GitHub: https://github.com/helga119/Machine-Learning-Tutorial.git

Title

Exploring the Behaviour of Support Vector Machines: A Tutorial on Linear vs RBF Kernels

Objective

Choosing the right kernel is one of the most crucial decisions when building a Support Vector Machine (SVM) model. Kernels serve as tools for the SVM to distinguish between different groups of data. You can think of a kernel as a method that SVM uses to draw a line (or boundary) that separates one class from another.

This tutorial aims to explore and compare the behavior of SVM using two popular kernels: the linear kernel and the Radial Basis Function (RBF) kernel. Our focus will be on understanding how these kernels influence the decision boundary and the performance of the model, using the Breast Cancer dataset as a real-world example. I will guide you step by step through the process of implementing SVM models and evaluating their performance in Python.

Why is this topic important?

Choosing the right kernel is a critical step in building an effective Support Vector Machine (SVM) model. A linear kernel separates data using a straight line (or a flat plane in higher dimensions). It is very simple, making it an excellent choice for problems where the data can be divided cleanly with a straight boundary. However, linear kernels may struggle when the data exhibits more complicated patterns or overlaps. In these cases, the Radial Basis Function (RBF) kernel is a better option.

The RBF kernel can create curved or flexible boundaries, allowing the SVM to handle data with more complex relationships. This flexibility, however, comes with a drawback: RBF kernels are slower to compute and harder to interpret. Knowing when to use a linear or RBF kernel can significantly impact the success of your model. For simpler problems, a linear kernel works well and saves time. For more complex datasets, such as those involving non-linear relationships or overlapping classes, the RBF kernel is more effective. By choosing the right kernel, you can build SVM models that perform better and efficiently address your specific problem.

What is SVM?

Support Vector Machine (SVM) is a supervised learning algorithm used for tasks like classification and regression. Its main goal is to find the optimal decision boundary, known as the hyperplane, that best separates data points into their respective classes. For example, in

the Breast Cancer dataset, SVM can classify tumors as benign (non-cancerous) or malignant (cancerous) based on features like size, texture, and smoothness. The algorithm identifies the hyperplane that maximizes the gap, or margin, between the two classes, ensuring robust and accurate predictions.

Understanding kernels

Kernels are essential to the functionality of Support Vector Machines (SVM), as they allow the algorithm to adapt to the structure of a dataset. Kernels transform data into a higher-dimensional space, enabling the SVM to define boundaries that effectively separate different classes. The choice of kernel determines the nature of these boundaries and directly impacts the model's success. The linear kernel is a straightforward option that performs well when the classes can be separated by a straight line or flat plane. Its computational simplicity and efficiency make it an ideal choice for linearly separable data or situations where interpretability is a priority.

In contrast, the RBF (Radial Basis Function) kernel offers greater flexibility by mapping data into a higher-dimensional space. This transformation allows the SVM to create non-linear boundaries that can accommodate complex patterns and overlapping data. Choosing the appropriate kernel requires an assessment of the dataset's characteristics. For simple, linearly separable problems, the linear kernel provides robust performance without incurring unnecessary computational costs. However, when dealing with non-linear patterns or datasets with significant class overlap, the adaptability of the RBF kernel often makes it the better choice. It is essential to evaluate the separability and complexity of the data through visualization or statistical analysis to make an informed decision. This ensures that the chosen kernel aligns with the problem's requirements and optimizes the model's effectiveness.

Mathematics behind kernels

Understanding the mathematics behind kernels is crucial for comprehending how Support Vector Machines (SVM) adapt to various types of data. Kernels play a pivotal role in determining how SVM separates classes, and their mathematical properties provide the flexibility needed to handle both simple and complex data sets. Below, we will break down the two most commonly used kernels: the Linear kernel and the Radial Basis Function (RBF) kernel.

Linear Kernel

The linear kernel is the simplest and most intuitive type of kernel. It computes the dot product between two vectors, effectively measuring their similarity in the original input space.

*It is defined as , **K(x, y) = x * y** , where x and y are the input vectors.*

- This operation involves multiplying corresponding elements of two vectors and summing them. For example, if x = [1,2] and y =[3,4] their dot product is:

  K(x, y) = 1 x 3 + 2 x4 = 3 + 8 = 11

  A larger dot product indicates higher similarity between the two vectors.
- In Support Vector Machine (SVM) classification, the linear kernel creates a hyperplane that separates different classes in the input space. This hyperplane serves as a flat boundary, such as a straight line in two dimensions (2D) or a flat plane in three dimensions (3D). This method is particularly effective when the data is linearly separable, meaning that a straight boundary can completely separate the classes without any overlap.
- Since the linear kernel operates directly within the input space, it is computationally efficient. This makes it an excellent choice for large datasets or situations where interpretability is important.

RBF (Radial Basis Function) kernel

The RBF kernel, or Radial Basis Function kernel, is a non-linear kernel that allows Support Vector Machines (SVM) to handle datasets with complex relationships. It achieves this by mapping the input data into a higher-dimensional space using an exponential function. This transformation enables the SVM to identify a linear boundary in the newly created space.

Kernal Function

It is defined as

$K(x, y) = exp(-gamma * ||x - y||^2)$ *, where x and y are the input vectors, gamma is a positive parameter, and ||x - y|| is the Euclidean distance between x and y.*

- The term ||x-y||2 represents the squared Euclidean distance between two data points , x and y. it quantifies how far apart the two points are in the space. Points

that are closer together have higher similarity scores, whiles those apart have lower similarity scores.

- Gamma is a parameter that determines the influence individual data points have on the decision boundary. A higher gamma value creates a more complex decision boundary, enabling the model to capture finer patterns in the data. In contrast, a lower gamma value results in a smoother, simpler boundary. Selecting the appropriate gamma value is crucial; an excessively high value can lead to overfitting, while a low value may result in underfitting the data.

Why RBF Creates Curved Boundaries

The RBF kernel transforms the data into an infinite-dimensional space, making it easier to separate non-linear patterns with a linear boundary. Here's how it works:

1. In the input space, data points may overlap or form complex shapes that cannot be separated by a straight line.
2. The RBF kernel maps these points into a higher-dimensional space, where the classes become linearly separable.
3. When this linear boundary is projected back into the original input space, it creates a curved decision boundary that can adapt to the complexity of the data.

For example:

Imagine a dataset consisting of two classes, where one class forms a circle nested inside another circle. In the original 2D space, these classes cannot be separated by a straight line. However, the Radial Basis Function (RBF) kernel can map this data into a higher-dimensional space, allowing for a linear boundary to divide the two classes. When this boundary is projected back into the 2D space, it appears as a circle.

Support Vector Machines for Breast Cancer Classification

This tutorial explores the application of Support Vector Machines (SVMs) to classify breast cancer cases as either malignant or benign. It highlights how SVMs excel in medical tasks that require accuracy, efficiency, and ethical considerations. By utilizing both linear and radial basis function kernels, the goal is to examine how the choice of kernel impacts classification performance. Metrics such as accuracy, precision, recall, and F1-score will be used to evaluate the effectiveness of each model, helping to identify the contexts in which one kernel may outperform the other.

The Breast Cancer Dataset is the foundation for this analysis. It contains 569 samples, each described by 30 numerical features related to tumor characteristics, including radius,

texture, and perimeter. The target variable indicates whether a tumor is malignant (1) or benign (0). Before modeling, the data undergoes preprocessing to ensure it is clean and ready for analysis. Unnecessary columns, such as "id" and "Unnamed: 32," are removed, and the diagnosis column is encoded as binary values.

```
data = data.drop(columns=['id', 'Unnamed: 32'])
data['diagnosis'] = data['diagnosis'].map({'M': 1, 'B': 0})
```

To prepare the dataset for Support Vector Machine (SVM) analysis, the features and target variable are divided into training and testing subsets, with 70% allocated for training and 30% reserved for testing. Since SVM is sensitive to feature scaling, all numerical features are standardized using StandardScaler to ensure a mean of zero and a variance of one. This step is crucial for optimizing SVM performance and ensuring consistent results.

```
X = data.drop(columns=['diagnosis'])
y = data['diagnosis']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

The tutorial compares two Support Vector Machine (SVM) models. The first model uses a linear kernel, which is ideal for datasets where a straight decision boundary can effectively separate classes. The second model utilizes a Radial Basis Function (RBF) kernel, which is more suitable for capturing non-linear relationships within the data. Both models are trained with the same regularization parameter (C=1.0) and default hyperparameters to ensure a fair comparison. This setup highlights the differences in how the kernels handle the dataset and provides insights into selecting the most appropriate kernel based on the complexity of the data.

```
linear_svm = SVC(kernel='linear', C=1.0, random_state=42)
linear_svm.fit(X_train, y_train)

rbf_svm = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=42)
rbf_svm.fit(X_train, y_train)
```

```
▾ SVC
SVC(random_state=42)
```

```
y_pred_linear = linear_svm.predict(X_test)
y_pred_rbf = rbf_svm.predict(X_test)
```

Results

The models are evaluated with the following results:

1. Linear kernel SVM performance

- Accuracy: 97.66%
- Precision:0.98(for benign and malignant classes)
- Recall: 0.98 (for benign) and 0.97 (for malignant)
- F1-score: 0.98
- Confusion matrix:

```
Confusion Matrix - Linear Kernel:
 [[106   2]
 [  2  61]]
```

2. RBF kernel SVM performance

- Accuracy: 97.66%
- Precision:0.98(for benign and malignant classes)
- Recall: 0.98 (for benign) and 0.97 (for malignant)
- F1-score: 0.98
- Confusion matrix:

```
Confusion Matrix - Linear Kernel:
 [[106   2]
 [  2  61]]
```

The similar results indicate that the dataset is mostly linearly separable. The linear kernel offers comparable accuracy with simpler calculations, proving sufficient for this problem.

Discussion

The results from both the linear and RBF kernels demonstrated identical performance metrics for this dataset, indicating that the data is predominantly linearly separable. However, in many cases, the choice of kernel can lead to differences in performance, making it essential to understand these variations for effective model evaluation.

When one kernel outperforms another in terms of accuracy, precision, recall, or F1-score, it suggests that the kernel is better at capturing the dataset's structure. For instance, an RBF kernel may excel in datasets with non-linear patterns, while a linear kernel may be sufficient for simpler problems. When performance differences are marginal, opting for the simpler linear kernel is often preferable because of its computational efficiency.

Another important consideration is the trade-off between computational cost and model performance. Linear kernels are computationally faster, making them ideal for large datasets or real-time applications. In contrast, the RBF kernel, while more resource-intensive, is better suited for datasets that require complex decision boundaries. Additionally, it is crucial to remain mindful of the potential for overfitting. If a model achieves high accuracy on the training set but performs poorly on the test set, this could indicate overfitting, which may result from using an excessively complex kernel like RBF. By understanding and interpreting these metrics, you can make informed decisions about which kernel is most suitable for your specific needs.

References

1. Tibrewal, T.P. (2023) Support Vector Machines (SVM): An Intuitive Explanation, Low Code for Data Science. Available at: https://medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106.
2. Support Vector Machine (SVM) in 2 minutes (no date) www.youtube.com. Available at: https://www.youtube.com/watch?v=_YPScrckx28.
3. Starmer, J. (2019) *Support Vector Machines Part 1 (of 3): Main Ideas!!!*, *www.youtube.com*. Available at: https://www.youtube.com/watch?v=efR1C6CvhmE.
4. GeeksforGeeks (2024) How to Choose the Best Kernel Function for SVMs, GeeksforGeeks. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/how-to-choose-the-best-kernel-function-for-svms/ (Accessed: 20 April 2024).
5. SVM Kernels : Data Science Concepts (no date) www.youtube.com. Available at: https://www.youtube.com/watch?v=OKFMZQyDROI.
6. SVM with polynomial kernel visualization (no date) www.youtube.com. Available at: https://www.youtube.com/watch?v=3liCbRZPrZA.
7. Mahesh Huddar (2023) What is Kernel Trick in Support Vector Machine | Kernel Trick in SVM Machine Learning Mahesh Huddar, YouTube. Available at: https://www.youtube.com/watch?v=Js2oAqEN_h0 (Accessed: 2 December 2024).
8. Udacity (2015) Kernel and Gamma - Intro to Machine Learning, YouTube. Available at: https://www.youtube.com/watch?v=pH51jLfGXe0 (Accessed: 2 December 2024).
9. ritvikmath (2021) 'SVM Dual : Data Science Concepts', YouTube. Available at: https://www.youtube.com/watch?v=6-ntMIaJpm0 (Accessed: 23 December 2022).
10. Siddhardhan (2021) 7.3.2. Math behind Support Vector Machine Classifier, YouTube. Available at: https://www.youtube.com/watch?v=y0_Qq6fXzCs&list=PLfFghEzKVmjvzS4DILijsdQk27Ew7xIPu&index=2 (Accessed: 2 December 2024).
11. GeeksforGeeks (2024) Implementing SVM from Scratch in Python, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/implementing-svm-from-scratch-in-python/#step-2-svm-class-definition (Accessed: 2 December 2024).
12. Introduction to Support Vector Machines (SVM) (2020) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/.
13. Team, P. (2022) Support Vector Machine, Python Geeks. Available at: https://pythongeeks.org/support-vector-machine/ (Accessed: 2 December 2024).
14. Bento, C. (2020) Support Vector Machines explained with Python examples, Medium. Available at: https://towardsdatascience.com/support-vector-machines-explained-with-python-examples-cb65e8172c85.
15. RBF SVM Parameters in Scikit Learn (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/rbf-svm-parameters-in-scikit-learn/.

16. Radial Basis Function Kernel - Machine Learning (2020) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/radial-basis-function-kernel-machine-learning/.
17. How to Make Better Models in Python using SVM Classifier and RBF Kernel (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/how-to-make-better-models-in-python-using-svm-classifier-and-rbf-kernel/.
18. Creating linear kernel SVM in Python (2018) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/creating-linear-kernel-svm-in-python/.
19. GeeksforGeeks (2024) What is the influence of C in SVMs with linear kernel?, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-is-the-influence-of-c-in-svms-with-linear-kernel/ (Accessed: 2 December 2024).
20. Support Vector Regression (SVR) using linear and non-linear kernels (no date) scikit-learn. Available at: https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html.
21. Malik, U. (2018) Implementing SVM and Kernel SVM with Python's Scikit-Learn, Stack Abuse. Stack Abuse. Available at: https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/.
22. Oleszak, M. (2021) SVM Kernels: What Do They Actually Do?, Medium. Available at: https://towardsdatascience.com/svm-kernels-what-do-they-actually-do-56ce36f4f7b8.