All SVM Classification

```
procedure OneVsAllSVM(X, y, C, eps, max_iter, kernel)
    Initialize weights W and biases b
    for each class i do
        Create binary labels (1 for class i, -1 for others)
        Train SVM using SMO on the entire dataset
        Save the trained model's weight and bias for class i
    end for
    return W, b
end procedure
```

Chapter 3

Results and Visualization

3.1 Comparison of SVM Multiclass Classification Strategies

This chapter provides a detailed comparison between the One-vs-All and One-vs-One strategies for SVM multiclass classification applied to the drug prescription dataset. Performance metrics such as accuracy, F1-score, and recall are analyzed, and the results of the confusion matrices are discussed to provide insights into each method's efficacy.

3.1.1 Overall Performance Analysis

The One-vs-All approach demonstrated a superior performance with an accuracy of 97.5%, closely matched by its F1-score and recall. In contrast, the One-vs-One approach, while slightly lower, achieved a commendable accuracy of 92.5% with a similar F1-score. These results indicate that while both methods are effective, One-vs-All shows a slight edge in overall metric performance.

3.1.2 Class-wise Performance and Misclassifications

Detailed examination of the classification reports and confusion matrices reveals that the One-vs-All method not only performs uniformly well across almost all classes but also manages class imbalances effectively. One-vs-One, while effective, shows variability, particularly with classes that have fewer samples, which can be seen from the misclassifications noted between class 0 and others.

3.1.3 Visualization of Results

A comparative bar chart of the performance metrics is provided to visualize the differences in accuracy, F1-score, and recall between the two methods. The chart clearly illustrates that One-vs-All consistently outperforms One-vs-One across these key metrics, reinforcing the numerical analysis.

3.1.4 Insights and Recommendations

From the analysis, it is clear that the One-vs-All method not only provides higher accuracy but also shows greater consistency across different classes, making it a preferred choice for this specific dataset. However, the One-vs-One method could still be considered where class-specific tuning is required, or where computational constraints exist in training numerous classifiers.

Given these findings, further exploration with alternative SVM kernel functions or different values of the regularization parameter C might yield improvements, particularly for the One-vs-One approach. Additional diagnostics such as ROC curves for each class could provide deeper insights into classifier behavior under both strategies.

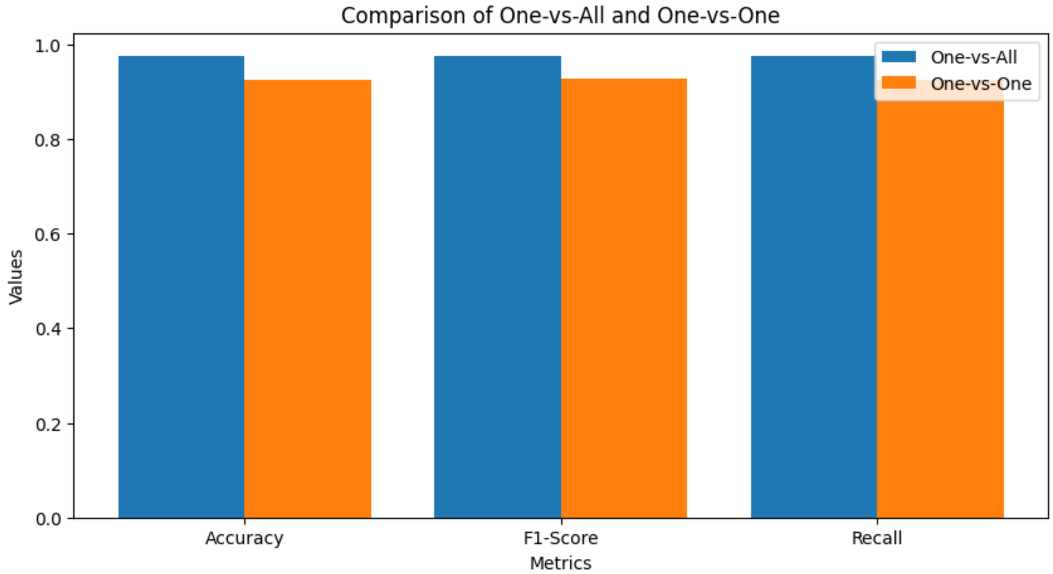

Figure 3.1: Bar chart comparing the accuracy, F1-score, and recall for One-vs-All and One-vs-One strategies.

3.1.5 Conclusion

The findings from this study suggest that the choice of multiclass classification strategy can significantly impact performance, and the optimal choice may depend on the specific characteristics and distribution of the dataset. One-vs-All emerges as the more robust method for this particular application, providing a balance of high accuracy and reliability across classes.

Bibliography

[1] Platt, J., *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*, 1998. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-98-14.pdf

[2] Kaggle, *Drug Classification*, accessed 2023. https://www.kaggle.com/datasets/prathamtripathi/drug-classification

[3] Hsu, C.-W. and Lin, C.-J., *A Comparison of Methods for Multiclass Support Vector Machines*, IEEE Transactions on Neural Networks, 2002. https://www.csie.ntu.edu.tw/~cjlin/papers/multisvm.pdf

[4] Rifkin, R. and Klautau, A., *In Defense of One-Vs-All Classification*, Journal of Machine Learning Research, 2004. http://www.jmlr.org/papers/volume5/rifkin04a/rifkin04a.pdf

[5] Schölkopf, B. and Smola, A.J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002. https://mitpress.mit.edu/9780262194754/learning-with-kernels/