scientific reports

Check for updates

OPEN

# Weighted *p*-norm distance t kernel SVM classification algorithm based on improved polarization

Wenbo Liu[1,2]✉, Shengnan Liang[1,2] & Xiwen Qin[3]

**The kernel function in SVM enables linear segmentation in a feature space for a large number of linear inseparable data. The kernel function that is selected directly affects the classification performance of SVM. To improve the applicability and classification prediction effect of SVM in different areas, in this paper, we propose a weighted *p*-norm distance t kernel SVM classification algorithm based on improved polarization. A t-class kernel function is constructed according to the t distribution probability density function, and its theoretical proof is presented. To find a suitable mapping space, the t-class kernel function is extended to the *p*-norm distance kernel. The training samples are obtained by stratified sampling, and the affinity matrix is redefined. The improved local kernel polarization is established to obtain the optimal kernel weights and kernel parameters so that different kernel functions are weighted combinations. The cumulative optimal performance rate is constructed to evaluate the overall classification performance of different kernel SVM algorithms, and the significant effects of different *p*-norms on the classification performance of SVM are verified by 10 times fivefold cross-validation statistical comparison tests. In most cases, the results using 6 real datasets show that compared with the traditional kernel function, the proposed weighted *p*-norm distance t kernel can improve the classification prediction performance of SVM.**

In the 1990s, Vapnik systematically introduced statistical learning theory and proposed the SVM algorithm[1]. Due to its excellent performance in the field of text mining[2] and fault diagnosis[3], SVM gradually became the mainstream technology of machine learning methods and directly promoted the climax of statistical learning development. The study of the kernel method was officially initiated based on the great success of SVM, and SVM promoted the rapid popularization and application of the kernel method. The kernel method has gradually expanded into many fields of machine learning, such as pattern recognition[4], feature selection[5], and deep learning[6,7]. The kernel function directly determines the performance of the SVM classification algorithm[8] and various kernel methods because a proper kernel function can map samples to an appropriate feature space. In an appropriate feature space, similar samples are close together and different samples are far apart. A kernel function is introduced to greatly improve the accuracy, recognition rate, and dimension reduction efficiency of machine learning algorithms.

Subsequently, many methods based on the kernel technique have been proposed. Schkopf[9] et al. proposed a kernel trick so that principal component analysis could be utilized as a nonlinear dimension reduction technique. As a result, nonlinear mapping from high-dimensional space to low-dimensional space can be achieved and the performance of the learner is improved. Mika[10] introduced the kernel function into linear discriminant analysis (LDA), which is also known as KLDA. KLDA can address the nonlinear data analysis problem and can achieve higher accuracy than LDA. Si proposed a new and improved kernel partial least squares method to address nonlinear characteristics in industrial processes[11]. Some kernel functions have been proposed for specific fields. For example, Huma[12] et al. proposed the application of a string kernel in natural language processing to improve the efficiency of text classification. Bernhard[13] et al. studied the application of kernel methods in the field of bioinformatics.

The above methods are only based on a single kernel. Because different kernel functions have different characteristics, the performance of kernel functions varies greatly in different application scenarios. When the sample size is large, the multidimensional data are irregular or the data are not evenly distributed in the feature space. Therefore, it is not reasonable to map the training set directly by a single kernel[14,15]. To improve the flexibility and

[1]School of Mathematics and Statistics, Qiannan Normal University for Nationalities, Duyun 558000, Guizhou, China. [2]Key Laboratory of Complex Systems and Intelligent Optimization of Qiannan, Duyun 558000, Guizhou, China. [3]School of Mathematics and Statistics, Changchun University of Technology, Changchun 130012, Jilin, China. ✉email: 874717829@qq.com

applicability of the kernel function, multiple kernel functions are combined, i.e., multiple kernel learning. Multiple kernel learning has been a long-standing, well-known and practical research direction in machine learning. Gone[16] provided a taxonomy and review of several multiple kernel learning (MKL) algorithms. They concluded that multiple kernel learning is useful in practice and that a better MKL algorithm could be devised for improved accuracy and decreased complexity and training time. In recent years, many multiple kernel methods have been proposed to solve specific problems. Rakotomamonjy[17] proposed a simple MKL algorithm. In the weighted 2-norm regularization form, an additional 1-norm constraint is applied to the multikernel weight coefficients, which provides a new idea for multiple kernel learning based on mixed norm regularization. Fan[18] proposed a multiple random empirical kernel learning machine (MREKLM), which adopts the random projection idea to map samples into multiple low-dimensional empirical feature spaces with lower computational complexity. Li[19] proposed the multiple kernel learning support vector machine particle swarm optimization model to identify pulmonary nodules and obtained better recognition efficiency. Gao[20] proposed a multiple kernel learning method with the Mahalanobis distance to classify hyperspectral images. Based on the linear weighted combination of the Mahalanobis basic kernel, the hyperspectral data are mapped to a feature space with a smaller intraclass distance and larger interclass distance, and then they are classified to improve the prediction accuracy. Wang[21] proposed a new model parameter selection method for support vector machines based on adaptive fusion of multiple kernel functions and realized adaptive selection of the multiple kernel function weighted coefficient, kernel parameters and regression parameters. Ergul[22] proposed a multiple composite kernel extreme learning machine for hyperspectral images, and the obtained results were presented comparatively along with state-of-the-art standard machine learning.

The multiple kernel model has better applicability and flexibility than the single kernel model. The above works have proven that the interpretability of the decision function can be enhanced and the performance of the learner can be boosted by using multiple kernels instead of a single kernel. In the multiple kernel framework, the convex combination of several single kernels,$\sum_{i=1}^{M} \omega_i \kappa_i$, $\sum_{i=1}^{M} \omega_i = 1$ is the most common form. The key to multiple kernel learning is the selection of a basic kernel and the calculation of weight coefficients. We can use the existing kernel as the basic kernel or create a new kernel according to kernel construction theory to use as the basic kernel[23]. There are two main ways to calculate the weight coefficients: heuristic algorithms[24] and optimization models. The former needs to be associated with the performance of subsequent classifiers, so it is too time-consuming, while the latter has strict theory and lower computational complexity. Examples of typical optimization models are described as follows. Lanckriet[25] obtained the weighted kernel matrix from data based on a semidefinite programming idea and solved the optimal weight coefficient. Sonnenburg[26] rewrote the convex quadratic constrained quadratic programming in reference[25] into a semi-infinite linear programming problem to solve the kernel weight. The gradient descent method was always adopted to optimize the weight by some researchers[27,28].

Obviously, the multiple kernel model consists of several basic single kernels. The expression of the single kernel function often determines the multiple kernel performance. Single kernel functions have the advantage of simple expression and fewer parameters over multiple kernel functions and can solve specific domain problems. Their deficiency lies in the fixed expression form, which results in poor universality. To solve this problem, a more flexible multiscale kernel was introduced[29,30]. In addition, according to distance metric learning theory, samples are mapped from the original space to the feature space so that the performance obtained in the feature space is better than that in the original space[31]. Obtaining a suitable space is essentially determining the proper distance metric. Therefore, the t class kernel function with multiscale form is constructed. To obtain a suitable distance metric, the t kernel is generalized to the $p$-norm t kernel.

In this study, a weighted $p$-norm distance t kernel (W$p$NDtK) SVM classification algorithm based on improved polarization is proposed for solving basic kernel construction and weight coefficient computation in a multiple kernel model. The main contribution of this paper is as follows. We construct a t-class kernel and provide a theoretical proof. To map the sample to a more suitable feature space, we generalize the t-class kernel as a weighted $p$-norm t-class kernel and give its properties. We define the affinity matrix and build an objective function of weight coefficients and kernel parameters according to local kernel polarization. The objective function is solved by the local gradient and the generalized Lagrange multiplier algorithm. The cumulative optimal performance rate is constructed to measure the overall classification performance of SVM algorithms with different kernels. The significance of the $p$-norm distance on SVM classification performance is verified based on the paired data t test with 10 times fivefold cross-validation. Through a large number of experiments on 6 real datasets, the results show that SVM classification prediction can appropriately improve performance when using W$p$NDtK compared with the traditional kernel function.

This paper is organized as follows. In "Introduction" section, we introduce the development and application of the kernel method and the optimal solution of weight coefficients in multiple kernel learning. The basic SVM model with multiple kernels is introduced in "Kernel support vector machine" section. In "t Class kernel and its generalization: section, we describe the construction of a weighted $p$-norm distance t kernel and provide a theoretical proof. In "Establishment and solution of the multiple kernel model" section, we describe the construction of the optimal model of weight coefficients and kernel parameters. The flow of the weighted $p$-norm t kernel SVM classification algorithm is shown in Weighted $p$-norm t kernel SVM classification algorithm" section. Our experimental studies and an evaluation of the performance of the proposed W$p$NDtK SVM algorithm are presented in "Experimental results and analysis" section. The paper is concluded in "Conclusions" section and suggestions for future work are provided.

## Kernel support vector machine

A support vector machine is a classification algorithm for binary classification problems and is based on the theory of structural risk minimization. Of course, SVM can also be extended to multiclass classification learning problems. The basic SVM model is a maximum interval linear classifier defined in the feature space. By introducing the kernel function, SVM essentially becomes a nonlinear classifier. The basic principle of kernel SVM is given as follows.

Given the training dataset $T = \{(\mathbf{x}_i, y_i) | x_i \in R^d, y_i \in \{+1, -1\}, i = 1, 2, ..., n\}$, where $\mathbf{x}_i$ is the $d$ dimensional input vector and $y_i$ is its class label. SVM can be formalized into the following convex quadratic programming problem.

$$\begin{aligned} &\min_{\omega,b,\xi} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t.\, y_i\left[\omega^T\phi(\mathbf{x}_i) + b\right] \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \ldots, n, \end{aligned} \tag{1}$$

where $\omega$ indicates the normal vector of the classification hyperplane, $C$ is a predefined positive trade-off parameter between model simplicity and classification error, $\xi_i$ is the vector of slack variables, $\phi(x)$ is the feature vector mapped from $x$, and $b$ is the bias term of the separating hyperplane. The goal of SVM is to maximize the interval $2/\|\omega\|$.

The dual formulation of Model (1) is generally used when solving SVM

$$\begin{aligned} &\max_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i \\ &= \max_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i \\ &s.t. \sum_{i=1}^{n}\alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, ..., n \end{aligned} \tag{2}$$

where $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$ is the kernel function and $\alpha_i$ is the Lagrangian multiplier. The bias term $b$ can be solved by the support vector in the training dataset. Its specific form is as follows:

$$b = \frac{1}{n_s}\left(\sum_{s=1}^{n_s} y_s - \sum_{i=1}^{n}\alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}_s)\right) \tag{3}$$

where $\mathbf{x}_s$ is the support vector and $n_s$ is the number of support vectors.

The final SVM classifier is

$$\begin{aligned} f(\mathbf{x}) &= \omega^T\phi(\mathbf{x}) + b \\ &= \sum_{i=1}^{n}\alpha_i y_i \kappa(\mathbf{x}, \mathbf{x}_i) + b. \end{aligned} \tag{4}$$

For kernel SVM, the selection of the kernel function is the key to the classification performance of SVM. If the kernel function is not properly selected, the sample is mapped to an inappropriate space, which leads to a poor classification effect. To improve the performance, it is necessary to constantly explore the new kernel functions. Since different kernels are applicable to different areas, the most straightforward idea is to combine several different kernels to integrate the advantages of different kernels.

The simplest and most common way to construct a multiple kernel model is to directly combine some single kernels into convex combinations, and the basic form of this concept is as follows.

$$\begin{aligned} \kappa(x, y) &= \omega_1\kappa_1(x, y) + \omega_2\kappa_2(x, y) + ... + \omega_M\kappa_M(x, y) \\ &= \sum_{i=1}^{M}\omega_i\kappa_i(x, y) \end{aligned} \tag{5}$$

where $\kappa_i(x, y)$ is the basic kernel function, $\omega_i$ is the kernel weight and $\sum_{i=1}^{M}\omega_i = 1$. We can combine existing kernels or construct new classes of kernels. For the determination of kernel weight, a heuristic algorithm or optimization model can be used to solve the weight. The optimization model is used in this work to solve $\omega_i$. Section "The optimization of kernel weight and kernel parameter" provides more details.

According to Model (2), the dual formulation of SVM with multiple kernels is as follows.

$$\max_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\sum_{k=1}^{M}\omega_k\kappa_k(x,y)\right)-\sum_{i=1}^{n}\alpha_i$$
$$s.t.\ \sum_{i=1}^{n}\alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1,2,...,n. \tag{6}$$

## t Class kernel and its generalization

In many practical tasks, samples are often linearly indivisible. Therefore, it is necessary to select the appropriate kernel function to map the samples to an appropriate feature space so that the samples are linearly separable in the feature space. If the kernel function is not properly selected, the sample cannot be linearly segmented in the feature space, resulting in poor SVM classification performance. Therefore, kernel functions directly determine the performance of SVM classification. This encourages us to construct new types of kernel functions to adapt to different fields. Inspired by the t distribution probability density function, a t class kernel function is constructed. For this kernel to have better flexibility and applicability, it is extended to the $p$-norm distance t kernel, and a reasonable distance measurement can be obtained by adjusting the norm.

***p*-norm t kernel.** **Theorem 1** [32] *Suppose that $f : X \to R$ is a bounded continuous integrable function. Then, $k(x - x') = f(x - x')$ is a kernel function if and only if $f(0) > 0$ and its Fourier transform.*

$$\tilde{f}(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-i\omega x}dx \ge 0. \tag{7}$$

**Theorem 2** *When $n \to +\infty$, the t distribution probability density function.*

$$f(t) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1+\frac{t^2}{n})^{-\frac{n+1}{2}} \tag{8}$$

*is the kernel function, where $\Gamma(\cdot)$ is the gamma function.*

***Proof*** Let $|x| = t^2, x \in (-\infty, \infty)$, Eq. (5) is transformed into.

$$f(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1+\frac{|x|}{n})^{-\frac{n+1}{2}}$$

and $f(0) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})} > 0$.

$$\tilde{f}(\omega) = \lim_{n\to+\infty}\int_X f(x)e^{-i\omega x}dx$$

$$= \int_X \lim_{n\to+\infty} f(x)e^{-i\omega x}dx$$
$$= \int_{-\infty}^{+\infty}\lim_{n\to+\infty}\frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1+\frac{|x|}{n})^{-\frac{n+1}{2}}e^{-i\omega x}dx$$
$$= \int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{|x|}{2}}e^{-i\omega x}dx$$

where $e^{-\frac{|x|}{2}}$ is the Laplacian kernel function. According to Theorem 1,

$$\int_{-\infty}^{+\infty} e^{-\frac{|x|}{2}}e^{-i\omega x}dx \ge 0,$$

Therefore,

$$\tilde{f}(\omega) = \int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{|x|}{2}}e^{-i\omega x}dx \ge 0$$

When $n \to \infty$, the function

$$f(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1+\frac{|x|}{n})^{-\frac{n+1}{2}} \tag{9}$$

is the kernel function.

Theorem 2 shows that when the sample size is sufficiently large, the probability density function of the t distribution can be used as the kernel function. The number of $n1$ that should be taken is often determined by experimental analysis. For the convenience of kernel function application, Corollary 1 is given as follows.

**Corollary 1** *When* $n$=1, Eq. (6) *is equivalent to*

$$f(x) = \frac{1}{\pi(1+|x|)}. \tag{10}$$

Then, Eq. (10) is the kernel function.

By generalizing the kernel function in Corollary 1, Corollary 2 is obtained as follows.

**Corollary 2** *Let*

$$f(x) = c\left(\frac{1}{1+|x|}\right)^{v} \tag{11}$$

*where* $c > 0, 0 < v \leq 1$*; then,* Eq. (11) *is the kernel function.*

***Proof*** When, $0 < \frac{1}{1+|x|} \leq 1, c > 0, 0 < v \leq 1$, we have

$$c\left(\frac{1}{1+|x|}\right) \leq c\left(\frac{1}{1+|x|}\right)^{v}$$
$$0 \leq \int_{-\infty}^{+\infty} \frac{c}{1+|x|} e^{-i\omega x} dx \leq \int_{-\infty}^{+\infty} c\left(\frac{1}{1+|x|}\right)^{v} e^{-i\omega x} dx$$

Therefore, $f(x) = c\left(\frac{1}{1+|x|}\right)^{v}$ is the kernel function. □

The kernel parameter in Corollary 2 ranges from 0 to 1. We can consider expanding the range of $v$ to increase the applicability of the kernel function.

**Theorem 3** [33] *$X \subset R^n$,$f : (0,\infty) \to R$,$\kappa$ is the function defined on $X \times X$ and $\kappa(\mathbf{x},\mathbf{z}) = f(\|\mathbf{x}-\mathbf{z}\|^2)$. When $f$ is completely monotone, $\kappa(\mathbf{x},\mathbf{z})$ is a positive definite kernel.*

**Corollary 3** *When* $c > 0, v > 0$,

$$f(x) = c\left(\frac{1}{1+x}\right)^{v} \tag{12}$$

*is the kernel function, where* $x > 0$.

***Proof***

$$f^{(n)}(x) = (-1)^n cv(v+1)...(v+(n-1))(1+x)^{-(v+n)}$$
$$(-1)^n f^{(n)}(x) = (-1)^{2n} cv(v+1)...(v+(n-1))(1+x)^{-(v+n)}$$
$$= cv(v+1)...(v+(n-1))(1+x)^{-(v+n)}$$

When $c > 0, v > 0$ and $(-1)^n f^{(n)}(x) \geq 0$, $f(x)$ is completely monotone. According to Theorem 3, $f(x)$ is the kernel function. □

According to the complete monotonicity of the function, Corollary 3 expands the range of kernel parameters on the basis of Corollary 2, which provides more choices for us to use the kernel function.

In practical applications, Eq. (12) is in the following form:

$$\kappa(\mathbf{x}_i,\mathbf{x}_j) = \left(\frac{c}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2}\right)^{v} \tag{13}$$

where the number 2 indicates the 2-norm. To find an appropriate distance measure in the mapped feature space, the Euclidean distance in Eq. (10) is generalized to the $p$-norm distance, and we can obtain

$$\kappa(\mathbf{x}_i,\mathbf{x}_j) = \left(\frac{c}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_p}\right)^{v} \tag{14}$$

where $p$ is the $p$-norm. Equation (14) is called the $p$-norm distance t class kernel for the short $p$-norm t kernel.

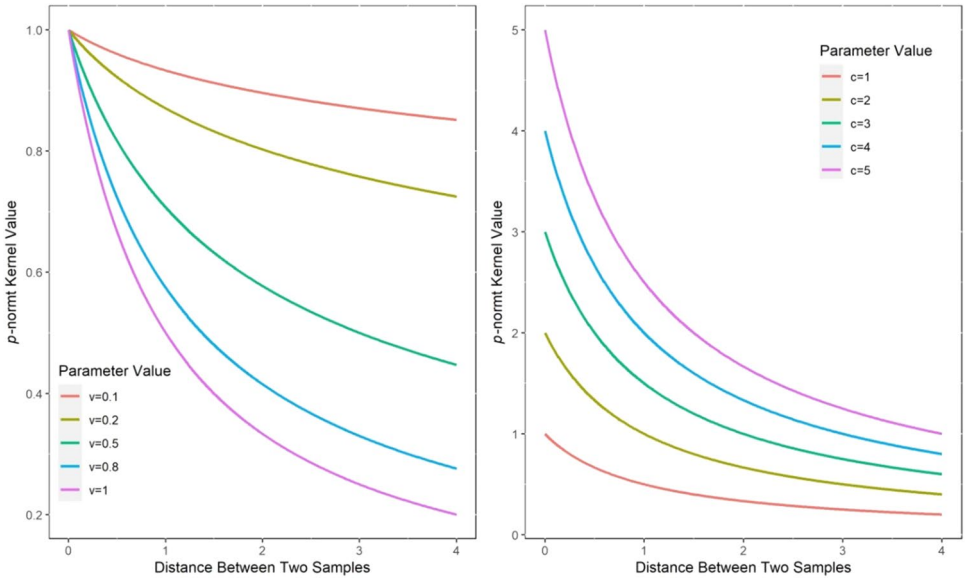


**Figure 1.** $p$-norm distance t-kernel value under different scale parameters.

**The properties of the kernel function.** Since the kernel function constructed in "$p$-norm t kernel" section is eventually extended to the form of Eq. (14), the corresponding properties are given in this section. We also discuss whether this kernel function is reasonable.

***Property 1*** When $c > 0, v > 0$, $f(x) = c\left(\frac{1}{1+x}\right)^v$ is a decreasing function of $x$, and.

$d_{ij} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_p$ is the $p$-norm distance of any two samples. According to Property 1, the closer the sample is, the larger the kernel value is, and vice versa. When $x = 0$, the kernel function is at its maximum value. This shows that the kernel function can describe the similarity between samples well. The larger the kernel value is, the higher the similarity between samples.

***Property 2*** The function $f(x) = c\left(\frac{1}{1+x}\right)^v$ has multiscale characteristics, where $c$ and $v$ are the scale parameters.

Property 2 is illustrated by function graphs, which are drawn by fixing $c = 1$ and $v = 1$, as shown in Fig. 1.

When the scale parameters $c$ and $v$ are small, the kernel function can adapt to the samples with drastic changes, and when the scale parameters are large, the kernel function can adapt to the samples with gentle changes[34] so that it has better adaptability in processing complex data. Similar to the Gaussian kernel function, the constructed kernel function in Eq. (14) is also a typical multiscale kernel.

## Establishment and solution of the multiple kernel model

**Weighted kernel function.** Because different kernel functions have different characteristics, their performance will be significantly different for different types of datasets. To make the kernel function more flexible in application, the multiple kernel learning model is formed by kernel combination. Using multiple kernels instead of a single kernel can enhance the interpretability of the decision function and result in better performance than a single kernel[35].

When the $p$-norm t kernel constructed in "$p$-norm t kernel" section is combined, we can obtain the combination kernel as follows.

$$\sum_{s=1}^{M} \omega_s \left(\frac{c}{1 + \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_p}\right)^{v_s} \tag{15}$$

Under the framework of a multiple kernel learning model, the representation of original samples in feature space is transformed into basic kernel selection and the calculation of weight coefficients. Each basic kernel corresponds to a basic feature space and how to fuse these basic feature spaces to obtain a suitable combined feature space. That is, the data can be better represented in the combined feature space to improve the classification prediction performance. Obtaining the combined feature space is essentially a problem of optimal calculation of weight coefficients.

Currently, there are two main methods to calculate the weight coefficient: a heuristic algorithm and an optimization algorithm. In this work, an optimization method that has a more rigorous theory is adopted to solve the weight coefficients. The key step to establish the optimization model is to give the objective function. In this study, the objective function is established based on kernel target alignment, and the optimal solution should maximize the target value. Kernel target alignment only relies on training samples and is unrelated to subsequent classifiers, so the implementation of this strategy is simple and has attracted a large amount of attention. Since the kernel function contains hyperparameters, the value of the kernel parameters also has a significant impact on the performance of the classification prediction results. Therefore, how to select the appropriate hyperparameters is also a key consideration. A direct approach is to put the kernel parameters and the weight together into the objective function for optimization.

**Kernel target alignment.** Kernel target alignment is a parameter optimization criterion established based on matrix alignment. This type of method only relies on training samples and is unrelated to the learning performance of subsequent classifiers. Therefore, the algorithm is simple and quick to implement, and its basic principle is as follows.

Given the training dataset $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and class label $\mathbf{y} = \{y_1, y_2, ..., y_n\}$, $y_i \in \{1, 2, ..., k\}$ shows that the dataset has $k$ classes, and $\mathrm{K} = (\kappa(\mathbf{x}_i, \mathbf{x}_j))_{n\times n}$ is the kernel matrix. Then, $Y = \mathbf{y}\mathbf{y}^T = (y_{ij})_{n\times n}$ is the class label matrix and is also called the ideal kernel matrix, where.

$$y_{ij} = \begin{cases} 1, & y_i = y_j \\ -1, y_i \neq y_j \end{cases}$$

.

The goal of the kernel target alignment is to maximize the cosine value between the kernel matrix and the ideal kernel matrix, and its expression is as follows.

$$A(\mathrm{K}, Y) = \frac{< \mathrm{K}, Y >_F}{\|\mathrm{K}\|_F \|Y\|_F} \tag{16}$$

where $< \cdot, \cdot >_F$ is the Frobenius inner product and $\| \cdot \|_F$ is the Frobenius norm. Reference[36] proves the reliability and practicability of the kernel target alignment and the boundedness of the generalization error of the kernel classifier. On the basis of Eq. (15), Baram proposed kernel polarization inspired by physics[37]. It is defined as the Frobenius inner product.

$$P(\mathrm{K}) = \sum_{i=1}^{n}\sum_{j=1}^{n} y_{ij}\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j). \tag{17}$$

where $P(\mathrm{K})$ only takes between-class separability into account but neglects the preservation of within-class local structures; therefore, Wang proposed local kernel polarization (LKP)[38], which is defined as.

$$L(\mathrm{K}) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}y_{ij}\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j). \tag{18}$$

The affinity coefficient is defined as

$$A_{ij} = \begin{cases} \exp(-t\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2), y_i = y_j \\ 1, \qquad\qquad\quad y_i \neq y_j \end{cases} \tag{19}$$

where $t > 0$ is the adjusting parameter. From Eq. (19), the affinity coefficient $A_{ij}$ is defined by the Gaussian kernel function. Certainly, there should be some other more appropriate manners of defining the affinity coefficient. Therefore, we redefine the affinity coefficient in "The optimization of kernel weight and kernel parameter" section to obtain better results.

**The optimization of kernel weight and kernel parameter.** Based on the basic idea of the LKP, an improved local kernel polarization model is constructed to obtain the optimal kernel weights and kernel parameters. The improved part is reflected in the redefinition of the affinity coefficient in the LKP. The specific optimization model is as follows.

$$\begin{aligned} &\max_{\omega, v_s} \sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}y_{ij}\sum_{s=1}^{M}\omega_s\left(\frac{c}{1+\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_{p_s}}\right)^{v_s} \\ &s.t.\ 0 \leq \omega_s \leq 1,\ v_s > 0, \sum_{s=1}^{M}\omega_s = 1. \end{aligned} \tag{20}$$

By redefining the affinity coefficient, we obtain.

$$A_{ij} = \begin{cases} \dfrac{1}{1 + \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_p}, y_i = y_j \\ 1, \qquad\qquad , y_i \neq y_j \end{cases} \tag{21}$$

For Model (20), an optimization algorithm combining the local gradient and generalized Lagrange multiplier is adopted[39]. The gradient form of the model is as follows:

$$\begin{aligned} &\frac{\partial \sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}y_iy_j \sum_{s=1}^{M} \omega_s \left(\frac{c}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_{p_s}}\right)^{v_s}}{\partial \omega_s} \\ &= \sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}y_iy_j \left(\frac{c}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_{p_s}}\right)^{v_s}, \\ &= \frac{\partial \sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}y_iy_j \sum_{s=1}^{M} \omega_s \left(\frac{c}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_{p_s}}\right)^{v_s}}{\partial v_s} \\ &= \sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}y_iy_j\omega_s \left(\frac{c}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_{p_s}}\right)^{v_s} \log\frac{1}{1+\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_{p_s}}, s = 1, 2, ..., M \end{aligned}$$

To facilitate calculation, the parameters in Eq. (20) can be specified in advance, and for convenience $c = 1$.

Equation (20) only contains the weighted $p$-norm t kernel. However, according to different field applications, the $p$-norm t kernel can also be combined with other types of kernel functions to obtain better classification performance.

## Weighted $p$-norm t kernel SVM classification algorithm

According to the construction principle of the $p$-norm t-kernel and the establishment and solving process of the multiple kernel model, the basic flow of the weighted $p$-norm t kernel SVM classification algorithm is as follows.

Input: $Train = \{(\mathbf{x}_i, y_i)\big|\mathbf{x}_i \in R^p, y_i \in Y, i = 1, 2, ..., n\}$, where $Y = \{1, 2, ..., l\}$ is the class label.

Output: The predicted class $\hat{y}_i$ of $Test = \{\mathbf{x'}_i\big|\mathbf{x}' \in R^p\}, i = 1, 2, ..., n'$.

Step 1: The dataset is divided into a training set and a test set by k-fold cross stratified sampling.

Step 2: A specific kernel function is selected according to Eq. (5).

Step 3: The affinity coefficient matrix is built according to Eq. (21).

Step 4: According to Eq. (18), the objective function of kernel weight and kernel parameter is established.

Step 5: Based on the training set, the local gradient and generalized Lagrange multiplier[39] are used to solve Model (20) and obtain the optimal weight coefficients $\omega_i$ and kernel parameters $v, \gamma, d$.

Step 6: The optimal parameters obtained in Step 5 are substituted into Eq. (5).

Step 7: Eq. (5), which is obtained in Step 6, is substituted into Model (6) to obtain the specific dual formulation of the multiple kernel SVM.

Step 8: The training set $Train$ obtained by stratified sampling is used to fit Model (6).

Step 9: The test set is put into the fitted Model (6) to obtain the predicted class label $\hat{y}_i$.

In Step 1, stratified sampling is used to prevent class imbalance in the training set and prevent the fitted SVM classification model from having class tendency. The specific form of each single kernel function must be specified in Step 2. In this study, the $p$-norm t-kernel constructed in "$p$-norm t kernel" section is mainly used for weighted combination. According to the experimental analysis in Step 6, to make use of the unique advantages of different kernel functions, the $p$-norm t kernel can also be combined with traditional kernel functions, including the Gaussian kernel and polynomial kernel. Steps 3 to 5 belong to the optimization process of model parameters, including the solution of weight coefficients and kernel parameters. The objective function is established according to the local kernel polarization, and the local gradient and the generalized Lagrange multiplier are used to solve it. Of course, other optimization algorithms can also be adopted. For details, please refer to reference[40]. Steps 6 to 8 fit the multiple kernel SVM model, and Step 9 predicts the test samples based on the fitted model. Finally, a specific evaluation index is used to evaluate the weighted $p$-norm t kernel SVM classification algorithm.

## Experimental results and analysis

**Experimental setting.** The experimental environment uses a Windows 10 64-bit operating system with an Intel i7-9700 @ 3.0 GHz CUP and 16 GB memory. The algorithm and experiment proposed in this paper are implemented based on R language (R 3.6.3) coding. Experimental data are from the Broad Institute Genome Data Analysis Center and UCI machine learning library. The specific information is shown in Table 1..

We compare the performance of the W$p$Nt + SVM algorithm with the following methods:

(i) Poly + SVM: The polynomial kernel is used in SVM.
(ii) Sig + SVM: The sigmoid kernel is used in SVM.
(iii) Gau + SVM: The Gaussian kernel is used in SVM.
(iv) Lap + SVM: The Laplace kernel is used in SVM.

| Dataset name | Sample size | Feature | Categories | Data source |
|---|---|---|---|---|
| Kidney | 400 | 24 | 2 | UCI |
| Dermatology | 366 | 34 | 6 | UCI |
| Sonar | 208 | 60 | 2 | UCI |
| Pima | 768 | 8 | 2 | UCI |
| Postcode | 7291 | 256 | 10 | [41] |
| Breast | 98 | 1213 | 3 | BIGDAC yyy[42,43] |

**Table 1.** Data information. UCI: http://archive.ics.uci.edu/ml/index.php. BIGDAC: http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi.

(v) Simple MKL: The linear combination of kernel approach is used. Two kernel functions are mixed in the experiment, including two Gaussian kernels, one Gaussian kernel and one linear kernel.

To compare the effects of different kernel functions on the performance of the SVM classification algorithm, the experiment used fivefold cross-validation to divide the training set and test set, and the evaluation criteria were classification accuracy, recall, Kappa coefficient[44] and training time. The training time of the algorithm is related to the range of parameter settings, as it often takes more time to obtain results with good performance. Different from the previous three evaluation indices, the training time of the algorithm is discussed separately in "Comparison experiment" section. Due to the large sample size of the postcode dataset, 10% random sampling is carried out in the training phase to reduce the time. Because of the high dimensionality of the breast dataset, PCA is used to reduce its dimensionality in advance. To evaluate the overall performance level of the W$p$Nt + SVM algorithm, the optimal performance rate is constructed as follows.

$$OPR = \frac{PN}{MN \times DN \times EN} \tag{22}$$

where $MN$ is the number of algorithms, $DN$ is the number of datasets, $EN$ is the number of evaluation indices, and $PN$ is the number of W$p$Nt + SVM that reaches the maximum under each evaluation index.

Equation (22) is generalized to obtain the cumulative optimal performance rate (COPR). Its definition is as follows:

$$COPR = \frac{\sum_{i=1}^{m} PN_i}{MN \times DN \times EN} \tag{23}$$

where $PN_i$ is the number of algorithms reaching the $i^{\text{th}}$ maximum under each evaluation index, and $m$ is the number of methods.

**Comparison experiment.** For different datasets, SVM classification based on different kernel functions yields different prediction effects. In experimental analysis, to obtain better classification and prediction performance, flexibility is required when encountering different datasets; that is, multiple $p$-norm distance t kernels should be combined or $p$-norm distance t kernels should be combined with traditional kernel functions when encountering different datasets. To reduce the complexity of the experiment, only two kernel functions are combined, and a positive trade-off parameter $C = 1$ is allowed in all the SVM models. After many comparative experiments, different weighted kernel functions are selected for different datasets. The form of weighted kernel functions is mainly as follows.

$$\omega_1 \left( \frac{1}{1 + \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{p_s}} \right)^{v_1} + \omega_2 \left( \frac{1}{1 + \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{p_s}} \right)^{v_2} \tag{24}$$

$$\omega_1 \left( \frac{1}{1 + \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{p_s}} \right)^{v_1} + \omega_2 \left( \mathbf{x}_i \cdot \mathbf{x}_j \right)^d \tag{25}$$

$$\omega_1 \left( \frac{1}{1 + \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{p_s}} \right)^{v_1} + \omega_2 \exp\left( -\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2 \right) \tag{26}$$

Equation (24) is applied to the Kidney and Pima datasets, Eq. (25) is applied to the Postcode and Breast datasets, and Eq. (26) is applied to the Dermatology and Sonar datasets. When calculating the kernel weight and the kernel parameters, optimization Model (20) is adopted, and the aforementioned local gradient and generalized Lagrange multiplier method are used to solve the problem. The results are shown in Table 2.

| Dataset | $\omega_1$ | $\omega_2$ | Kernel parameter 1 | Kernel parameter 2 |
|---|---|---|---|---|
| Kidney | 0.78 | 0.22 | $v_1 = 1.00$ | $v_2 = 0.80$ |
| Dermatology | 0.23 | 0.77 | $v_1 = 0.94$ | $\gamma = 0.01$ |
| Sonar | 0.91 | 0.29 | $v_1 = 0.94$ | $\gamma = 0.04$ |
| Pima | 0.78 | 0.22 | $v_1 = 0.91$ | $v_2 = 0.86$ |
| Postcode | 0.86 | 0.14 | $v_1 = 0.84$ | $d = 1.00$ |
| Breast | 0.65 | 0.35 | $v_1 = 0.99$ | $d = 1.00$ |

**Table 2.** The optimized result of the weight coefficients and kernel parameters.

| Dataset | Poly + SVM | Sig + SVM | Gau + SVM | Lap + SVM | SMKL + SVM | W*p*Nt + SVM |
|---|---|---|---|---|---|---|
| Kidney | $d = 4$<br>0.9575 | $\beta = 1\ \theta = -8$<br>0.9850 | $\sigma = 0.1$<br>**0.9975** | $\sigma = 0.1$<br>**0.9975** | $\sigma_1 = 0.01\ \sigma_2 = 0.05$<br>**0.9975** | $p = 1.5$<br>**0.9975** |
| Dermatology | $d = 3$<br>0.9344 | $\beta = 0.1\ \theta = -2$<br>0.9672 | $\sigma = 0.05$<br>0.9645 | $\sigma = 0.1$<br>0.9699 | $\sigma_1 = 0.01\ \sigma_2 = 0.04$<br>0.9672 | $p = 1.5$<br>**0.9726** |
| Sonar | $d = 3$<br>**0.8559** | $\beta = 0.01\ \theta = -1$<br>0.7978 | $\sigma = 0.01$<br>0.8413 | $\sigma = 0.05$<br>0.8170 | $\sigma_1 = 0.01\ \sigma_2 = 0.02$<br>0.8364 | $p = 2$<br>0.8459 |
| Pima | $d = 2$<br>0.7448 | $\beta = 1\ \theta = -1$<br>0.6771 | $\sigma = 0.1$<br>0.7643 | $\sigma = 0.05$<br>0.7735 | $\sigma_1 = 0.01\ \sigma_2 = 0.02$<br>**0.7748** | $p = 2$<br>0.7696 |
| Postcode | $d = 2$<br>0.9243 | $\beta = 0.01\ \theta = -2$<br>**0.9257** | $\sigma = 0.01$<br>0.8432 | $\sigma = 0.05$<br>0.9230 | $\sigma = 0.01, d = 1$<br>0.9243 | $p = 1.5$<br>**0.9257** |
| Breast | $d = 2$<br>0.8684 | $\beta = 1\ \theta = -1$<br>0.7653 | $\sigma = 0.01$<br>0.8384 | $\sigma = 0.05$<br>0.6947 | $\sigma = 0.02, d = 1$<br>0.8684 | $p = 2.5$<br>**0.8789** |

**Table 3.** The fivefold cross-validation classification accuracy based on the SVM algorithm with different kernel functions. Significant values are in bold.

| Dataset | Poly + SVM | Sig + SVM | Gau + SVM | Lap + SVM | SMKL + SVM | W*p*Nt + SVM |
|---|---|---|---|---|---|---|
| Kidney | $d = 4$<br>0.9494 | $\beta = 1\ \theta = -8$<br>0.9875 | $\sigma = 0.1$<br>**0.9979** | $\sigma = 0.1$<br>**0.9979** | $\sigma_1 = 0.01\ \sigma_2 = 0.05$<br>**0.9979** | $p = 1.5$<br>**0.9979** |
| Dermatology | $d = 3$<br>0.9453 | $\beta = 0.1\ \theta = -2$<br>0.9727 | $\sigma = 0.05$<br>0.9704 | $\sigma = 0.1$<br>0.9749 | $\sigma_1 = 0.01\ \sigma_2 = 0.04$<br>0.9727 | $p = 1.5$<br>**0.9772** |
| Sonar | $d = 3$<br>**0.8720** | $\beta = 1\ \theta = -1$<br>0.8077 | $\sigma = 0.01$<br>0.8678 | $\sigma = 0.05$<br>0.8237 | $\sigma_1 = 0.01\ \sigma_2 = 0.02$<br>0.8398 | $p = 2$<br>0.8636 |
| Pima | $d = 2$<br>0.7491 | $\beta = 1\ \theta = -1$<br>0.6820 | $\sigma = 0.1$<br>0.7761 | $\sigma = 0.05$<br>0.7883 | $\sigma_1 = 0.01\ \sigma_2 = 0.02$<br>**0.7925** | $p = 2$<br>0.7774 |
| Postcode | $d = 2$<br>0.9211 | $\beta = 0.01\ \theta = -2$<br>**0.9381** | $\sigma = 0.1$<br>0.8693 | $\sigma = 0.05$<br>0.9358 | $\sigma = 0.02, d = 1$<br>0.9370 | $p = 1.5$<br>0.9370 |
| Breast | $d = 2$<br>0.8070 | $\beta = 1\ \theta = -1$<br>0.6377 | $\sigma = 0.1$<br>0.7820 | $\sigma = 0.05$<br>0.5789 | $\sigma = 0.02, d = 1$<br>0.8070 | $p = 2.5$<br>**0.8158** |

**Table 4.** The fivefold cross-validation classification recall based on the SVM algorithm with different kernel functions. Significant values are in bold.

The $p$-norm value is set by the wrapping strategy in Eqs. (24)–(26). The $p$-norm value in $[a, b]$ is set and the step size $\lambda$ is given. For different $p$ values, each performance index of $N$ times k-fold cross-validation of the proposed method is calculated, including accuracy, recall and Kappa coefficient. Finally, the $p$-norm value corresponding to the optimal performance index is determined.

For W*p*Nt + SVM algorithm, the objective function with the kernel weight and kernel parameter is established according to the improved local polarization. The local gradient and generalized Lagrange multiplier is adopted to obtain the optimal weights and parameters. For the other comparison algorithms, grid search strategy and k-fold cross validation are used to obtain the optimal parameters.

The different kernel SVM methods are denoted as Poly + SVM, Sig + SVM, Gau + SVM, Lap + SVM, SMKL + SVM and W*p*Nt + SVM. These methods are used to perform fivefold cross-validation classification prediction for the 6 datasets shown in Table 1. The obtained comparative experimental results are shown in Tables 3, 4 and 5, and the optimal results are bolded.

According to the experimental results in Tables 3, 4 and 5, the accuracy of the W*p*Nt + SVM algorithm is optimal for 4 datasets and suboptimal in 1 datasets, the recall of the W*p*Nt + SVM algorithm is optimal for 3 datasets and suboptimal for 1 dataset, and the Kappa coefficient of the W*p*Nt + SVM algorithm is optimal for 3 datasets and suboptimal for 2 datasets. According to Eqs. (18) and (19), the optimal performance rate and cumulative optimal performance rate of W*p*Nt + SVM are calculated as follows.

| Dataset | Poly + SVM | Sig + SVM | Gau + SVM | Lap + SVM | SMKL + SVM | W*p*Nt + SVM |
|---|---|---|---|---|---|---|
| Kidney | $d = 4$<br>0.9113 | $\beta = 1\ \theta = -8$<br>0.9679 | $\sigma = 0.1$<br>**0.9945** | $\sigma = 0.1$<br>**0.9945** | $\sigma_1 = 0.01\,\sigma_2 = 0.05$<br>**0.9945** | $p = 1.5$<br>**0.9945** |
| Dermatology | $d = 3$<br>0.9174 | $\beta = 0.1\ \theta = -2$<br>0.9586 | $\sigma = 0.05$<br>0.9550 | $\sigma = 0.1$<br>0.9619 | $\sigma_1 = 0.01\,\sigma_2 = 0.04$<br>0.9585 | $p = 1.5$<br>**0.9654** |
| Sonar | $d = 3$<br>**0.7055** | $\beta = 1\ \theta = -1$<br>0.5922 | $\sigma = 0.01$<br>0.6778 | $\sigma = 0.05$<br>0.6300 | $\sigma_1 = 0.01\,\sigma_2 = 0.02$<br>0.6692 | $p = 2$<br>0.6861 |
| Pima | $d = 2$<br>0.4154 | $\beta = 1\ \theta = -1$<br>0.2894 | $\sigma = 0.1$<br>0.4573 | $\sigma = 0.05$<br>0.4743 | $\sigma_1 = 0.01\,\sigma_2 = 0.02$<br>**0.4761** | $p = 2$<br>0.4741 |
| Postcode | $d = 2$<br>0.9148 | $\beta = 0.01\ \theta = -2$<br>**0.9163** | $\sigma = 0.1$<br>0.8232 | $\sigma = 0.05$<br>0.9130 | $\sigma = 0.02, d = 1$<br>0.9146 | $p = 2$<br>0.9161 |
| Breast | $d = 2$<br>0.7683 | $\beta = 1\ \theta = -1$<br>0.5935 | $\sigma = 0.1$<br>0.7106 | $\sigma = 0.05$<br>0.3879 | $\sigma = 0.02, d = 1$<br>0.7694 | $p = 2.5$<br>**0.7862** |

**Table 5.** The fivefold cross-validation classification Kappa coefficient based on the SVM algorithm with different kernel functions. Significant values are in bold.

| Dataset | Poly + SVM | Sig + SVM | Gau + SVM | Lap + SVM | SMKL + SVM | W*p*Nt + SVM |
|---|---|---|---|---|---|---|
| Kidney | 0.05 | 0.12 | 0. 87 | 0.05 | 0.64 | 0.78 |
| Dermatology | 0.07 | 0.18 | 2.64 | 0.133 | 0.54 | 0.65 |
| Sonar | 0.04 | 0.18 | 0.60 | 0.03 | 0.21 | 0.29 |
| Pima | 4.20 | 0.29 | 2.01 | 0.08 | 2.53 | 2.71 |
| Postcode | 1.37 | 3.73 | 35.25 | 1.71 | 8.21 | 17.53 |
| Breast | 0.31 | 0.36 | 0.70 | 0.33 | 0.32 | 0.39 |

**Table 6.** The fivefold cross-validation classification training time based on the SVM algorithm with different kernel functions (minutes).

$$OPR = \frac{4+3+3}{1 \times 6 \times 3} \approx 0.5566$$
$$COPR = \frac{(5+3+2) + (1+1+2)}{1 \times 6 \times 3} \approx 0.7778$$

In the 6 datasets analysed, W*p*Nt + SVM is optimal in 10 cases and suboptimal in 6 cases, and the cumulative optimal performance rate is 0.7778, which is close to 80%. This shows that the *p*-norm t kernel constructed for this study can effectively improve the classification and prediction performance of the SVM algorithm. In addition, the combination of the *p*-norm t kernel with the classical Gaussian kernel and polynomial kernel is often better than the single kernel function. Therefore, multiple learning methods can utilize the advantages of each single kernel effectively.

In classification prediction, the training time of the algorithm is also an important evaluation index. Since the final parameters of the comparison algorithm are determined by the wrapping strategy, the grid search strategy is used to set the range of hyperparameters in advance.

The specific setup information is polynomial kernel:$d = 1:5$, and the step size is 1; Gaussian kernel $\sigma = 0.01:4$, and the step size is 0.01; Laplace kernel:$\sigma = 0.01:1$, and the step size is 0.05; Sigmoid kernel: $\beta = 1:5, \theta = -10:-1$, and the step size is 1. The optimization model is used to solve the kernel weights and parameters of the W*p*Nt + SVM algorithm, so there is no need to set parameters in advance. See Table 6 for the specific training time (in minutes) of all algorithms.

According to Table 6, except for the Gau + SVM algorithm, in general, the training time of W*p*Nt + SVM is higher than that of the other comparison algorithms in most cases. It should be emphasized that for Poly + SVM, Sig + SVM, Gau + SVM and Lap + SVM, the training time is dependent on the setting range of the parameters. The optimization model is established to solve the parameters of W*p*Nt + SVM and SMKL + SVM based on the improved local polarization. Therefore, the algorithm proposed in this study does not depend on the setting range of the parameters. The hyperparameter in the Gaussian kernel has the smallest step size compared to other single kernels. The training time of Gau + SVM is significantly higher than that of W*p*Nt + SVM and SMKL + SVM in all datasets except the Pima dataset. This indicates that the training time of Poly + SVM, Sig + SVM, Gau + SVM and Lap + SVM will certainly exceed the training time of W*p*Nt + SVM and SMKL + SVM if the value range of parameters is added and the step size is continuously reduced. When dealing with the large sample data, R or Python's GPU module can be called for training the model. W*p*Nt + SVM can be parallel computing, so that the training time is reduced.

**Statistical measurement comparison test of *p*-norm distance.** For the W*p*Nt + SVM algorithm, different *p*-norm distances are set for different datasets because in the process of experimental analysis, it was

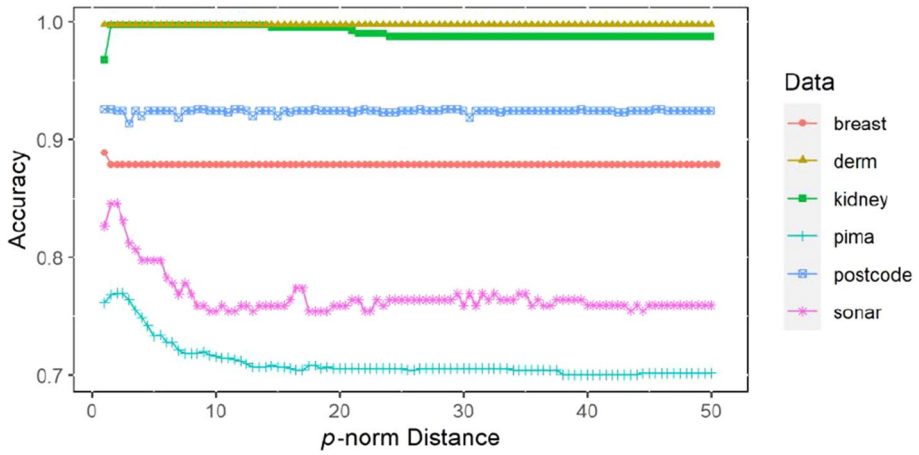


**Figure 2.** The fivefold cross-validation accuracy varies with $p$-norm distance based on 6 datasets.

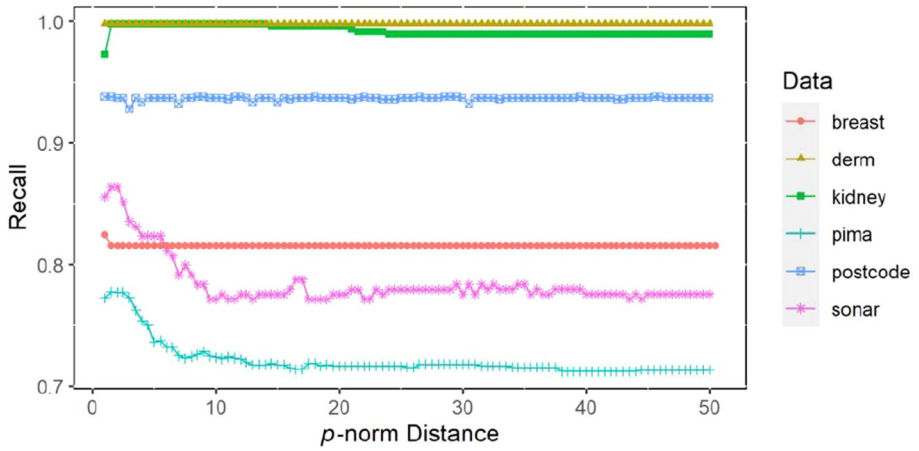


**Figure 3.** The fivefold cross-validation recall varies with $p$-norm distance based on 6 datasets.

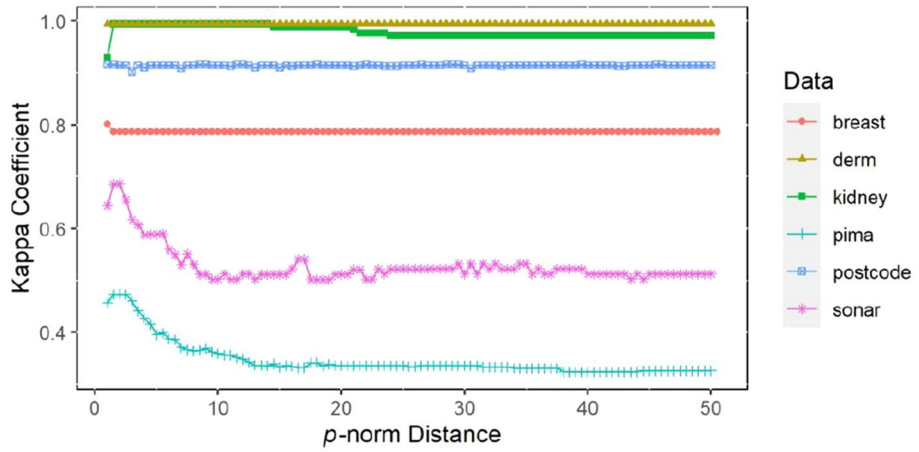


**Figure 4.** The fivefold cross-validation Kappa coefficient varies with $p$-norm distance based on 6 datasets.

found that different norms in the $p$-norm t kernel affect the classification performance of the SVM algorithm. For details, please refer to Figs. 2, 3 and 4, where $p \in [1,\ 50]$ and the step size is 0.5.

It can be clearly seen from Figs. 2, 3 and 4 that the accuracy, recall and Kappa coefficient using the W$p$Nt + SVM algorithm on the Sonar, Pima and Kidney datasets show significant changes with increasing $p$-norm distance. There is a gradual increase in these metrics until the highest point is reached, then an overall downwards

| Dataset | *p*-norm distance |
|---|---|
| Kidney | $p_1 = 1.5\ p_2 = 2$ |
| Dermatology | $p_1 = 1.5\ p_2 = 2$ |
| Sonar | $p_1 = 1.5\ p_2 = 2$ |
| Pima | $p_1 = 3\ p_2 = 2$ |
| Postcode | $p_1 = 1.5\ p_2 = 3$ |
| Breast | $p_1 = 2.5\ p_2 = 3$ |

**Table 7.** The *p*-norm distance setting in different datasets.

| | Accuracy test | | | Recall test | | | Kappa coefficient test | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | *t* Value | $\bar{x} - \bar{y}$ | Sig | *t* Value | $\bar{x} - \bar{y}$ | Sig | *t* Value | $\bar{x} - \bar{y}$ | Sig |
| Kidney | 6.488 | 0.0232 | **Yes** | 7.489 | 0.0279 | **Yes** | 4.421 | 0.0359 | **Yes** |
| Dermatology | − 1.214 | − 0.0009 | No | − 0.9019 | -0.0005 | No | − 1.68 | -0.0030 | No |
| Sonar | 3.074 | 0.0057 | **Yes** | 2.775 | 0.0054 | **Yes** | 2.874 | 0.0104 | **Yes** |
| Pima | − 5.116 | − 0.0060 | **Yes** | − 3.283 | -0.0055 | **Yes** | − 5.254 | − 0.0118 | **Yes** |
| Postcode | 1.937 | 0.0025 | No | 1.157 | 0.0015 | No | 1.934 | 0.028 | No |
| Breast | 1.5 | 0.0020 | No | 1.5 | 0.0017 | No | 1.496 | 0.0035 | No |

**Table 8.** The statistical comparison test of the weighted t kernel SVM classification performance at the 2 level *p*-norm. Sig: significance. Significant values are in bold.

trend with a slight fluctuation in the middle is observed. Finally, the results become stable. For the breast dataset, the accuracy, recall and Kappa coefficient of the model began to decline after reaching the highest point and basically remained at the same level. For the dermatology and postcode datasets, the accuracy, recall and Kappa coefficient of the model fluctuate only slightly with the change in the *p*-norm. From the visualized results in Figs. 2, 3 and 4, it can be concluded that setting different *p*-norm distances has a significant effect on the performance of the classification algorithm in some datasets, while the effect is relatively minimal in other datasets.

The above analysis verifies the influence of different *p*-norm distances on SVM classification performance through a set of cross-validation results, which often has strong randomness. We need to determine whether this significant or nonsignificant effect is necessary or random, so "statistical hypothesis testing" provides an important theoretical basis[45,46]. Next, a t test based on pairwise data is used to verify whether different *p*-norms have a significant impact on the classification performance of SVM algorithms in 6 datasets.

The specific operation steps are as follows.

(i) Given the two different norm distances $p_1$ and $p_2$, we perform 10 times fivefold cross-validation under the two norm distances. The two groups of classification evaluation indices of the SVM algorithm are obtained, including precision, recall and Kappa coefficient. They are represented as $x_i$ and $y_i$.
(ii) Let $d_i = x_i - y_i \sim N(\mu, \sigma^2)$, $H_0 : \mu = 0$; $H_1 : \mu \neq 0$;
(iii) Let the statistic

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \sim t(n-1), \tag{27}$$

where $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$, $s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2}$ and the significance level $\alpha = 0.05$;

(iv) The $t$ value in Eq. (27) is calculated. If $|t| > t_{1-\alpha/2}(n-1)$, then in a statistical sense, different *p*-norm distances have a significant effect on the SVM performance; otherwise, different *p*-norm distances do not have a significant effect on the SVM performance.

The null hypothesis and the alternative hypothesis in Step (ii) are equivalent to $H_0$ : the use of different *p*-norm distances has a significant effect on the classification performance of SVM, and $H_1$ : the use of different *p*-norm distances has no significant effect on the SVM classification performance. The critical value is $t_{0.975}(9) = 2.262$ in Step (iv).

To compare whether different *p*-norm distances have significant effects on the performance of the proposed algorithm, the principle of the *p*-norm setting is as follows:

(i) Let $p \in [a, b]$, and the step size is $\lambda$;
(ii) The algorithm performance $MI_i, i = 1, 2, ...s$ is calculated corresponding to different norms $p_i$;
(iii) When $|MI_i - MI_j| \geq \varepsilon, 1 \leq i, j \leq s$, the corresponding $p_i$ and $p_j$ are fixed.

For convenience, let $a = 1, b = 10, \lambda = 0.5, \varepsilon = 0.1$. If $\left|MI_i - MI_j\right| \geq \varepsilon$ does not exist in $[a, b]$, $\varepsilon$ is reduced appropriately. For the 6 datasets in the experiment, the 2-level $p$-norm distance is set, and the specific information is shown in Table 7.

According to the above steps, the test statistic is calculated and compared to the critical value. The test results are shown in Table 8.

For the Kidney, Sonar and Pima datasets, the test results in Table 8 show that there is a significant difference in accuracy, recall and Kappa coefficient. For the other three datasets, there is no difference in accuracy, recall or Kappa coefficient at different $p$-norm levels, which is basically consistent with the results shown in Figs. 2, 3, and 4. In summary, it can be concluded that the change in the $p$-norm distance for different datasets will have different influences on the classification performance of SVM. In some datasets, such as the Sonar, Pima and Kidney datasets, the influence of the change in the $p$-norm distance is significant; in other datasets, such as the Postcode, Dermatology and Breast datasets, the influence is of the change in the $p$-norm distance is minimal. Therefore, when the kernel functions have the form of the $p$-norm distance, such as $p$-norm t kernel constructed in this paper and the traditional Gaussian kernel, we need to consider the influence of the norm distance on the performance of SVM and obtain the appropriate norm distance through experimental analysis to achieve the best classification prediction effect of SVM.

## Conclusions

For the classical SVM algorithm, the kernel function plays a crucial role in the classification prediction process because an appropriate kernel function can map samples to an appropriate feature space so that similar samples are close together and different samples are far apart. In view of this characteristic of the SVM algorithm, the $p$-norm distance t kernel is constructed according to the $t$ probability density function, and a strict theoretical proof is given. To make use of the advantages of different types of kernel functions, the kernel functions are combined. The affinity matrix is redefined according to the local kernel polarization, and then an optimization model is established to solve the weight coefficients and kernel parameters. The weighted $p$-norm t kernel is applied to the SVM classification. Experimental analysis on six datasets shows that the proposed weighted $p$-norm t kernel can effectively improve the classification prediction performance of the SVM algorithm compared with the traditional single kernel function. Finally, the influence of the $p$-norm distance on the performance of the SVM algorithm is analysed based on a statistical comparison test. It is concluded that for different datasets, different norm distances will have different effects on the performance of the algorithm, some of which are significant and some of which are minimal.

The multiple kernel method based on improved local polarization in this paper is applied to SVM classification. Our method is also suitable for dimensionality reduction, kernel clustering and medical drug screening. In future work, this method will be improved and generalized in these research directions. However, the proposed method in this paper is only a simple linear combination of multiple kernel functions. There is no complete and effective theoretical basis for the selection and combination of kernel functions; the optimization of kernel weights and kernel parameters still faces the problem of nonconvergence, which needs to be further solved.

Received: 14 December 2021; Accepted: 29 March 2022
Published online: 13 April 2022

## References

1. Vapnik, V. N. *The nature of statistical learning theory* (Springer, 1995).
2. Joachims, T. Text classification with support vector machines: Learning with many relevant features, in *Proceedings of the 10th European Conference on Machine Learing (ECML)*, 137–142 (Chemnitz, 1998).
3. Cui, M., Wang, Y., Lin, X. & Zhong, M. Fault diagnosis of rolling bearings based on an improved stack autoencoder and support vector machine. *IEEE Sens. J.* **21**(4), 4927–4937 (2021).
4. Gu, B. & Sheng, V. S. A robust regularization path algorithm for v-support vector classification. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(5), 1241–1248 (2017).
5. Shang, R. *et al.* Unsupervised feature selection based on kernel fisher discriminant analysis and regression learning. *Mach. Learn.* **108**(4), 659–686 (2019).
6. Welchowski, T. & Schmid, M. Sparse kernel deep stacking networks. *Comput. Stat.* **34**(3), 993–1014 (2019).
7. Wang, T., Zhang, L. & Hu, W. Bridging deep and multiple kernel learning: A review. *Inf. Fus.* **67**(2), 3–13 (2021).
8. Roman, I. *et al.* In-depth analysis of SVM kernel learning and its components. *Neural Comput. Appl.* **33**, 6575–6594 (2021).
9. Scholkopf, B., Smola, A., Muller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**(5), 1299–1319 (1998).
10. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R. Fisher discriminant analysis with kernels, in *Proceedings of the Conference on Neural Networks for Signal Processing.* Washington D. C., USA: IEEE: pp. 41–48 (1999).
11. Si, Y., Wang, Y. & Zhou, D. Key-performance-indicator-related process monitoring based on improved kernel partial least squares. *IEEE Trans. Industr. Electron.* **68**(3), 2626–2636 (2021).
12. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. & Watkins, C. Text classification using string kernels. *J. Mach. Learn. Res.* **2**, 419–444 (2002).
13. Scholkopf, B. *et al.* (eds) *Kernel methods in computational biology* (The MIT Press, 2004).
14. Ong, C. S., Smola, A. J. & Williamson, R. C. Learning the kernel with hyperkernels. *J. Mach. Learn. Res.* **6**(7), 1043–1071 (2005).
15. Zheng, D. N., Wang, J. X. & Zhao, Y. N. Nonflat function estimation with a multiscale support vector regession. *Neurocomputing* **70**(1–3), 420–442 (2006).
16. Gonen, M. & Alpaydin, E. Multiple kernel learning algorithms. *J. Mach. Learn. Res.* **12**, 2211–2268 (2001).
17. Rakotomamonjy, A. *et al.* SimpleMKL. *J. Mach. Learn. Res.* **9**(3), 2491–2521 (2008).
18. Fan, Q., Wang, Z., Zha, H. & Gao, D. MREKLM: A fast multiple empirical kernel learning machine. *Pattern Recogn.* **61**, 197–209 (2017).
19. Yang, L., Zhichuan, Z., Alin, H., *et al.* Pulmonary Nodule Recognition Based on Multiple Kernel Learning Support Vector Machine-PSO. Computational & Mathematical Methods in Medicine 2018, 1–10 (2018).

20. Gao, W. & Peng, Y. Hyperspectral image classification based on multiple kernel learning with Mahalanobis distance. *Chinese J. Sci. Instr.* **39**(3), 250–257 (2018).
21. Wang, H., Xu, D. & Martinez, A. Parameter selection method for support vector machine based on adaptive fusion of multiple kernel functions and its application in fault diagnosis. *Neural Comput. Appl.* **32**, 183–193 (2020).
22. Ergul, U. & Bilgin, G. MCK-ELM: Multiple composite kernel extreme learning machine for hyperspectral images. *Neural Comput. Appl.* **32**, 6809–6819 (2020).
23. Smola, A. & Scholkopf, B. A tutorial on support vector regression. *Stat. Comput.* **14**(3), 199–222 (2004).
24. Tanabe, H., Ho, T.B., Nguyen, C.H., & Kawasaki, S. Simple but effective methods for combining kernels in computational biology. in *Proceedings of IEEE International Conference on Research, Innovation and Vision for the Future* (2008).
25. Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E. & Jordan, M. I. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.* **5**(1), 27–72 (2004).
26. Sonnenburg, S., Ratsch, G., Schafer, C. & Scholkopf, B. Large scale multiple kernel learning. *J. Mach. Learn. Res.* **7**(7), 1531–1565 (2006).
27. Chen, Q., Liu, Z., Ma, X. & Wang, Y. Artificial neural correlation analysis for performance-indicator-related nonlinear process monitoring. *IEEE Trans. Industr. Inf.* **18**(2), 1039–1049 (2022).
28. Lou, Z. & Wang, Y. New nonlinear approach for process monitoring: Neural component analysis. *Ind. Eng. Chem. Res.* **60**(1), 387–398 (2021).
29. Kingsbury, N., Tay, D.B.H., Palaniswami, M. Multi-scale kernel methods for classification, in *Proceedings of the IEEE Workshop on Machine Learning for Signal Processing*. Washington D. C., USA, IEEE, pp. 43–48 (2005).
30. Zheng, D. N., Wang, J. X. & Zhao, Y. N. Non-flat function estimation with a multi-scale support vector regession. *Neurocomputing* **70**(1–3), 420–429 (2006).
31. Weinberger, K. Q. & Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* **10**(1), 207–244 (2009).
32. Sheng, W. G. Properties and construction methods of kernel in support vector machine. *Comput. Sci.* 33(6), 172–175 (2006).
33. Vapnik, V. N. *Statistical learning theory* (Wiley, 1998).
34. Wang, H. Q. *et al.* On multiple kernel learning Methods. *Acta Automatica Sinica* **36**(8), 1037–1050 (2010).
35. Lee, W.J., Verzakov, S., Duin R.P. Kernel combination versus classifier combination, in *Proceedings of the 7th International Workshop on Multiple Classifier Systems*. Springer, pp. 22–31 (2007).
36. Cristianini, N., Shawe-Taylor, J., Elisseeff, A. & Kandola, J. On kernel-target alignment. *Proc. Adv. Neural Inf. Process. Syst.* **14**(5), 367–373 (2001).
37. Baram, Y. Learning by kernel polarization. *Neural Comput.* **17**(6), 1264–1275 (2005).
38. Wang, T. H., Tian, S. F., Huang, H. K. & Deng, D. Y. Learning by local kernel polarization. *Neurocomputing* **72**(13–15), 3077–3084 (2009).
39. Birgin, E. G. & Martínez, J. M. Improving ultimate convergence of an augmented Lagrangian method. *Optim. Methods Softw.* **23**(2), 177–195 (2008).
40. Johnson, S. G. The NLopt nonlinear-optimization package, http://ab initio.mit.edu/nlopt (2020).
41. Franklin, J. The elements of statistical learning: Data mining, inference and prediction. *Math. Intell.* **27**(2), 83–85 (2005).
42. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**, 530–536 (2002).
43. Hoshida, Y. *et al.* Subclass mapping: identifying common subtypes in independent disease data sets. *PLoS ONE* **2**(11), e1195 (2007).
44. Svanholm, H. *et al.* Reproducibility of histomorphologic diagnoses with special reference to the kappa statistic. *APMIS* **97**(8), 689–698 (2010).
45. Wellek, S. *Testing statistical hypotheses of equivalence and noninferiority* (CRC Press, 2010).
46. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006).


## Acknowledgements

This research is supported by Qiannan Science and Technology Project (No. 2019XK03ST) and Natural Science Foundation of Guizhou Province Education Department (grant numbers QianJiaoHeKYZi [2019]200).

## Author contributions

Conceptualization, W.L. and S.L.; methodology, W.L.; software, W.L. and X.Q; validation, W.L., S.L. and X.Q.; formal analysis, W.L.; investigation, S.L.; resources, X.Q.;writing—original draft preparation, W.L. and X.Q.; writing—review and editing, W.L. and S.L.; visualization, W.L.; supervision, S.L.; project administration, X.Q.; funding acquisition, W.L. and X.Q. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-09766-w.

**Correspondence** and requests for materials should be addressed to W.L.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.