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

This choice of kernel is one of the most important decisions when building an SVM model. Kernels are the tools that SVM uses to determine how to separate different groups of data. You can think of a kernel as a technique SVM applies to draw a line (or boundary) that divides one class from another.

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

Why is this topic important?

Choosing the right kernel us a critical step in building an effective SVM model. A linear kernel separates data using a straight line (or flat plane in higher dimension). It is fast, simple, and easy to understand, making it a great choice for problems where the data can be divided cleanly with a straight boundary. However, linear kernels may struggle when the data has more complicated patterns or overlaps.

In these cases, the RBF kernel becomes the better option. RBF kernel can create curved or flexible boundaries, allowing the SVM to handle data with more complex relationships. This flexibility however comes with a draw back- RBF kernels are slower to compute and harder to interpret.

Knowing when to use linear or RBF kernel can make a significant difference in the success of your model, for simpler problems, a linear kernel works well and saves time, for more complex dataset, such as 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 solve your specific problem efficiently.

What is SVM?

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

For instance in the Breast Cancer dataset, SVM might divide tumours into benign (non-cancerous) and malignant(cancerous) based on features like size , texture, and smoothness. The algorithm identifies the hyperplane that creates the largest possible gap, or margin, between the two classes, ensuring robust and accurate predictions.

The power of SVM lies in its ability to handle both:

1. **Linearly separable data**, where classes can be divided with a straight boundary
2. **Non-linearly separable data,** where more complex boundaries are required

Understanding kernels

Kernels are integral to the functionality of Support Vector Machines(SVM), allowing the algorithm to adapt to the structure of a dataset. They transform data into a higher-dimensional space, enabling SVM to define boundaries that separate classes effectively. The type of kernel chosen determines the nature of these boundaries and directly influences the model's success.

The linear kernel is a straightforward approach that works well when the data classed 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 problems where interpretability is a priority. In contrast, RBF kernel provides greater flexibility by mapping data into a higher dimensional space. This transformation allows SVM to create non-linear boundaries that can handle complex patterns and overlapping data.

Choosing the appropriate kernel involves assessing the dataset's characteristics. For straightforward linearly separable problems, the linear kernel delivers robust performance without unnecessary computational costs. However, when dealing with non-linear pattern or datasets with significant class overlap, the RBF kernel's adaptability often makes it the better choice. Evaluating the separability and complexity of the data through visualization or statistical analysis is essential to make an informed decision, ensuring 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 essential to understanding how SVM adapt to different types of data, kernels play a pivotal role in determining how SVM separate

classes and their mathematical properties give SVM the flexibility to handle both simple and complex data sets. Below we break down the two most commonly used kernels: the Linear kernel and the RBF kernel.

Linear kernel

The linear kernel is the simplest and most intuitive 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 SVM, the linear kernel produces a hyperplane that separates classes in the input space. The hyperplane is a flat boundary, such as a straight line in 2D or a flat plane in 3D. This approach is highly effective when the data is linearly separable, meaning that a straight boundary can completely separate the classes without overlap
- Because the linear kernel works directly in the input space, it is computationally efficient, making it ideal for a large dataset or when interpretability is key.

RBF (Radial Basis Function) kernel

The RBF kernel is a non – linear kernel that enables SVM to tackle datasets with complex relationships, it works by mapping the input data into a higher dimensional space using an exponential function, allowing the SVM to find a linear boundary in this transformed 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 : the parameter controls how much influence individual data points have on the decision boundary. A higher gamma value makes the decision boundary more complex, allowing the model to capture fine – grained patterns in the data. Conversely, a lower gamma value creates a smoother, simpler boundary. Choosing the right gamma value is critical, as an excessively high value can lead to overfitting, while a low value might underfit the data.

Why RBF Creates Curved Boundaries

The RBF kernel transform the data into an infinite – dimensional space where it becomes easier to separate non-linear patterns with a linear boundary, here's out it works:

1. In the input space, the data points may overlap of 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 form a curved decision boundary that can adapt to the data's complexity.

For example:

- Imagine a dataset where the two classes from one circle nested inside another. In the original 2 –d space this cannot be separated by a straight line. However the RBF kernel maps this data into a higher dimensional space where a linear boundary can divide the two classes. This boundary, when projected back into the 2-d space appears as a circle.

Support Vector Machines for Breast Cancer Classification

This tutorial explores the application of Support Vector Machines to classify breast cancer cases as either malignant or benign highlighting SVMs excel in medical tasks, requiring accuracy, efficiency and ethics. By using both linear and radial basis function kernels, the goal is to examine how kernel choice impacts classification performance. Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the effectiveness of each model and identify the contexts in which one kernel may outperform the other.

The Breast Cancer Dataset serves as a basis for this analysis. It consists of 569 samples, each described by 30 numerical features to tumour characteristics, including radius , texture, and perimeter. The target variable indicated whether a tumour is malignant(1) or benign(0). Before modelling 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 SVM, the features and target variable are split into training and testing subsets, with 70 % used for training and 30% reserved for testing. Since SVM is sensitive to feature scaling, all numerical features are standardised using StandardScalar to ensure a mean of zero and a variance of one, this steps 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 SVM models. The first uses a linear kernel, ideal for datasets where a straight decision boundary is sufficient to separate classes. The second employs an RBF kernel, which is better suited for capturing non-linear relationships in 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 difference in how the kernels handle the dataset and provides insights into selecting the most appropriate kernel based on dataset complexity.

```
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

Confusion matrices and metrics such as precision, recall, and F1-score help to evaluate SVM performance. A strong model will have High true positives and true negatives, with few false positives and false negatives. Precision measures how many predicted positives are correct, recall measure how many actual positives are identified, and F1-score balances these metrics for a comprehensive performance assessment.

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 identical results suggest the dataset is predominantly linearly separable. The linear kernel achieves comparable accuracy with simpler computations, making it sufficient for this problem.

Discussion

The results from both the linear and RBF kernels showed identical performance metrics for this dataset, indicating that the data is predominantly linearly separable. However, in many cases, the choice of kernel leads to differences in performance, and understanding these variations is critical for effective model evaluation.

When one kernel outperforms another in accuracy, precision, recall, or F1-score, it suggests that the kernel better captures the structure of the dataset. For example, an RBF kernel may excel in datasets with non-linear patterns, while a linear kernel may suffice for simpler problems. In cases where the performance difference is marginal, opting for the simpler kernel, such as the linear one is often preferable due to its computational efficiency.

Another important consideration is the trade-off between computational cost and model performance. Linear kernels are computational faster, making them ideal for large datasets or real-time applications. On the other hand, the RBF kernel, while more resource-intensive, is better suited for datasets requiring complex decision boundaries. Additionally, attention must be paid to the potential for overfitting. If a model achieves high accuracy on the training set but it performs poorly on the test set. This is often a sign of overfitting, which may result from an overly complex kernel such as RBF. By understanding and interpreting these metrics, you can make informed decisions about which kernel is most suitable to use.

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.