AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°1; pp 93-109
Submitted Jan. 2017; Revised March 15, 2017, Accepted April 15, 2017

A Probability Distribution Kernel based on Whitening Transformation

Jiangsheng Gui, Yuanfeng Chi, Qing Zhang, Xiaoan Bao

The college of information, Zhejiang Sci-Tech University
Hangzhou 310018, China (dewgjs@126.com)

Abstract

A kernel function can change linearly inseparable vectors into linearly separable vectors by mapping the original feature space into another feature space. But the dimension of the new feature space is usually several times of the original feature space, which results in a more computational complexity. This paper aims at introducing a Dirichlet probability distribution kernel based on whitening transformation which is called DPWT kernel function for short. The DPWT kernel function first mapping feature vectors of the original feature space into new vectors of another same dimension feature space, and then classifies new vectors in the new feature space so as to achieve the purpose of classifying original feature vectors. The DPWT kernel doesn't augment the dimension of the new feature space. What's more, the DPWT kernel can effectively eliminate the correlation between vectors, and reduce the redundancy of data which can further improve the accuracy of classification. In this paper, we use the DPWT kernel and other five commonly used kernels on the three benchmark datasets (VOC2007, UIUCsport and Caltech101) for image classification experiments. Experiments show that the DPWT kernel exhibits superior performances compared to the other state of the art kernels.

Key words

Whitening transformation, Dirichlet probability distribution, Kernel function， Image classification

1. Introduction

The study of kernel functions has caused a boom in image classification. People usually use a high efficiency and low cost linear classifier for image classification. But feature vectors are linear inseparable in low dimensional space. According to recognition theory, the low dimensional linear inseparable vectors can become linear separable in the high dimensional feature space by kernel functions. Radial basis kernel (RBF) function is a common kernel function, and many scholars have made a deep research on the RBF kernel function. Rahimi uses a positive definite shift-invariant kernel to approximate the RBF kernel function, which further improves the performance of the RBF kernel function [1]. Vempati uses the display features to map the RBF kernel function, which makes a trade-off between the high computation complexity and high classification accuracy [2]. Additive kernels are also commonly used kernel functions, and the additive kernels include an Intersection kernel [3], a Homker kernel and a Hellinger kernel [4]. Perronnin proposes that simply obtain the square-root of the feature vectors can lead to large improvements, and the step perfect interprets the basic principles of the Hellinger kernel function [5]. In order to compare the effectiveness of additive kernels, Vedaldi proposes a unified framework for all additive kernels [6]. That framework can use a closed form express all common kernels and experiments show that the performance of the Homker kernel is the best. In order to adapt to a wide range of data samples, Guillaumin proposes a multiple kernel learning (MKL) for image classification [7]. The method can effectively eliminate the noise samples and improves the classification accuracy of samples. In order to increase the discriminative power of an object categorization method, it is effective to model the intra category diversity and the inter category correlation among images. According to that idea, Yang introduces a group-sensitive multiple kernel learning method to accommodate the diversity and the correlation, the experiments show that the group-sensitive multiple kernel learning method has achieved quite good performance in three challenging datasets [8]. In order not to be limited by the performance of the existing kernel function, Tom proposes a gene kernel function [9], the gene kernel function shows better performance compared with the polynomial kernel function, the gauss kernel function and the sigmoid kernel function. Chen proposes a concept of an inner double kernel and the kernel in that inner double kernel function can be arbitrary kernel function [10]. The handwritten recognition

rate of the inner double kernel function is higher than the Gaussian kernel function and polynomial kernel function. Ionescu proposes a new kernel function that is called PQ kernel [11]. The PQ kernel defines the number of concordant pairs between two vectors is P. In the same way, the number of discordant pairs is Q. Experiment shows that the PQ kernel which combined with the JS (Jensen-Shannon) kernel can further improve the image classification accuracy. Wang proposes a fast stochastic intersection kernel machine training algorithm (SIKMA), the method is more effective than the linear kernel in image matching, retrieval and classification [12]. Considering the computational complexity of the kernel functions will be very high, Chang proposes a probability integral kernel as the similarity measure of vectors and experiments show that the probability integral kernel reduced the computing time dramatically [13].

Although many people use kernel function to measure the similarity between feature vectors [14][15][16], but compared with the original features, the kernel function increased feature space's dimension after mapping features, and add the computational complexity. In order to solve that problem, we introduce a dirichlet probability distribution kernel based on whitening transformation which is called DPWT kernel function for short. The main purpose of the DPWT kernel function is to map the feature vectors of the original feature space into new vectors of another feature space which has a same dimension, and then classify new vectors so as to achieve the purpose of classifying original feature vectors. The DPWT kernel function not only keeps the same dimension between the new feature space and the original feature space, but also effectively eliminates the correlation between the dimensions and reduces the data redundancy.

In this paper, we introduced the basic concept of whitening transformation at first, and then the Dirichlet probability distribution kernel (DPWT) based on whitening transformation is introduced in detail. Finally, the validity of DPWT kernel function on image classification is verified on VOC2007, UIUCsport and Caltech101 datasets.

2. Foundations of the DPWT kernel

The whitening transformation is a kind of decorrelation linear transformation, which can transform a d dimension vector $J(x)$ whose mean is μ and covariance matrix is G into a new

random vector $z(x)$.

$$Z(x) = AJ(x) \tag{1}$$

where the dimension of vector $z(x)$ is also d, and the covariance matrix of vector $z(x)$ is the unit matrix, A represents a whitening vector that satisfies the condition $A^TA=G^{-1}$. The covariance matrix of vector $z(x)$ is a unit matrix, so the random variables of the vector are irrelevant, and the variance of all random variables is 1, so the whitening transform can effectively eliminate the correlation between vectors, and reduce the redundancy of data. There are many kinds of whitening transformations, in this paper, we choose the simplest Mahalanobis whitening transform [17].

The mean of gradient vector $H(x)$ of the ln-likelihood function for any probability distribution is:

$$\begin{aligned} E\left[H(x)\right] &= E\left[\nabla_\beta \ln p(x,\beta)\right] \\ &= E\left[\frac{\nabla_\beta p(x,\beta)}{p(x,\beta)}\right] \\ &= \int \frac{\nabla_\beta p(x,\beta)}{p(x,\beta)} p(x,\beta)\,dx \\ &= \int \nabla_\beta p(x,\beta)\,dx \\ &= \nabla_\beta 1 \\ &= 0 \end{aligned} \tag{2}$$

$\beta = [\beta_1, \beta_2, ..., \beta_W]^T$ is the distribution parameter when the Dirichlet probability distribution $p(x,\beta)$ have the max ln-likelihood function.

The covariance matrix of vector $H(x)$ is:

$$G = E\left[\left(H(x) - E\left[H(x)\right]\right)\left(H(x) - E\left[H(x)\right]\right)^T\right] = E\left[H(x)H^T(x)\right] \tag{3}$$

According to the definition of the Mahalanobis whitening transformation, for any vector $H(x)$ whose mean value is 0 and the covariance matrix is G, the vector $\Psi(x)$ after whitening transformation is defined as the following form:

$$\Psi(x) = G^{-1/2} H(x) \tag{4}$$

In order to verify the applicability of Mahalanobis whitening transformation to this paper, we calculate the expectation and covariance matrix of vector $\Psi(x)$ according to formula (5) and formula (6),

$$E\left[\Psi(x)\right] = E\left[G^{-1/2} H(x)\right] = G^{-1/2} E\left[H(x)\right] = 0 \tag{5}$$

$$\begin{aligned} Cov\left(\Psi(x)\right) &= E\left[\left(\Psi(x) - E\left[\Psi(x)\right]\right)\left(\Psi(x) - E\left[\Psi(x)\right]\right)^T\right] \\ &= E\left[\Psi(x)\Psi^T(x)\right] \\ &= E\left[\left(G^{-1/2} H(x)\right)\left(G^{-1/2} H(x)\right)^T\right] \\ &= G^{-1/2} E\left[H(x) H^T(x)\right] G^{-1/2} \\ &= G^{-1/2} G G^{-1/2} \end{aligned} \tag{6}$$

By the definition of covariance matrix, we can know that G is a semi positive definite symmetric matrix, so there is a positive definite symmetric square root $G^{1/2}$ can make sure $G^{1/2}G^{1/2}=G$.

$$Cov\left(\Psi(x)\right) = I \tag{7}$$

where I represents the unit vector. By formula (5) and formula (7), we can know that the expectation of the vector $\Psi(x)$ is 0, and the covariance matrix of the vector $\Psi(x)$ is the unit matrix. Therefore, in order to effectively eliminate the correlation between vectors, and reduce the redundancy of data, we can carry out the whitening transformation of the gradient vector $H(x)$.

3. DPWT kernel

The main idea of the DPWT kernel function is to map the feature vector x, y of the original feature space into vector $\Psi(x)$, $\Psi(y)$of the new feature space, and then classify new vectors so as to achieve the purpose of classifying original feature vectors.

We use Dirichlet probability distribution to get better gradient vector $H(x)$.Given a feature vector x=$[x_1, x_2, \ldots, x_w]^T$, the formula of Dirichlet probability distribution is:

$$D(\beta) = p(x, \beta) = \frac{\Gamma(\sum_{i=1}^{W} \beta_i)}{\prod_{i=1}^{W} \Gamma(\beta_i)} \prod_{i=1}^{W} x_i^{\beta_i - 1} \tag{8}$$

Where w is a feature dimension which is equal to the number of parameters of the model. $\beta = [\beta_1, \beta_2, ..., \beta_W]^{\mathrm{T}}$ denotes model parameter. $\Gamma(\beta) = \int_0^{\infty} \beta^{t-1} e^{-t} dt = (\beta - 1)!$, Γ indicates a gamma function.

The DPWT kernel function is defined as:

$$K_{DPWT}(x, y) = \langle \Psi(x), \Psi(y) \rangle = \left\langle G^{-1/2} H(x), G^{-1/2} H(y) \right\rangle = H^T(x) G^{-1} H(y) \tag{9}$$

where vector $\Psi(x)$ and vector $\Psi(y)$ are obtained by whitening transformation. $H(x)$ is the gradient vector of the ln-likelihood function of Dirichlet probability distribution, where ∇_β expresses the derivation of β.

$$H(x) = \nabla_\beta \ln p(x, \beta)$$

$$= \nabla_\beta \ln \frac{1}{B(\beta)} \prod_{i=1}^{W} x_i^{\beta_i - 1}$$

$$= \nabla_\beta \ln \frac{\Gamma\left(\sum_{i=1}^{W} \beta_i\right)}{\prod_{i=1}^{W} \Gamma(\beta_i)} \prod_{i=1}^{W} x_i^{\beta_i - 1}$$

$$= \nabla_\beta \ln \Gamma\left(\sum_{i=1}^{W} \beta_i\right) - \nabla_\beta \ln \prod_{i=1}^{W} \Gamma(\beta_i) + \nabla_\beta \ln \prod_{i=1}^{W} x_i^{\beta_i - 1}$$

$$= \frac{\Gamma'\left(\sum_{i=1}^{W} \beta_i\right)}{\Gamma\left(\sum_{i=1}^{W} \beta_i\right)} \nabla_\beta(\gamma) - \nabla_\beta \sum_{i=1}^{W} \ln \Gamma(\beta_i) + \nabla_\beta \sum_{i=1}^{W} (\beta_i - 1) \ln x_i$$

$$= \frac{\Gamma'\left(\sum_{i=1}^{W} \beta_i\right)}{\Gamma\left(\sum_{i=1}^{W} \beta_i\right)} [1, 1, ..., 1]^{\mathrm{T}} - \left[\frac{\Gamma'(\beta_1)}{\Gamma(\beta_1)}, \frac{\Gamma'(\beta_2)}{\Gamma(\beta_2)}, .., \frac{\Gamma'(\beta_W)}{\Gamma(\beta_W)} \right]^{\mathrm{T}} + [\ln x_1, \ln x_2, ..., \ln x_W]^{\mathrm{T}} \tag{10}$$

since

$$\frac{\Gamma'(\lambda)}{\Gamma(\lambda)} = \frac{d}{d\lambda} \ln \Gamma(\lambda) = \varphi(\lambda) \tag{11}$$

so

$$H(x) = \varphi\left(\sum_{i=1}^{W} \beta_i\right) \mathbf{1} - \varphi(\beta) + \ln(x) \tag{12}$$

where φ is a digamma function, $\varphi\left(\sum_{i=1}^{W} \beta_i\right)$, $\ln(x)$ and $\varphi(\beta)$ are all input functions, and $\mathbf{1} \in R^W$ is a vector which all elements are 1. The covariance matrix of the formula (9) can be obtained by formula

(13):

$$G = \int p(x,\beta)[\nabla_\beta \ln p(x,\beta)][\nabla_\beta \ln p(x,\beta)]^{\mathrm{T}} dx = diag\{\varphi'(\beta)\} - \varphi'(\sum_{i=1}^{W}\beta_i)B \tag{13}$$

$$\varphi'(\beta) = \frac{d^2}{d\beta^2}\ln\Gamma(\beta) \tag{14}$$

diag(.) is the diagonal matrix which is generated from the input vectors. φ' is a trigamma function. B is a square matrix which elements are all 1, and the dimension of B is $w \times w$. G is a constant vector and the value of G has nothing to do with x and y.

The DPWT kernel function can be further written as:

$$K_{DPWT}(x,y) = \left[\ln(x) - \left\{\varphi(\beta) - \varphi\left(\sum_{i=1}^{W}\beta_i\right)\mathbf{1}\right\}\right]^T G^{-1}\left[\ln(y) - \left\{\varphi(\beta) - \varphi\left(\sum_{i=1}^{W}\beta_i\right)\mathbf{1}\right\}\right] \tag{15}$$

We set

$$\left\{\varphi(\beta) - \varphi\left(\sum_{i=1}^{W}\beta_i\right)1\right\} = \mu_\beta \tag{16}$$

Therefore, the DPWT kernel function can be rewritten as:

$$K_{DPWT}(x,y) = [\ln(x) - \mu_\beta]^T G^{-1}[\ln(y) - \mu_\beta] \tag{17}$$

According to the the formula (17), we can see that the value of the DPWT kernel function is closely related to the value of ln(x) and ln(y).

4. Parameter Optimization

The Ln function is very important for the DPWT kernel function. It should be noted that $\ln(x) \to \infty$ when $x \to 0$, at this point, the kernel function is obviously unreasonable, so we modify ln(x) to ln($x+f$) using small parameter f<<1. The optimized DPWT kernel function is:

$$K_{DPWT} = [\ln(x+f) - \mu_\beta]^T \left[diag\{\varphi'(\beta)\} - \varphi'(\sum_{i=1}^{W}\beta_i)B\right]^{-1}[\ln(y+f) - \mu_\beta] \tag{18}$$

The basic problem of support vector learning (SVL) is the selection of kernel function and kernel parameters, now we further study the modified parameters f in the formula (18).

Assume that the parameters β_i are not affected by the change of parameters i, we give the marginal distribution of Dirichlet distribution by Beta distribution, as shown in formula (19):

$$p(x,\beta) = \frac{x^{\beta_i - 1}(1-x)^{\beta_0 - \beta_i - 1}}{B(\beta_i, \beta_0 - \beta_i)} = \frac{x^{m-1}(1-x)^{n-1}}{B(m,n)} = p(x,m,n) \tag{19}$$

$m=\beta_i$, $n=\beta_0-\beta_i$. First, the variable $v=\ln(x)$ is used to instead of the variable x to optimize the formula (19) into a formula (20):

$$p(v;m,n)=\frac{\{\exp(v)\}^{m-1}\{1-\exp(v)\}^{n-1}}{B(m,n)} \tag{20}$$

And then according to the variable transformation formula $U=\ln\{\exp(v)+f\}$, the $\ln(x)$ is optimized to be written in the form of $\ln(x+f)$, as shown in the formula (21):

$$p(U,m,n)=\frac{\{\exp(U)-f\}^{m-1}\{1+f-\exp(U)\}^{n-1}}{B(m,n)} \tag{21}$$

Function $\ln(x+f)$ acts on the following two aspects: on the one hand, it is roughly removed the $\ln(x)$ in the $x<f$ and only keep $\ln(f)$.The tiny histogram value is treated as 0, which enhances the difference between the appropriate small x, like a local contrast enhancement, which reflects the negative example theory [17]. At the same time, by determining the value of f, $U=\ln(x+f)$ forms a smooth distribution on the top of $\ln(f)$, and it can keep a smooth distribution state around the distribution mode. During the process of determining f, too small f pushes the lower boundary $\ln(f)$ away from the distribution pattern, which exaggerates the negative example theory. On the other hand, The larger the f will merge the low boundary $\ln(f)$ with the distribution pattern, which is contrary to the theory of negative examples. So, the suitable f can show the smooth distribution between the distribution pattern and the low boundary $\ln(f)$.

In order to get the appropriate f, we define f based on the cumulative percentage of the empirical distribution, we usually set the cumulative percentage of 25%, the optimization parameter $f=p^{-1}(0.25)\approx 0.001$, which $P(f)=\int_0^f p(x)dx$.

Assuming that the dimension of the original feature vector x is $w\times 1$, then the feature vector $\Psi(x)$ after the DPWT kernel function mapping has the following form:

$$\Psi(x)=G^{-1/2}H(x)=\left(diag\{\varphi'(\beta)\}-\varphi'(\sum_{i=1}^{W}\beta_i)B\right)^{-1/2}\left[\ln(x+f)-\mu_\beta\right] \tag{22}$$

According to the formula (22), we can see that the dimension of vector $\Psi(x)$ is still $w\times 1$, so the dimension of the feature space after the DPWT kernel function mapping is consistent with the original feature space.

5. Experiments

In this paper, we use three datasets (VOC2007 dataset, UIUCsport dataset and Caltech101 dataset) to carry out the image classification experiments. Several state of the art kernels (Homker[6], JS+PQ[11], SIKMA[12], MKL[7], Linear) were compared with the DPWT kernel on the three datasets. Our image classification is performed using a PC with an Intel Core 7-3770 CPU, 3.4 GHz, 8 GB RAM, and the algorithm is developed using MATLAB R2014a and GCC4.6 compiler under an Ubuntu12.04 (64-bit) operating system.

5.1. Dataset

In order to make experiments more diversity, we choose VOC2007 dataset, UIUCsport dataset and Caltech101 dataset for image classification. Our first set of experiments is on VOC2007 dataset and VOC2007 dataset is composed of color images with multi labels. VOC2007 dataset contains 9963 images and includes 24640 annotated objects. The VOC2007 dataset is divided into 20 categories (Aeroplane, Bicycle, Bird, Boat, Bottle and so on), each of which contains 193~4015 images. What's more, a large number of images are medium resolution (500*333 pixels). We use an identical experimental setup as Chatfield [18], and about 50% images of the dataset are used for training/validation samples and others for testing samples. UIUCsport dataset is selected as the second dataset in our experiments and UIUCsport dataset is composed of color images with a single label. UIUCsport dataset contains 8 categories (rowing, badminton, polo, bocce, snowboarding, croquet, sailing, rock climbing) which are all sports event. There are 1579 images in the dataset and there are about 200 images in each category. For each category, we randomly selected 70 images for training and 60 for testing. The third dataset is Caltech101 and Caltech101 is composed of color and gray images. There are 9144 images in Caltech101 dataset and these images are distributed into 102 categories (101 object categories and a background category), each of which contains 40~800 images. Furthermore, each image of the dataset is about 350 x 200 pixels. For each category, we select 15 random images for training and test on the rest.

5.2. Procedure

In this paper, the training samples and test samples of each data set need four steps: feature extraction, feature clustering, feature coding, and histogram formation, then the training samples

are input into the SVM model to determine the decision function of the model, the test samples are input into the SVM model to get the type of each image in the test sample. The parameterbetween training samples and test samples are same in the four steps, for the sake of simplicity, we use the training sample to illustrate the parameters of the four steps. The first step, using SIFT dense sampling methods on three data sets of training images for feature extraction, Then, 1000000 feature points are randomly selected from the feature points extracted from each data set, which is used for feature clustering. The second step, the K mean algorithm is used to cluster the feature points of the three data sets into 3000 feature types, and the number of iterations is set to 150.The third step, we use KCB coding to encode the features, in order to improve the efficiency of the computation, each feature is encoded by considering five visual words which are closest to their own [20].The fourth step, we using spatial Pyramid model to form visual word histogram for each image, the image is divided into 3 layers, the three layer includes 1*1, 3*1 (three horizontal strips), and 2*2 (four quadrants) bins, for a total of 8 bins [19].The fifth step, The histogram vector of the training image is input into the SVM model to complete the training of the SVM model, and then test the image input to the SVM model to get the category of each image, complete the SVM model test. We choose the DPWT kernel (optimization parameter set to 0.001) and other five kernels (Homker, JS+PQ, SIKMA, MKL, Linear) for SVM model in image classification. We set the penalty parameter to 7.6, which expresses the tolerance of the misclassified data items. What's more, we use the one-versus-rest principle to put the feature vectors of the testing images into SVM to get categories of the testing images [19]. Specific steps are as follows: first set the first class of the data set as a positive sample, all other categories are negative samples, then each test image is input into the SVM model to calculate the value of the decision function, if the value of the decision function is greater than 0, indicates that the test image is a positive sample, otherwise it is a negative sample, then take the next category as a positive sample, and all the other categories as negative samples, continue to run the decision function until all categories as a positive sample.

In this paper, there are two experimental evaluation index, according to PASCAL VOC2007's official statement: accuracy / recall curve is more intuitive and more sensitive to the performance evaluation than the receiver operating characteristic curve. In this paper, we use the

mAP value of a method on a given set of data indicates the average accuracy of the method in all categories, if a mAP value is high, then the classification performance is better. The PR curve is another indicator of the experiments in this paper, we utilize the PR curve to analyze the classification performance of different methods. If the PR curve of a certain method is closer to the upper right corner, the classification performance of the method is better. To obtain reliable results, we repeated the experimental process 5 times for each method.

5.3. Experiments Results

Table 1 and table 2 show the average precision for each category of six methods (DPWT, Homker, JS+PQ, SIKMA, MKL, Linear) respectively on VOC2007 and UIUCsport datasets, furthermore, the highest average precision for each category is in bold. There are 102 categories in Caltech101 dataset, so we do not list the average precision for each category one by one. We just give the mean average precisions (mAPs) of six methods (DPWT, Homker, JS+PQ, SIKMA, MKL, Linear) on Caltech101 dataset in table 3. We select four categories of the highest classification accuracy on Caltech101 dataset in figure 1. What's more, we select four categories of the lowest classification accuracy on Caltech101 dataset in figure 2.

From table 1, table 2, and table 3, we can clearly see that the mAPs of five kernels were well above the mAP of the baseline linear kernel. What's more, in table1 and table 2, the mAP of the DPWT kernel in most of the categories is better than other kernels. In VOC2007 dataset, the DPWT kernel has obtained the best results for 19 out of 20 categories. In UIUCsport dataset, the DPWT kernel has obtained the best results for 7 out of 8 categories. To be concrete, in Tvmonitor category of VOC2007 dataset, the mAP of the DPWT kernel is 0.3194 above the mAP of the JS+PQ kernel, 0.2311 above the mAP of the SIKMA, 0.2105 above the mAP of the MKL method and 0.1885 above the mAP of the Homker kernel. In badminton category of UIUCsport dataset, the mAP of the DPWT kernel is 0.2672 above the mAP of the JS+PQ kernel, 0.1575 above the mAP of the SIKMA, 0.0816 above the mAP of the MKL method and 0.0624 above the mAP of the Homker kernel.

Table 1. Performance on VOC2007

Category	Linear	JS+PQ	SIKMA	MKL	Homker	DPWT
Aeroplane	0.5587	0.6236	0.6330	0.6431	0.6634	**0.6672**

Bicycle	0.3185	0.3880	0.4454	0.4918	0.5034	**0.5180**
Bird	0.1706	0.2775	0.3054	0.3506	0.3583	**0.3592**
Boat	0.4912	0.5309	0.5851	0.5933	0.5755	**0.5942**
Bottle	0.1141	0.1529	0.1581	0.2471	0.2454	**0.2535**
Bus	0.3233	0.4177	0.4812	0.5607	0.5323	**0.5619**
Car	0.5675	0.6339	0.6707	0.7105	0.7187	**0.7244**
Cat	0.3072	0.3954	0.4287	0.4739	0.4690	**0.4743**
Chair	0.3434	0.4254	0.4361	0.4533	0.4532	**0.4646**
Cow	0.1300	0.2827	0.2948	0.2702	**0.3120**	0.2929
Diningtable	0.2686	0.3009	0.3439	0.4065	0.4356	**0.4492**
Dog	0.1927	0.2915	0.3273	0.3520	0.3621	**0.3683**
Horse	0.6277	0.6675	0.6939	0.7337	0.7476	**0.7503**
Motorbike	0.2926	0.4256	0.4949	0.5620	0.5903	**0.5996**
Person	0.6326	0.7186	0.7575	0.7931	0.7939	**0.8009**
Pottedplant	0.0914	0.1406	0.1913	0.2078	0.2161	**0.2238**
Sheep	0.1388	0.2945	0.3343	0.3388	0.3602	**0.3655**
Sofa	0.2317	0.2953	0.3621	0.3916	0.3845	**0.4093**
Train	0.5603	0.5934	0.6656	0.7034	0.7132	**0.7168**
Tvmonitor	0.2463	0.2802	0.3685	0.3891	0.4111	**0.4367**
mAP	0.3304	0.4068	0.4489	0.4836	0.4923	0.5015

Table 2. Performance on UIUCsport

Category	Linear	JS+PQ	SIKMA	MKL	Homker	DPWT
rowing	0.8925	0.8740	0.9029	0.9109	0.9150	**0.9186**
badminton	0.4145	0.3517	0.4614	0.5373	0.5565	**0.6189**
polo	0.6434	0.8181	**0.8263**	0.7465	0.7859	0.8086
bocce	0.7966	0.8441	0.8584	0.8966	0.9195	**0.9208**
snowboarding	0.9139	0.9280	0.9327	0.9472	0.9632	**0.9774**
croquet	0.5734	0.5699	0.5451	0.5678	0.5876	**0.6268**
sailing	0.6450	0.9105	0.9496	0.9609	0.9890	**0.9955**
rock climbing	0.7501	0.7782	0.8178	0.8515	0.8835	**0.9095**
mAP	0.7037	0.7593	0.7868	0.8023	0.8250	0.8470

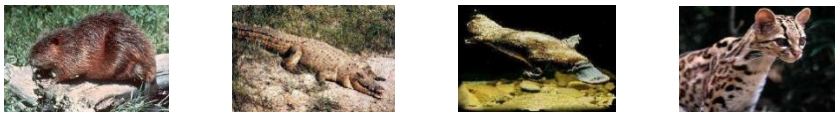
Table 3. Performance on Caltech101

Method	Linear	JS+PQ	SIKMA	MKL	Homker	DPWT
mAP	0.4232	0.5286	0.5807	0.584	0.6306	0.6576




car_side (100%) Motorbikes (99.96%) pagoda (100%) trilobite (100%)

Fig. 1 Four categories of the highest classification accuracy on Caltech101 dataset



Beaver (17.91%) crocodile (20.76%) platypus (20.20%) wild_cat (12.00%)

Fig. 2 Four categories of the lowest classification accuracy on Caltech101 dataset

What's more, Precision/Recall (PR) analysis gives more intuitive and sensitive evaluation than the ROC analysis. So we choose PR analysis to evaluate the performance of different kernels. We draw the PR curves of the six kernels in figure 3, figure 4, and figure 5. DPWT kernel depicted in red line, Homker kernel depicted in cyan line, JS+PQ kernel depicted in blue line, SIKMA method depicted in magenta line, MKL method depicted in green line and Linear kernel depicted in black line.

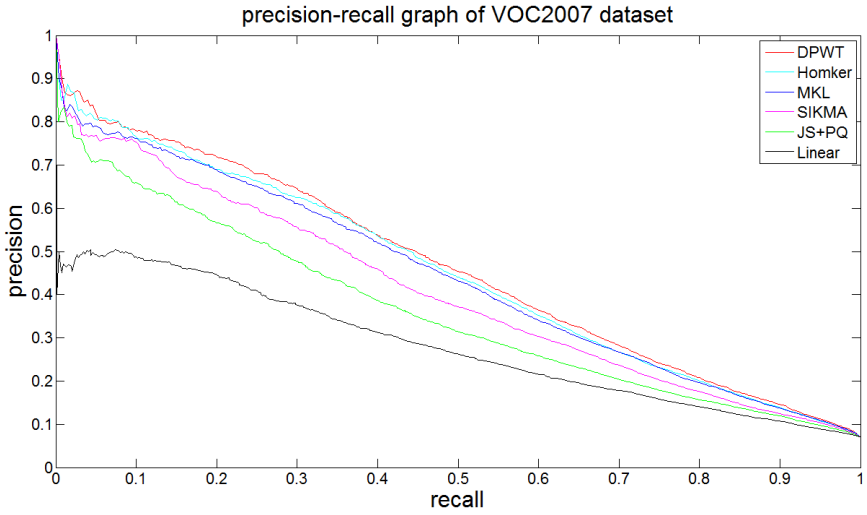


Fig. 3 The PR curves of six kernels on VOC2007 dataset

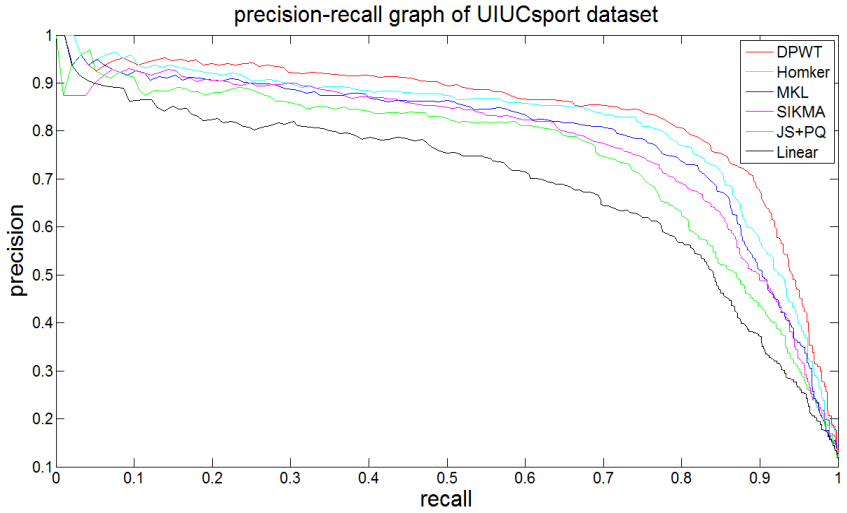


Fig. 4 The PR curves of six kernels on UIUCsport dataset

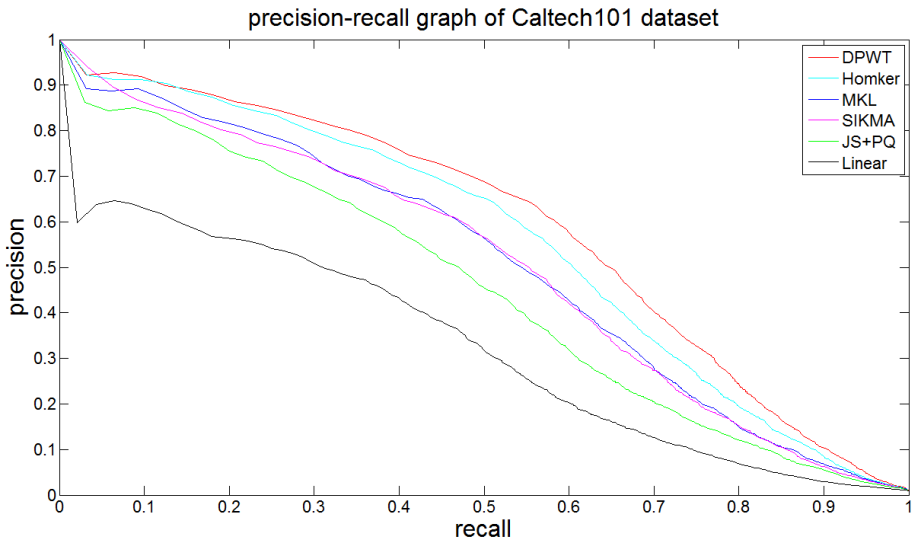


Fig. 5 The PR curves of six kernels on Caltech101 dataset

From the PR curves of VOC2007 dataset, it can be seen clearly that DPWT kernel is better than other five methods. On Caltech101 and UIUCsport datasets, the DPWT kernel has an excellent PR curve again. In terms of the PR curves, five kernels are obvious better than the baseline linear kernel. The MKL kernel combines an RBF kernel and a linear kernel for image classification, so it obtains a pretty good PR curve. However, the RBF kernel function is very sensitive to parameters, so the PR curve is lower than the Homker kernel function which is not affected by the parameters. The SIKMA kernel function is not only sensitive to the parameters, but also only takes into account the similarity between the original vectors, so the PR curve is slightly lower than the MKL kernel function. JS+PQ kernel function is related to the number of concordant and discordant pairs between two vectors, so it has a limited description and a poor PR curve.

As can be seen in table 4, the training time of six kernels were measured in seconds. The classification accuracy of linear kernel function is the lowest, and the training time is the shortest. SIKMA kernel combines a stochastic learning method with an intersection kernel, so the complexity of calculation is lower than a tradition intersection kernel. According to the formulation (22), the DPWT kernel has not improved the feature dimension, so it has a low computational complexity. The Homker kernel increases the feature dimension from 1W to 7W , so it increases the computation time. The JS+PQ kernel need to calculate the number of concordant and discordant pairs between two vectors, so the computation complexity is very

high. Because the MKL kernel requires multiple kernel matrices in the operation, the computational complexity is high. From table 4, the computational complexity of the DPWT kernel is smaller than Homker's, JS+PQ's and MKL's. Despite training time of the DPWT kernel is higher than the Linear and SIKMA kernel, the DPWT kernel has made great progress in precision.

Table 4. The training time for six kernels on three dataset

	Linear	SIKMA	DPWT	Homker	JS+PQ	MKL
VOC2007	46.70	96.22	149.05	961.45	5185.07	7374.96
UIUCsport	3.90	10.98	17.18	106.50	576.02	819.66
Caltech101	15.30	27.83	50.54	320.64	1710.23	2449.74

6. Conclusion and Further Work

This paper presents a new kernel, a dirichlet probability distribution kernel based on whitening transformation, which we called it as the DPWT kernel. Fair experiments using three benchmark datasets (VOC2007, UIUCsport and Caltech101) were carried out on six methods (DPWT, Homker, JS+PQ, SIKMA, MKL, Linear). From the experiments we conclude the following results. The mAP of the DPWT kernel is higher than the mAPs of other five kernels on three datasets. In terms of the computational time, the DPWT kernel is shorter than kernels (Homker, JS+PQ and MKL) and a little longer than the SIKMA kernel. On the one hand, the DPWT kernel improves the classification precision without increasing the feature dimension. On the other hand, the DPWT kernel can effectively eliminate the correlation between vectors, and reduce the redundancy of data. In future work, we will optimize the DPWT kernel algorithm and thoroughly study the optimization parameter of the DPWT kernel function. Furthermore, we will use a TF-IDF measure for features and eliminate the features which have a low TF-IDF score.

References

1. A. Rahimi, B. Recht. Random Features for Large-Scale Kernel Machines. Advances in Neural Information Processing Systems, 2007, vol. 20, pp. 1177-1184.
2. S. Vempati, A. Vedaldi, A. Zisserman, et al. Generalized RBF Feature Maps for Efficient Detection// British Machine Vision Conference, BMVC 2010, Aberystwyth, UK, August 31 - September 3, 2010. Proceedings. DBLP, 2010:1-11.

3. S. Maji, A.C. Berg, J. Malik. Classification using intersection kernel support vector machines is efficient[C]// Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008:1-8.
4. S. Maji, A.C. Berg, J. Malik. Efficient Classification for Additive Kernel SVMs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, vol. 35, no. 1, pp. 66-77.
5. F. Perronnin, J. Sánchez, Y. Liu. Large-scale image categorization with explicit data embedding. 2010, vol. 23, no. 3, pp. 2297-2304.
6. A. Vedaldi, A Zisserman. Efficient Additive Kernels via Explicit Feature Maps. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2012, vol. 34, no. 3, pp. 480.
7. M. Guillaumin, J. Verbeek, C. Schmid. Multimodal semi-supervised learning for image classification// Computer Vision and Pattern Recognition. IEEE, 2010, pp. 902-909.
8. J. Yang, Y. Li, Y. Tian, et al. Group-sensitive multiple kernel learning for object categorization// IEEE, International Conference on Computer Vision. IEEE, 2009, pp. 436-443.
9. T. Howley, M.G. Madden. The Genetic Kernel Support Vector Machine: Description and Evaluation. Artificial Intelligence Review, 2005, vol. 24, no, 3, pp. 379-395.
10. G.Y. Chen, P. Bhattacharya. Function Dot Product Kernels for Support Vector Machine. 2006, vol. 20, no. 2, pp. 614-617.
11. R.T. Ionescu, M. Popescu. Kernels for Visual Words Histograms// Image Analysis and Processing – ICIAP 2013. 2013, pp. 81-90.
12. G. Wang, D. Hoiem, D. Forsyth. Learning image similarity from Flickr groups using Stochastic Intersection Kernel MAchines// IEEE, International Conference on Computer Vision. IEEE Xplore, 2009, pp. 428-435.
13. H W Chang, H T Chen. A square-root sampling approach to fast histogram-based search[C]// Computer Vision and Pattern Recognition. IEEE, 2010, pp. 3043-3049.
14. D Chen, J M Phillips. Relative Error Embeddings for the Gaussian Kernel Distance. 2016.
15. C. Tonde, A. Elgammal. Learning Kernels for Structured Prediction using Polynomial Kernel Transformations. 2016.
16. H. Hong, B. Pradhan, M.N Jebur, et al. Spatial prediction of landslide hazard at the Luxi area

(China) using support vector machines. Environmental Earth Sciences, 2016, vol. 75, no. 1, pp. 1-14.

17. P.E. Szabó. Response to “Variable directionality of gene expression changes across generations does not constitute negative evidence of epigenetic inheritance” Sharma, A. Environmental Epigenetics, 2015, 1-5. Genome Biology, 2016, vol. 17, no. 1, pp. 1-4.
18. K. Chatfield, V. Lempitsky, A Vedaldi, et al. The devil is in the details: An evaluation of recent feature encoding methods// British Machine Vision Conference. 2011: 76.1-76.12.
19. X. Zhang, M.H. Mahoor. Task-dependent multi-task multiple kernel learning for facial action unit detection. Pattern Recognition, 2016, vol. 51, pp. 187-196.