Project1 Machine learning Report
Support Vector Machine

Abdelateef Khaled Abdelateef 202001344

Department of Engineering, University of Science and Technology at Zewail City, Egypt

Contents

1 Problem Definition and Motivation

In supervised learning, particularly with Support Vector Machines (SVMs) for classification, selecting an appropriate kernel is crucial. Kernels, such as linear, polynomial, and radial basis function, play a key role in mapping input data into higher dimensions for SVMs to establish non-linear decision boundaries. The project is motivated by the substantial impact of kernel selection on SVM performance and the role of regularization in preventing overfitting.
We aim to assess the effects of different kernels, with and without regularization, on SVM models, utilizing the cost parameter to balance training error and model simplicity. This investigation seeks a nuanced understanding of how kernel choices and regularization techniques shape SVM performance.

Importance:

- **Optimal Model Generalization:** Selecting the right kernel and regularization parameters is crucial for achieving a balance between fitting the training data well and preventing the model from being overly complex, ensuring effective generalization to new, unseen data.
- **Robust Performance:** Understanding the impact of different kernels allows tailoring the model to specific data characteristics, enhancing its robustness and adaptability to varying degrees of complexity and non-linearity.
- **Avoiding Overfitting:** Regularization is fundamental to prevent overfitting, ensuring the model captures underlying patterns in the data rather than memorizing noise.
- **Interpretability and Computational Efficiency:** Different kernels have implications on the interpretability of the model and its computational efficiency, necessitating a balance between interpretability and computational cost.

By comprehensively exploring the impact of different kernels and regularization, we aim to provide insights guiding practitioners in making informed decisions when applying SVMs to classification tasks, contributing to the broader field of machine learning research and application.

2 Dataset

The Shape Sets dataset, part of the Clustering Basic Benchmark from the University of Eastern Finland, is a collection of synthetic datasets tailored for evaluating clustering algorithm performance. Designed by Fränti and colleagues, these datasets exhibit diverse characteristics such as varied cluster shapes, sizes, and degrees of separation. As a versatile benchmark, Shape Sets allows researchers and practitioners to assess clustering algorithm effectiveness across a spectrum of scenarios.

Shape Sets Composition:

1. **Aggregation:** This dataset comprises 788 vectors distributed across 7 clusters in 2D space, presenting aggregated shapes challenging clustering algorithms with non-convex structures (Source: A. Gionis, H. Mannila, and P. Tsaparas).

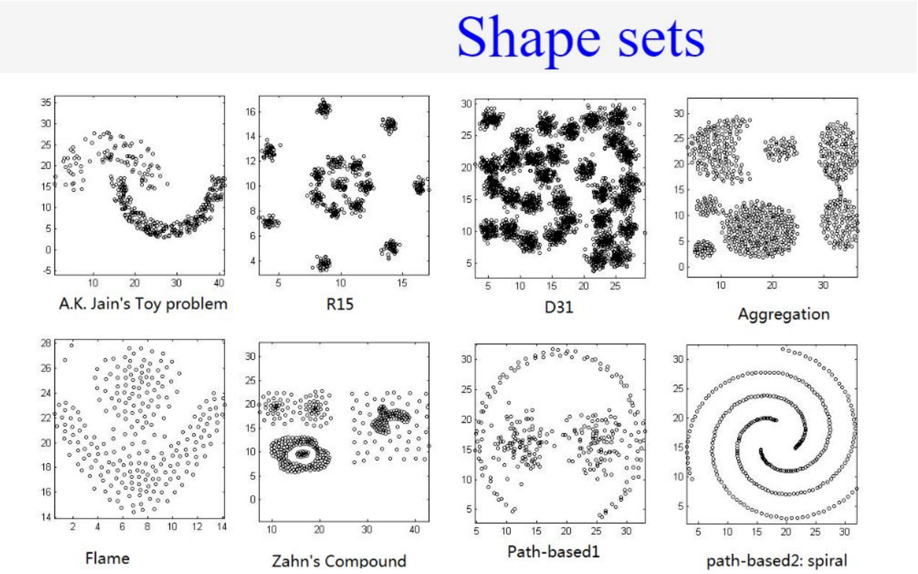
2. **Compound:** With 399 vectors across 6 clusters in 2D space, the Compound dataset challenges clustering algorithms to detect complex patterns (Source: C.T. Zahn).

3. **Flame:** Consisting of 240 vectors among 2 clusters in 2D space, Flame challenges clustering algorithms to identify flame-shaped clusters (Source: L. Fu and E. Medico).

4. **Jain:** This dataset includes 373 vectors distributed into 2 clusters in 2D space, evaluating clustering algorithms' capability to handle clusters with distinct shapes (Source: A. Jain and M. Law).

5. **Pathbased:** Comprising 300 vectors organized into 3 clusters in 2D space, Pathbased assesses clustering algorithms' ability to identify clusters along paths (Source: H. Chang and D.Y. Yeung).

6. **Spiral:** With 312 vectors across 3 clusters in 2D space, Spiral evaluates clustering algorithms' performance on spiral-shaped clusters (Source: H. Chang and D.Y. Yeung).



A.K. Jain's Toy problem

R15

D31

Aggregation

Flame

Zahn's Compound

Path-based1

path-based2: spiral

Figure 1: Shape Sets

Significance: The Shape Sets dataset serves as a versatile evaluation platform for clustering algorithms, enabling a thorough assessment of their robustness across various cluster shapes and configurations. Particularly valuable for scenarios with non-linear and intricate structures, this dataset empowers researchers to understand algorithmic behavior, identify strengths, and acknowledge limitations. It provides valuable insights for informed decision-making when selecting clustering algorithms for real-world applications.

3 Methodology

Data Pre-processing: The initial step involves loading the Shape Sets dataset from the provided text files using the 'load' function. This function extracts the necessary information, such as features of the data set and its labels, for subsequent analysis.

Model Parameters:

- **Different Kernels Without Regularization:** In this phase, three distinct kernels—Linear, Polynomial, and RBF—were applied without regularization. To nullify the impact of regularization, the regularization parameter (C) was set to a large value (10^6), making its influence negligible.
- **Use Kernels With Regularization:** Identifying the RBF kernel as the most effective from the previous analysis, we further explored its performance with different levels of regularization. Two sets of experiments were conducted: one with a soft margin (small C values - 0.01 and 0.1) and another with a hard margin (large C values - 10 and 100).
- **GridSearchCV :** Optimal hyperparameters for the SVM model are determined using GridSearchCV. The 'rbf' kernel is exclusively considered, and parameters (C and gamma) are tuned within specific ranges. The best hyperparameters and model score are printed, and the tuned parameters are used to create an SVM model for evaluation on the test data.

Model Evaluation: Model performance was assessed using accuracy as the metric, providing a straightforward measure of classification effectiveness. Results for each dataset were recorded, allowing a comprehensive comparison of various kernels and regularization settings. Evaluation involved visualizing data, and for Grid Search, the best score and resulting accuracy were calculated by comparing with the test dataset.

This systematic approach ensures a thorough exploration of the Shape Sets dataset, considering various kernels, regularization strategies, and hyperparameter tuning techniques. The methodology allows for a nuanced understanding of SVM model behavior under different conditions, facilitating informed decisions for practical applications.

4 Implementation

In the analysis, I utilized the Python programming language and several powerful libraries to implement and evaluate Support Vector Machine (SVM) models on the Shape Sets dataset. The primary tools and libraries employed are as follows:

1. **Python:** Python served as the core programming language, offering flexibility and ease of use for machine learning tasks.

2. **NumPy:** NumPy was employed for efficient numerical operations and array manipulations, providing a foundation for data handling.

3. **scikit-learn:** scikit-learn, a comprehensive machine learning library in Python, was extensively utilized for SVM implementation, model evaluation, and hyperparameter tuning.

4. **matplotlib:** The matplotlib library was instrumental in creating visualizations, such as scatter plots and decision boundaries, enabling a clear understanding of the data and model behavior.

5. **mlxtend:** mlxtend, a library built on top of scikitlearn, provided the 'plotdecisionregions' function for visualizing decision boundaries in SVM models.

Data Pre-processing: The 'file_load' function, implemented using NumPy, facilitated the loading of data from text files, and extracted it as a form of data array to process it the analysis.
Model Implementation: The SVM models were implemented using the scikit-learn library, specifically the 'SVC' class, allowing flexibility in choosing different kernels (Linear, Polynomial, RBF) and setting hyperparameters such as the regularization parameter (C).
Hyperparameter Tuning: The 'SVM_tune' function utilized GridSearchCV from sci-kit-learn to systematically search hyperparameter space and identify the optimal combination for the SVM model. The hyperparameters tuned included C (regularization parameter) and γ, and the 'rbf' kernel was exclusively considered.
Model Evaluation: The model's accuracy was evaluated using sci-kit-learn's 'accuracy_score' metric on the test data. Additionally, decision boundaries and classification regions were visualized using the 'plot_decision_regions' function from mlxtend, providing an intuitive understanding of the model's performance.
Results Visualization: Matplotlib was employed to create visualizations of the original data, SVM decision boundaries, and the best-performing SVM model. These visualizations aid in interpreting the results and gaining insights into the behavior of the SVM models.

This comprehensive implementation and toolset allowed us to systematically explore the impact of different kernels and regularization strategies on the Shape Sets dataset, providing a robust foundation for analysis and interpretation.

5 Conclusion

Our SVM implementation on the Shape Sets dataset offered key insights:

1. **Model Flexibility:** SVMs demonstrated adaptability through varied kernels and regularization settings, showcasing versatility.

2. **Hyperparameter Sensitivity:** GridSearchCV emphasized the need for careful tuning in regularization and kernel selection.

3. **Visualization Impact:** 'mlxtend' visualizations were crucial for understanding decision boundaries and SVM classifications.

4. **Performance Assessment:** Rigorous accuracy-based evaluation provided a comprehensive understanding of SVM effectiveness.

Learnings:

- SVM implementation provided hands-on insights into model complexity and generalization trade-offs.

- Hyperparameter tuning underscored the impact of regularization, necessitating a systematic approach.

Improvement Tips:

- Experiment with additional datasets for broader insights.

- Explore advanced SVM techniques like ensemble methods for enhanced robustness.

- Investigate feature engineering to uncover potential model performance improvements.

In summary, our implementation provided valuable insights into SVMs, their applicability, and the importance of thoughtful hyperparameter tuning. Continued exploration with diverse datasets and advanced techniques is recommended for further refinement.