

Comparison between Dissimilarity and Spread as Metrics in the Similarity Space for Hyperparameter Tuning of Support Vector Machines

1st Eduardo Henrique Basilio de Carvalho
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
eduardohbc@ufmg.br

2nd Joao Vitor Braga da Silva Alves
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
jvbsa@ufmg.br

Abstract—The performance of Support Vector Machines (SVMs) with Radial Basis Function (RBF) kernels is highly dependent on the choice of hyperparameters, particularly the kernel width. Traditional tuning methods like grid search with cross-validation are computationally expensive. This paper explores alternative tuning strategies based on the geometric properties of data in a kernel-induced similarity space. We implement and evaluate two such methods: one using the established “dissimilarity” metric and a novel approach using a “spatial spread” metric. These methods aim to find an optimal kernel width by maximizing class separability in the similarity space. We compare their performance in terms of accuracy and computation time against a standard, non-tuned SVM implementation from the scikit-learn library across twelve datasets. Our results indicate that while the proposed tuning methods are significantly more time-consuming, they do not offer a statistically significant improvement in accuracy over the baseline model. This suggests that for the datasets tested, the default heuristics of standard libraries provide a competitive and far more efficient solution.

Index Terms—Support Vector Machines, Kernel Methods, Hyperparameter Tuning, Similarity Space, Dissimilarity Metric

I. INTRODUCTION

Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks [6]. Their effectiveness is greatly enhanced by the use of kernel functions, which enable them to model complex, non-linear relationships by implicitly mapping data to a higher-dimensional feature space. Among the most popular choices is the Radial Basis Function (RBF) kernel, defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (1)$$

where $\gamma = 1/(2h^2)$ is a hyperparameter that controls the width of the kernel. The performance of an RBF kernel SVM is critically sensitive to the choice of γ .

A common approach for tuning this hyperparameter is to perform an exhaustive grid search combined with k-fold cross-validation. While effective, this method is computationally intensive and its cost grows exponentially with the number of hyperparameters to be tuned.

An alternative paradigm, proposed by Menezes et al. [1], suggests that an optimal kernel width can be determined by

analyzing the structure of the data in a “similarity space” induced by the kernel itself. This approach aims to find a kernel width that maximizes a metric of class separability, avoiding the need for repeated model training inherent to cross-validation.

This paper investigates two such metrics. The first is the **dissimilarity** metric, as proposed in [1]. The second is a novel metric we term **spatial spread**. We implement SVM classifiers that use these metrics to optimize the RBF kernel width. We then conduct a comparative analysis of these two models against a standard, out-of-the-box **Sklearn’s Standard SVC** [2], evaluating their performance across a variety of datasets. The goal is to provide an honest and fair comparison of njihovich respective effectiveness and computational cost.

II. DISSIMILARITY SVC

The Dissimilarity SVC is based on the methodology presented in [1]. The core idea is to project the dataset into a 2D similarity space and find the kernel width h that maximizes the separability between the classes in this new space.

First, the likelihood of an arbitrary point \mathbf{x} to a given class C_j is defined as the average similarity between that point and all N_j samples belonging to class C_j :

$$B(\mathbf{x}, C_j) = \frac{1}{N_j} \sum_{\mathbf{x}_l \in C_j} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_l\|^2}{2h^2}\right) \quad (2)$$

Using this, we can define the similarity between two classes, C_i and C_j , as the average likelihood of samples from C_i with respect to C_j :

$$S_{ij} = \text{Sim}(C_i, C_j) = \frac{1}{N_i} \sum_{\mathbf{x}_k \in C_i} B(\mathbf{x}_k, C_j) \quad (3)$$

For a binary classification problem (classes 0 and 1), this leads to two vectors in the similarity space, $\mathbf{V}_0 = [S_{00}, S_{01}]$ and $\mathbf{V}_1 = [S_{10}, S_{11}]$, which represent the mean points of the mapped samples for each class.

The dissimilarity metric is then defined as the product of the Euclidean distance and the cosine of the angle between these two vectors:

$$\mathcal{D}(\mathbf{V}_0, \mathbf{V}_1) = \|\mathbf{V}_0 - \mathbf{V}_1\| \cdot \cos(\mathbf{V}_0, \mathbf{V}_1) \quad (4)$$

The Dissimilarity SVC finds the optimal hyperparameter h^* by maximizing this function. Our implementation achieves this by minimizing its negative. Once h^* is found, a final SVM is trained using the precomputed kernel matrix $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2(h^*)^2})$.

III. SPREAD SVC

The Spread SVC is a novel approach that also operates in the similarity space. The data is first projected into a 2D space where the coordinates of each sample \mathbf{x}_i are given by its total similarity to class 0 and class 1, respectively. Let $Q_0(\mathbf{x}_i) = \sum_{\mathbf{x}_k \in C_0} K(\mathbf{x}_i, \mathbf{x}_k)$ and $Q_1(\mathbf{x}_i) = \sum_{\mathbf{x}_l \in C_1} K(\mathbf{x}_i, \mathbf{x}_l)$. The projected sets of points for each class are:

$$\mathcal{C}_0 = \{[Q_0(\mathbf{x}_i), Q_1(\mathbf{x}_i)] | \mathbf{x}_i \in C_0\} \quad (5)$$

$$\mathcal{C}_1 = \{[Q_0(\mathbf{x}_i), Q_1(\mathbf{x}_i)] | \mathbf{x}_i \in C_1\} \quad (6)$$

We then define the average intra-class distance for a class \mathcal{C}_k and the average inter-class distance between \mathcal{C}_0 and \mathcal{C}_1 :

$$d_{intra}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|(|\mathcal{C}_k| - 1)} \sum_{\mathbf{p}_i, \mathbf{p}_j \in \mathcal{C}_k, i \neq j} \|\mathbf{p}_i - \mathbf{p}_j\| \quad (7)$$

$$d_{inter}(\mathcal{C}_0, \mathcal{C}_1) = \frac{1}{|\mathcal{C}_0||\mathcal{C}_1|} \sum_{\mathbf{p}_i \in \mathcal{C}_0, \mathbf{p}_j \in \mathcal{C}_1} \|\mathbf{p}_i - \mathbf{p}_j\| \quad (8)$$

The Spatial Spread metric is defined as:

$$\mathcal{S} = \frac{d_{intra}(\mathcal{C}_0) + d_{intra}(\mathcal{C}_1)}{2} - |d_{intra}(\mathcal{C}_0) - d_{intra}(\mathcal{C}_1)| + d_{inter}(\mathcal{C}_0, \mathcal{C}_1) \quad (9)$$

This metric rewards mappings where classes are compact and well-separated, while penalizing mappings where the dispersion of the two classes is very different. The Spread SVC finds the optimal h^* by maximizing \mathcal{S} and then proceeds to train the final model with the precomputed kernel.

IV. SKLEARN'S STANDARD MODEL

The baseline for our comparison is the standard Support Vector Classifier (SVC) from the popular scikit-learn library [2]. This model is used with its default hyperparameter configuration. Notably, the RBF kernel width is determined by the default `gamma='scale'` option, which sets $\gamma = 1/(n_{\text{features}} \cdot X.\text{var}())$. The regularization parameter is set to $C=1.0$.

It serves as a benchmark representing a typical, non-expert application of the SVM, highlighting the trade-off between the computational cost of custom tuning and the performance of a robust, general-purpose default.

V. METHODOLOGY

The three classifiers were compared on 12 publicly available binary classification datasets from the UCI Machine Learning Repository [3]. Before training, all datasets were preprocessed using a pipeline consisting of `StandardScaler` to normalize features and `PCA` for dimensionality reduction.

The performance of each classifier was evaluated using 5-fold cross-validation, recording the accuracy on each fold. The mean accuracy and standard deviation over the five folds are reported. The total wall-clock time required to complete the 5-fold cross-validation for each model on each dataset was also measured.

To determine if the performance differences between the models were statistically significant, we employed the Wilcoxon signed-rank test. This non-parametric statistical test was applied to the paired accuracy scores from the cross-validation folds for each pair of classifiers. A p-value greater than 0.05 was taken to indicate that the null hypothesis (that the two models have equivalent performance) cannot be rejected.

VI. RESULTS

The accuracy and timing results for the three classifiers across all datasets are presented in Table I and Table II, respectively. The p-values from the Wilcoxon signed-rank tests are shown in Table III.

VII. DISCUSSION

The results tables reveal several key insights. As shown in Table II, the Dissimilarity and Spread SVCs are consistently and significantly slower than the Sklearn's Standard SVC. This is expected, as they perform an explicit optimization procedure to find the kernel width h for each of the 5 cross-validation folds, whereas the standard model uses a fast, non-iterative heuristic. For some datasets like `Digits`, `5 vs rest` and `Titanic`, the custom methods are orders of magnitude slower.

In terms of accuracy (Table I), no single method is a clear winner across all datasets. While the custom methods occasionally produce a higher mean accuracy (e.g., Spread SVC on the `diabetes` dataset), they also sometimes perform significantly worse (e.g., both on the `sonar` dataset). The standard SVC provides robust and competitive performance throughout.

Most importantly, the statistical analysis in Table III shows that in nearly every comparison, the null hypothesis of model equivalence cannot be rejected (p-value ≥ 0.05). This is a crucial finding. It suggests that despite the sophisticated optimization and the vastly increased computational cost, neither the Dissimilarity SVC nor the Spread SVC provides a statistically significant improvement in classification accuracy over the simple, default Sklearn's Standard SVC.

TABLE I
MODEL ACCURACY (MEAN \pm STD. DEV.) ACROSS DATASETS

Dataset	Sklearn's Standard SVC	Dissimilarity SVC	Spread SVC
blood-transfusion-service-center	0.759 \pm 0.038	0.769 \pm 0.132	0.763 \pm 0.130
breast_cancer	0.974 \pm 0.015	0.974 \pm 0.018	0.958 \pm 0.035
diabetes	0.772 \pm 0.025	0.775 \pm 0.030	0.781 \pm 0.041
digits_binary_0_vs_1	0.992 \pm 0.011	0.992 \pm 0.017	0.992 \pm 0.017
digits_binary_5_vs_rest	0.992 \pm 0.002	0.991 \pm 0.003	0.988 \pm 0.003
ionosphere	0.949 \pm 0.032	0.940 \pm 0.024	0.883 \pm 0.079
iris_binary_setosa_vs_rest	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000
iris_binary_setosa_vs_versicolor	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000
qsar-biodeg	0.868 \pm 0.038	0.820 \pm 0.104	0.704 \pm 0.303
sonar	0.639 \pm 0.095	0.432 \pm 0.055	0.322 \pm 0.094
sylvine	0.997 \pm 0.006	0.997 \pm 0.006	0.997 \pm 0.006
titanic	0.714 \pm 0.045	0.651 \pm 0.147	0.619 \pm 0.144

TABLE II
TOTAL EXECUTION TIME (SECONDS) FOR 5-FOLD CROSS-VALIDATION

Dataset	Sklearn's Standard SVC	Dissimilarity SVC	Spread SVC
blood-transfusion-service-center	0.031	0.782	1.569
breast_cancer	0.042	0.290	0.654
diabetes	0.085	0.710	1.684
digits_binary_0_vs_1	0.115	1.149	1.941
digits_binary_5_vs_rest	0.144	8.728	8.301
ionosphere	0.099	0.772	0.954
iris_binary_setosa_vs_rest	0.021	0.045	0.045
iris_binary_setosa_vs_versicolor	0.010	0.017	0.024
qsar-biodeg	0.065	0.710	1.702
sonar	0.120	1.104	1.190
sylvine	1.508	2.191	1.408
titanic	5.262	10.187	10.294

TABLE III
STATISTICAL EQUIVALENCE (P-VALUES FROM WILCOXON TEST)

Dataset	Standard vs. Dissimilarity	Standard vs. Spread	Dissimilarity vs. Spread
blood-transfusion-service-center	1.000	1.000	0.750
breast_cancer	1.000	0.688	0.500
diabetes	0.750	0.625	0.625
digits_binary_0_vs_1	1.000	1.000	1.000
digits_binary_5_vs_rest	1.000	0.250	0.375
ionosphere	0.688	0.188	0.188
iris_binary_setosa_vs_rest	1.000	1.000	1.000
iris_binary_setosa_vs_versicolor	1.000	1.000	1.000
qsar-biodeg	1.000	0.750	0.250
sonar	0.063	0.063	0.188
sylvine	1.000	1.000	1.000
titanic	0.438	0.313	0.063

VIII. CONCLUSION

This paper presented a comparative study of three SVC models: two featuring hyperparameter optimization based on similarity space metrics (Dissimilarity and Spatial Spread) and a standard library implementation with default parameters. The experiment aimed to determine if these data-driven tuning methods could offer a practical advantage over a baseline approach.

Our findings indicate a clear trade-off. The Dissimilarity and Spread SVCs require substantially more computation time

due to their optimization loops. However, this investment of resources did not translate into a consistent or statistically significant increase in accuracy on the twelve datasets tested. The far simpler and faster Sklearn's Standard SVC proved to be a remarkably strong baseline, achieving statistically equivalent results in almost all cases.

We conclude that for many common classification problems, the well-chosen heuristics of modern machine learning libraries like scikit-learn are highly effective, and more complex tuning strategies must demonstrate a significant performance

gain to justify their computational expense. Future work could involve testing these metrics on a wider range of more complex datasets or exploring their utility for other kernel types.

REFERENCES

- [1] M. V. F. Menezes, L. C. B. Torres, and A. P. Braga, Width optimization of RBF kernels for binary classification of support vector machines: A density estimation-based approach," *Pattern Recognition Letters*, vol. 128, pp. 1-7, 2019.
- [2] F. Pedregosa et al., Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [3] D. Dua and C. Graff, UCI Machine Learning Repository," Irvine, CA: University of California, School of Information and Computer Science, 2019. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [4] I-Cheng Yeh, King-Jang Yang and Ting-Ting Hsu, Building real-time warning systems for residents in disaster-prone areas," *Life Information*, 2009. (for blood-transfusion-service-center)
- [5] O. L. Mangasarian and W. H. Wolberg, Cancer diagnosis and prognosis via linear programming," *SIAM News*, vol. 23, no. 5, pp. 1-18, 1990. (for breast_cancer)
- [6] C. Cortes and V. Vapnik, Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
- [7] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus." In *Proceedings of the Annual Symposium on Computer Application in Medical Care* (p. 261). American Medical Informatics Association. (for diabetes)
- [8] R. A. Fisher, The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179-188, 1936. (for iris)
- [9] V. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262-266, 1989. (for ionosphere)
- [10] R. P. Gorman and T. J. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Networks*, vol. 1, no. 1, pp. 75-89, 1988. (for sonar)
- [11] F. A. D. de Campos, qsar-biodeg. Data Set," UCI Machine Learning Repository, 2011. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation>
- [12] E. Alpaydin and C. Kaynak, Optical Recognition of Handwritten Digits," UCI Machine Learning Repository, 1998. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits>
- [13] I. Guyon, sylvine. Data Set," UCI Machine Learning Repository, 2008. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/sylvine>
- [14] F. E. Harrell, Jr., Titanic: Machine Learning from Disaster," Kaggle, 2012. [Online]. Available: <https://www.kaggle.com/c/titanic>