Mathematical modeling

Математическое моделирование

UDC 004.8, 004.852
https://doi.org/10.32362/2500-316X-2026-14-1-82-90
EDN KCUBAG

RESEARCH ARTICLE

Feature space transformation in the support vector method

Aleksey V. Fedorov [@], **Denis V. Parfenov**

MIREA – Russian Technological University, Moscow, 119454 Russia
[@] *Corresponding author, e-mail: fedorov_av@mirea.ru*

• **Submitted:** 22.05.2025 • **Revised:** 02.07.2025 • **Accepted:** 10.11.2025

Abstract

Objectives. This study focuses on the development and investigation of a generalized nonlinear Support Vector Machine (SVM) method incorporating an adaptive transformation of the feature space. Its aim is to improve computational efficiency while maintaining high classification accuracy. The binary classification problem is used as a case study. The main objective of the research is to quantitatively evaluate the performance of the proposed approach when compared to classical SVM models using fixed kernel functions, and to analyze how the transformation parameters affect classification quality.

Methods. The proposed approach involves a preliminary transformation of the input data using a learnable nonlinear mapping with a fixed structure. This mapping is implemented as a composition of elementary functions and is parameterized by a limited number of trainable weights which allows control over model complexity. A linear SVM with L2 regularization is applied after the transformation. The model is trained using conventional, unconstrained numerical optimization methods. The classification quality is evaluated using the Accuracy metric averaged over 10-fold cross-validation. The work also studies the behavior of the model with varying feature space dimensionality. In addition, computational complexity is analyzed in terms of the number of operations and inference time required on test datasets.

Results. Numerical experiments demonstrate that the proposed model significantly reduces classification time when compared to a polynomial-kernel SVM, while maintaining a comparable level of accuracy. The runtime analysis confirms that the proposed approach scales much better than traditional kernel methods. At the same time, the structure of the model remains interpretable and can be further adapted to the specifics of the application domain.

Conclusions. The method developed provides an efficient alternative to traditional kernel-based algorithms. Through the use of a parameterized transformation of the feature space, the method enables adaptability, interpretability, and scalability, making it promising for practical applications in machine learning tasks.

Keywords: classification, feature space transformation, nonlinear mapping, nonlinear support vector machine, kernel functions, computational complexity

For citation: Fedorov A.V., Parfenov D.V. Feature space transformation in the support vector method. *Russian Technological Journal.* 2026;14(1):82–90. https://doi.org/10.32362/2500-316X-2026-14-1-82-90, https://www.elibrary.ru/KCUBAG

Financial disclosure: The authors have no financial or proprietary interest in any material or method mentioned.

The authors declare no conflicts of interest.

НАУЧНАЯ СТАТЬЯ

Трансформация пространства признаков в методе опорных векторов

А.В. Федоров [@], **Д.В. Парфенов**

МИРЭА – Российский технологический университет, Москва, 119454 Россия
[@] *Автор для переписки, e-mail: fedorov_av@mirea.ru*

• **Поступила:** 22.05.2025 • **Доработана:** 02.07.2025 • **Принята к опубликованию:** 10.11.2025

Резюме

Цели. Работа посвящена разработке и исследованию обобщенного нелинейного метода опорных векторов (support vector machine, SVM) с использованием адаптивной трансформации пространства признаков, направленного на улучшение вычислительной эффективности при сохранении высокого качества классификации. В качестве задачи-примера рассматривается двухклассовая классификация. Целью исследования является количественная оценка производительности предложенного подхода в сравнении с классическими SVM-моделями, использующими фиксированные ядровые функции, а также изучение влияния параметров трансформации на качество классификации.

Методы. Предлагается модифицированный подход, при котором входные данные предварительно преобразуются с помощью обучаемого нелинейного отображения фиксированной структуры. Это отображение реализуется в виде композиции элементарных функций и параметризуется ограниченным числом обучаемых весов, что обеспечивает контроль над сложностью модели. После трансформации применяется линейный SVM с L2-регуляризацией. Для обучения модели используются стандартные методы численной оптимизации без ограничений. Качество классификации оценивается с помощью метрики точности (Accuracy), усредненной по результатам 10-кратной перекрестной валидации. Рассматривается поведение модели при изменении размерности признакового пространства. Проводится анализ вычислительной сложности по числу операций и времени применения модели на тестовых выборках.

Результаты. Численные эксперименты показали, что предложенная модель позволяет существенно сократить время классификации по сравнению с SVM с полиномиальным ядром, обеспечивая при этом сопоставимое качество. Анализ временных затрат подтвердил, что предложенный подход масштабируется значительно лучше, чем классические ядровые методы. При этом структура модели сохраняет интерпретируемость и может быть дополнительно адаптирована под особенности предметной области.

Выводы. Разработанный метод представляет собой эффективную альтернативу классическим ядровым алгоритмам. Благодаря параметризуемому отображению признакового пространства он обеспечивает адаптивность, интерпретируемость и масштабируемость, что делает его перспективным для практического применения в задачах машинного обучения.

Ключевые слова: классификация, преобразование признакового пространства, нелинейное отображение, нелинейный метод опорных векторов, функции ядра, вычислительная сложность

Для цитирования: Федоров А.В. Парфенов Д.В. Трансформация пространства признаков в методе опорных векторов. *Russian Technological Journal*. 2026;14(1):82–90. https://doi.org/10.32362/2500-316X-2026-14-1-82-90, https://www.elibrary.ru/KCUBAG

Прозрачность финансовой деятельности: Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

INTRODUCTION

A fundamental problem in machine learning (ML) is addressing the challenge of classifying linearly inseparable data. A traditional method for tackling this involves employing a generalized nonlinear support vector machine (SVM) with kernel functions. These kernels facilitate the computation of scalar products in a transformed space which offers improved geometric properties for class separation [1]. However, this approach has certain limitations. The commonly used kernels often have a limited number of parameters, restricting their flexibility. Furthermore, the variety of kernels utilized in practice is relatively small, which reduces the adaptability of the model.

The paper introduces an alternative approach which differs from traditional kernel methods by enabling the direct optimization of nonlinear transformation parameters within the feature space during SVM training. The proposed transformation is divided into two stages. Initially, the input features undergo a linear mapping defined by a matrix containing trainable coefficients. Subsequently, one-dimensional nonlinear transformations, executed through polynomials with trainable coefficients, are applied to each component of the resulting vector. This approach offers the flexibility to tailor the characteristic space to specific data needs while preserving high computational efficiency.

The primary benefit of the proposed model lies in its considerable reduction in computational costs when utilizing the pre-trained model, compared to approaches reliant on kernel functions. In traditional SVMs which use kernel functions, the computational complexity depends heavily on the number of support vectors. In real-world scenarios these can be quite large, thereby substantially increasing classification time. In contrast, the proposed approach performs data transformation with a fixed computational complexity determined solely by the dimensions of the input and output spaces and the selected polynomial order. This feature makes the method especially efficient for processing large datasets.

Recent efforts have focused on enhancing the effectiveness and flexibility of classification techniques in ML applications. For example, study [2] presents a robust SVM model enhanced with a customized optimization approach to ensure resistance against noise in the data. Research [3] establishes the theoretical basis for multiscale, adaptive feature extraction, proposing it as a viable alternative to traditional kernel-based approaches. Study [4] investigates the geometric interpretation of adaptable feature transformations in neural networks, in line with the idea of a parametrizable representation of feature spaces. Furthermore, study [5] introduces a method involving the diagonal expansion of parameters within a Hilbert space endowed with a reproducing kernel. This facilitates the simultaneous training of both the feature structures and the kernel parameters. Finally, the adaptive law-based transformation (ALT) method discussed in [6] is designed for adaptive feature transformations, specifically targeting time series classification. It bears conceptual similarities to the approach presented in this paper.

1. TRANSFORMATION OF THE FEATURE SPACE

As previously mentioned, each object in the original dataset, denoted as $\mathbf{x} \in \mathbb{R}^n$ (where n represents the dimensionality of the original space), is transformed through the $m \times n$ matrix $\mathbf{A}$. The resultant vector $\tilde{\mathbf{x}} \in \mathbb{R}^m$ is then calculated using the following equation:

$$\tilde{\mathbf{x}} = \mathbf{A}\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}.$$

The dimension m, chosen by the user, specifies the number of features in the transformed space, influencing whether the data space is expanded or reduced. Matrix $\mathbf{A}$ contains trainable parameters, enabling adjustments to align the linear transformation with the specific data structure.

For each element in the resultant vector $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m)^{\mathrm{T}}$, a nonlinear function is applied in the form of polynomial $p_i(\tilde{x}_i)$ of degree d, where $i = \overline{1,m}$. The resultant vector, $\tilde{\tilde{\mathbf{x}}} \in \mathbb{R}^m$, is then calculated as follows:

$$\tilde{\tilde{\mathbf{x}}} = \begin{bmatrix} p_1(\tilde{x}_1) \\ p_2(\tilde{x}_2) \\ \dots \\ p_3(\tilde{x}_m) \end{bmatrix}.$$

The polynomials $p_i(\cdot)$ may vary for each coordinate, providing high flexibility within the model. The general form of polynomial $p_i(\tilde{x})$ is defined as follows:

$$p_i(\tilde{x}) = c_{i1}\tilde{x} + c_{i2}\tilde{x}^2 + \dots + c_{id}\tilde{x}^d,$$

wherein c_{ij} represents trainable parameters, and d denotes the fixed-degree polynomial.

The inclusion of nonlinear functions enables the adjustment of the feature space geometry, facilitating a more effective separation of classes. A key aspect of the proposed classifier model lies in the simultaneous training of both the feature space transformation parameters and the support vector method.

The objective of training the model can be outlined as follows:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max\left(1 - y_i\left(\left\langle \mathbf{w}, \Phi_{\mathbf{A},\mathbf{P}}(\mathbf{x}_i)\right\rangle + b\right), 0\right)^2 \to \min_{\mathbf{w},b,\mathbf{A},\mathbf{P}}.$$

Here, $\mathbf{x}_i \in \mathbb{R}^n$ represents the input vectors; $y_i \in \{-1; +1\}$ are the associated class labels. $\Phi_{\mathbf{A},\mathbf{P}}(\mathbf{x})$ represents the original vector $\mathbf{x}$ after the linear transformation $\mathbf{A}$ and the application of polynomial functions p_i; $\mathbf{A}$ and $\mathbf{P}$ are transformation parameters for the linear and polynomial operations, respectively; $\mathbf{w}$ and b are linear classifier parameters. C represents the penalty weight assigned to deviations, while N indicates the total number of training samples included in the dataset.

The proposed approach incorporates L2 regularization [7, 8], crucial for stabilizing the training process and mitigating the risk of overtraining. This regularization is applied not only to the parameters of the linear transformation matrix but also to the coefficients of the polynomial mapping. By employing L2 regularization, the growth of weight norms is restricted, thereby decreasing the probability of overtraining. Numerical experiments have demonstrated that this regularization technique supports effective learning outcomes. Consequently, the objective function has the following form:

$$\sum_{i=1}^{N}\max\left(1 - y_i\left(\left\langle \mathbf{w}, \Phi_{\mathbf{A},\mathbf{P}}(\mathbf{x}_i)\right\rangle + b\right), 0\right)^2 + \\ + \frac{1}{2C}\|\mathbf{w}\|^2 + \lambda_1\|\mathbf{A}\|^2 + \lambda_2\|\mathbf{P}\|^2 \to \min_{\mathbf{w},b,\mathbf{A},\mathbf{P}}, \quad (1)$$

wherein λ_1 and λ_2 are regularization coefficients.

In order to address problem (1), an optimization method without BFGS[1] constraints is utilized [9], provided through the *Optim.jl* package[2]. The gradients of the objective function are computed using automatic differentiation based on dual numbers [10, 11], facilitated by the *ForwardDiff.jl* package.

2. BENCHMARKING

Benchmarking involves evaluating the performance of the classifier using datasets from the Curated Classification Benchmark OpenML-CC18 [12]. These datasets are specifically curated for comparing various ML algorithms. Table 1 provides an overview of the datasets utilized, highlighting their key attributes such as the number of objects, the dimension of the feature space, and concise descriptive details.

All tasks involve binary classification, meaning each task contains exactly two categories.

In order to evaluate model performance, 10-fold cross-validation [13] is implemented, ensuring a more dependable measurement of classifier quality.

Table 1. Overview of dataset characteristics

No.	Dataset	Number of objects	Dimension	Summary
1	blood-transfusion-service-center	748	4	Prediction of blood donation behavior in the future is required (2 categories)
2	phoneme	5404	5	Classification of speech sounds into two categories is required (2 categories)
3	diabetes	768	8	Detection of diabetes presence or absence is required (2 categories)
4	qsar-biodeg	1055	41	Prediction of chemical biodegradability is required (2 categories)
5	kc1	2109	21	Detection of the software module errors is required (2 categories)
6	pc1	1109	21	The task is similar to kc1 that involves classifying software components into “with errors” and “no errors” categories (2 categories)

[1] The Broyden–Fletcher–Goldfarb–Shanno algorithm.

[2] *Optim.jl Documentation. Optim.jl* is Julia optimization library. https://julianlsolvers.github.io/Optim.jl/stable/. Accessed July 07, 2025.

Employing multiple datasets facilitates testing the ability of the model to generalize across diverse inputs. Additionally, using a standardized test dataset ensures comparability with existing methods and published research, providing an objective analysis of the model's strengths and weaknesses.

Prior to training the classifier, all data is subjected to preliminary normalization processes, such as centering and scaling. These steps are essential for enhancing model stability and improving its generalization capabilities. Combined with 10-fold cross-validation and testing across diverse samples, these measures ensure an objective and reliable evaluation of the classifier's quality (Accuracy) [14].

The results of testing this classifier model with adaptive feature space transformation are presented in Table 2.

The proposed model is evaluated against the SVM method utilizing a polynomial kernel, as referenced in studies [15, 16]. In the case of SVM with polynomial kernels, classification takes place within an implicitly defined higher-dimensional space, in which the scalar product is computed directly via the kernel function, thus eliminating the need for explicit construction of the feature transformation. Conversely, the proposed method differentiates itself by explicitly constructing a nonlinear mapping of the original space. This is then subjected to linear separation through the support vector method.

It should be noted that SVM with a polynomial kernel is often regarded as a viable benchmark model. This is because, while the polynomial kernel typically delivers slightly lower classification performance when compared to more advanced kernels such as the radial basis function (RBF), it offers notable computational efficiency. This makes it a practical trade-off between classification accuracy and computational demands. The proposed approach achieves performance comparable to that of the SVM with a polynomial kernel, while simultaneously offering even greater computational efficiency.

Evaluation tests have been conducted where the parameters of the polynomial kernel are selected using a random search strategy over a predefined grid of values. The parameter selection process employs 10-fold cross-validation, ensuring careful optimization of the kernel hyperparameters and facilitating an accurate comparison across models. The test results for the SVM with a polynomial kernel are presented in Table 3.

Table 2. Classifier model testing results

No.	Dataset	Output space dimension (m)	Polynomial degree (d)	Accuracy, %
1	blood-transfusion-service-center	4	3	79.29
2	phoneme	5	3	83.33
3	diabetes	8	3	77.47
4	qsar-biodeg	41	3	85.97
5	kc1	21	3	84.45
6	pc1	21	3	92.24

Table 3. Testing results for SVM with a polynomial kernel

No.	Dataset	Degree (d)	Accuracy, %	Number of support vectors
1	blood-transfusion-service-center	3	77.14	367
2	phoneme	3	83.81	2,074
3	diabetes	2	76.95	397
4	qsar-biodeg	2	86.24	429
5	kc1	3	84.69	643
6	pc1	3	93.06	158

Table 4. Accuracy comparison between the proposed model and SVM with a polynomial kernel

No.	Dataset	Accuracy, %	
		Offered model	Polynomial kernel
1	blood-transfusion-service-center	79.29	77.14
2	phoneme	83.33	83.81
3	diabetes	77.47	76.95
4	qsar-biodeg	85.97	86.24
5	kc1	84.45	84.69
6	pc1	92.24	93.06

Table 4 presents a comparison of the classification accuracy of the proposed model and the SVM with a polynomial kernel on each dataset.

As shown in Table 4, the proposed classifier achieves an accuracy level similar to that of the SVM with a polynomial kernel. This highlights the flexibility of the model and its capacity to adapt effectively to the dataset characteristics.

3. ANALYSIS OF COMPUTATIONAL COMPLEXITY

In many ML tasks, the focus extends beyond just achieving high model accuracy: it also evaluates its computational complexity during classification. This becomes particularly critical in scenarios which demand rapid data processing, such as real-time systems used for object detection, data flow analysis, or diagnostics. Additionally, limited computational resources pose a significant challenge, especially when working with low-power devices such as mobile phones, embedded systems, or microcontrollers. These devices often lack sufficient memory, energy, or processing capabilities, making complex models impractical for deployment. Therefore, when selecting a model, it is crucial to account for its complexity and performance relative to the capabilities of the targeted hardware.

After training the proposed model, the resulting classifier (decision function) is as follows:

$$f(\mathbf{x}) = \left\langle \mathbf{w}, \Phi_{\mathbf{A},\mathbf{P}}(\mathbf{x}) \right\rangle + b.$$

The transformation $\Phi_{\mathbf{A},\mathbf{P}}(\mathbf{x})$ can be reduced to matrix multiplication and polynomial component transformation. Multiplying by matrix $\mathbf{A}$ involves $n \times m$ multiplication operations and $(n-1) \times m$ addition operations. Applying the polynomials $p_i(x)$ necessitates $d \times m$ multiplications along with $(d-1) \times m$ additions. Additionally, the shift dot product entails m more multiplication operations and m added additions.

Consequently, the classification of a single object involves $m \times (n + d + 1)$ multiplication operations and $m \times (n + d - 1)$ addition operations. Thus the computational complexity of the classification process remains independent of the number of support vectors. This represents a fundamental distinction between the proposed approach and the classical SVM, reliant on kernel functions and resolving the dual problem [17]. Leveraging an explicit parameterized features transformation $\Phi_{\mathbf{A},\mathbf{P}}(\mathbf{x})$, eliminates the need to transition to the dual problem, a hallmark of kernel tricks [15, 18]. This approach substantially reduces computational complexity, particularly in scenarios where resolving a dual problem results in a large number of support vectors, a common occurrence with noisy or high-dimensional datasets.

By applying kernel transformation, the dual minimization problem is addressed, leading to the following decision function:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b,$$

where $k(\mathbf{x}_i, \mathbf{x})$ represents the kernel function, while $\mathbf{x}_i$ denotes the support vectors.

The polynomial kernel is expressed as follows:

$$k(\mathbf{x}, \mathbf{x}') = \left(\left\langle \mathbf{x}, \mathbf{x}' \right\rangle + c\right)^d,$$

where c and d refer to the kernel parameters.

The calculation of the polynomial kernel involves $(n + d - 1) \times l$ multiplication operations when dealing with low powers ($d \in \{2; 3\}$), as well as $n \times l$ addition operations.

Here, n represents the space dimensionality (the length of vector x), d denotes the degree of the polynomial kernel, and l refers to the number of support vectors. In order to make a classification decision, the algorithm sums over all reference vectors, entailing l multiplication operations (where the kernel function value is multiplied by the coefficient α_i) and $l + 1$ addition operations. Consequently, the total number of basic operations required for classification is $l \times (n + d)$ multiplication operations and $l \times (n + 1) + 1$ addition operations.

In practical scenarios, the number of support vectors, l, is typically substantial, resulting in a significant rise in computational requirements. In order to preliminarily analyze the computational complexity of the proposed feature space transformation method in comparison to the conventional approach based on kernel functions, the following commonly observed relationships can be considered:

$$l \gg n, l \gg m, l \gg d, m \sim n, d < n.$$

An approach utilizing kernel functions typically demands considerably more calculations for classification purposes at the output stage. The findings are consolidated in Table 5, which outlines the total number of elementary operations necessary for object classification across all datasets.

The results demonstrate that the proposed model performs classification with substantially fewer arithmetic operations, such as additions and multiplications. Consequently, this method is more computationally efficient than when compared to using a polynomial kernel.

4. MODEL PERFORMANCE

The experiment evaluates the time taken to classify a single object from each dataset. The *BenchmarkTools.jl* package[3] is utilized to ensure precise runtime measurement, since it reduces the influence of background processes and provides an accurate

Table 5. Analysis of computational complexity between the proposed model and SVM with a polynomial kernel

No.	Dataset	Number of operations	
		Offered model	Polynomial kernel
1	blood-transfusion-service-center	56	4405
2	phoneme	80	29037
3	diabetes	176	7544
4	qsar-biodeg	3608	36466
5	kc1	1008	29579
6	pc1	1008	7269

Table 6. Performance comparison between the proposed model and SVM with a polynomial kernel

No.	Dataset	Performance, μs	
		Offered model	Polynomial kernel
1	blood-transfusion-service-center	0.356	7.75
2	phoneme	0.384	45.1
3	diabetes	0.422	11.3
4	qsar-biodeg	1.94	21.8
5	kc1	0.822	22.2
6	pc1	0.822	5.25

[3] *BenchmarkTools.jl* Documentation. *BenchmarkTools.jl* Manual. https://juliaci.github.io/BenchmarkTools.jl/stable/manual/. Accessed July 07, 2025.

estimation of computational costs. The measurements are conducted on pre-trained models, with classification times recorded in microseconds. The median runtime across all runs is chosen as the key metric, since it is less affected by outliers and effectively represents the typical model performance. A summary of the results is presented in Table 6 which facilitates a comparison of the classification speeds between the proposed model and the SVM with a polynomial kernel.

The experimental findings indicate that the proposed model provides significantly improved computational performance than when compared to an SVM using a polynomial kernel. For all datasets tested, the time taken to classify a single instance is noticeably shorter in the proposed model. This is in line with theoretical predictions regarding its computational complexity. In contrast to the polynomial kernel, which has a classification complexity depending on the number of support vectors, the proposed method has a fixed computational complexity, making it substantially more efficient for processing large datasets. Analysis of median runtime values reveals that this increase in classification speed is achieved without a notable decline in performance quality, thus positioning the proposed approach as a viable and competitive alternative to kernel-based techniques. Therefore, the experimental findings affirm that the proposed model presents a promising solution for tasks demanding high classification speed while preserving accuracy.

Additional experiments have been conducted using synthetic data, in order to investigate how classification time varies with the dimensionality of the feature space. For these experiments, the linear transformation matrix $\mathbf{A}$ is selected as a square matrix with the same number of rows and columns ($m = n$). This study allows the impact of dimensionality on computation time to be evaluated. The experiments are conducted on generated datasets with varying dimensions, and for each scenario, the median time required for classification of a single object is recorded. The results are visualized (see Figure) to show the increase in computation time as the dimension of the feature space increases.

The results demonstrate that for dimensions $n \leq 600$, the classification time grows at a moderate pace. This suggests that the proposed method can be effectively applied even to problems involving high-dimensional features. Consequently, the model maintains computational efficiency at medium to moderately high dimensions, making it well-suited for a broad spectrum of practical applications.

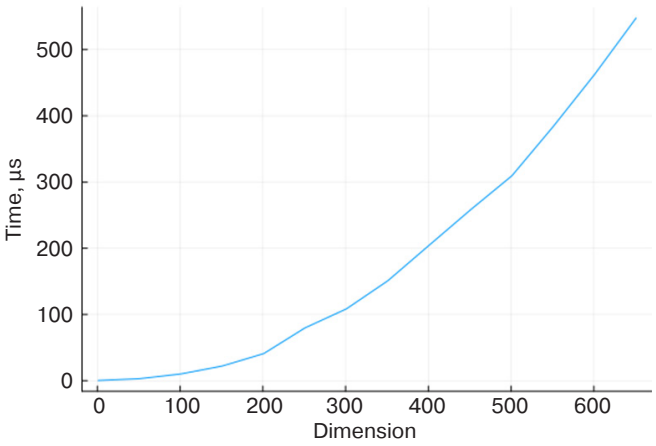

Figure. Classification time versus dimension plot

CONCLUSIONS

The paper introduces a method for transforming the feature space, aimed at effectively classifying linearly inseparable data. Unlike traditional kernel methods, this approach employs an explicit nonlinear feature mapping, with parameters optimized during the training process.

In order to evaluate the method performance, numerical experiments have been conducted. These include comparisons with classical kernel techniques in terms of classification accuracy and computational efficiency. The results demonstrate that the proposed model significantly outperforms SVM with a polynomial kernel in classification speed while maintaining comparable classification quality. Further analysis of runtime dependency on feature space dimensions reveals that for dimensions up to $n \leq 600$, the increase in classification time remains moderate. This indicates that the method is well-suited for handling high-dimensional data effectively.

In summary, the method developed presents a computationally efficient alternative to conventional kernel methods, offering a practical solution for tasks which demand high processing speed without compromising classification accuracy.

Authors' contributions

A.V. Fedorov—development of the model, its detailed study, and formulation of part of the method's idea.

D.V. Parfenov—problem statement, formulation of part of the method's idea, and overall supervision of the work.

REFERENCES

1. Vapnik V. *Statistical Learning Theory*. New York: Wiley; 1998, 736 p. ISBN 978-0-471-03003-4
2. Maggioni F., Spinelli A. A novel robust optimization model for nonlinear Support Vector Machine. *Computers & Operations Research*. 2024;157:105059.
3. Rubin N., Fischer K., Lindner J., Dahmen D., Seroussi I., Ringel Z., Krüger M., Helias M. From Kernels to Features: A Multi-Scale Adaptive Theory of Feature Learning. *arXiv* preprint arXiv:2402.03210. 2024. https://doi.org/10.48550/arXiv.2502.03210
4. LeJeune D., Alemohammad S. An Adaptive Tangent Feature Perspective of Neural Networks. In: *Proceedings of the 37th International Conference on Machine Learning* (*ICML*). 2024. Available from URL: https://proceedings.mlr.press/v234/lejeune24a/lejeune24a.pdf. Accessed July 07, 2025.
5. Li Y., Lin Q. Diagonal Over-parameterization in Reproducing Kernel Hilbert Spaces as an Adaptive Feature Model: Generalization and Adaptivity. *arXiv* preprint arXiv:2501.08679. 2025. https://arxiv.org/abs/2501.08679
6. Kurbucz M.T., Benkő Z., Varga L., et al. Adaptive Law-Based Transformation (ALT): A Fast and Transparent Feature Transformation Method for Time Series Classification. *arXiv* preprint arXiv:2501.09217. 2025. https://arxiv.org/abs/2501.09217
7. Tikhonov A.N., Arsenin V.Y. *Solutions of Ill-Posed Problems*. Washington, D.C.: W.H. Freeman and Co.; 1977, 258 p.
8. Bishop C.M. *Pattern Recognition and Machine Learning*. New York: Springer; 2006, 738 p.
9. Fletcher R. *Practical Methods of Optimization*. New York: John Wiley & Sons; 1987, 464 p.
10. Baydin A.G., Pearlmutter B.A., Radul A.A., Siskind J.M. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research* (*JMLR*). 2018;18(153):1–43.
11. Fisher J. Automatic differentiation with dual numbers. *arXiv* preprint arXiv:2201.00024.
12. Bischl B., Casalicchio G., Feurer M., Hutter F., Lang M., Mantovani R.G., van Rijn J.N., Vanschoren J. OpenML Benchmarking Suites. *arXiv* preprint arXiv:1708.03731v2 [stat.ML], 2019. https://arxiv.org/abs/1708.03731v2
13. Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence* (*IJCAI*). 1995;14(2):1137–1143.
14. Labatut V., Cherifi H. Accuracy Measures for the Comparison of Classifiers. *arXiv* preprint arXiv:1207.3790. https://doi.org/10.48550/arXiv.1207.3790
15. Schölkopf B., Smola A.J. *Learning with Kernels*. Cambridge: MIT Press; 2002, 300 p.
16. Hofmann T., Schölkopf B., Smola A.J. Kernel Methods in Machine Learning. *Ann. Statist*. 2008;36(3):1171–1220. https://doi.org/10.1214/009053607000000677
17. Boser B.E., Guyon I.M., Vapnik V. A Training Algorithm for Optimal Margin Classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (*COLT*). 1992. P. 144–152. https://doi.org/10.1145/130385.130401
18. Cortes C., Vapnik V. Support-Vector Networks. *Mach. Learn*. 1995;20(3):273–297. https://doi.org/10.1007/BF00994018

About the Authors

Aleksey V. Fedorov, Postgraduate Student, Higher Mathematics Department, Institute of Artificial Intelligence, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: fedorov_av@mirea.ru. https://orcid.org/0009-0003-2314-7400

Denis V. Parfenov, Cand. Sci. (Eng.), Associate Professor, Higher Mathematics Department, Institute of Artificial Intelligence, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: parfenov@mirea.ru. Scopus Author ID 57217119805, RSCI SPIN-code 7463-3220, https://orcid.org/0009-0004-0905-3827

Об авторах

Федоров Алексей Викторович, аспирант, кафедра высшей математики, Институт искусственного интеллекта, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: fedorov_av@mirea.ru. https://orcid.org/0009-0003-2314-7400

Парфенов Денис Васильевич, к.т.н., доцент, кафедра высшей математики, Институт искусственного интеллекта, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: parfenov@mirea.ru. Scopus Author ID 57217119805, SPIN-код РИНЦ 7463-3220, https://orcid.org/0009-0004-0905-3827

Translated from Russian into English by Kirill V. Nazarov
Edited for English language and spelling by Dr. David Mossop