# A Method of Learning the Kernel Using the Alignment

[*1,2,3]Zhiying Tan, [1,4]Yong Feng, [5]Kejia Xu, [1]Kun She, [3]Xiaobo Song
[*1]*School of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu 610054, China, tanzhiying1010@gmail.com*
[2]*Chengdu Inst. of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China,*
[3] *Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 213164, China,* sxb811023@163.com
[4]*Chongqing Inst. of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 401122, China,* yongfeng@casit.ac.cn
*5 School of electrical, South West Jiao Tong University，Chengdu 610031, China, meckyxu@gmail.com*

## *Abstract*

*Kernel alignment is an approach to measure the similarity between different kernel functions. Based on the property, some learning methods can be proposed. On some fixed data set, the kernel matrix can be totally determined by the kernel and the responding parameters. Then the similarity between the gram matrix and the kernel matrix is determined by the kernel function and the parameters. In the paper, we apply the kernel alignment to construct an optimal problem which is used to learn the kernel function and select the optimal parameters. The experiment results indicate that the recognition rate is basically consistent with the alignment. This means that selecting the bandwidth according to alignment is reasonable. The nonlinear features can be extracted by the kernel methods.*

**Keywords:** *Alignment, Kernel Methods, Kernel Matrix, Newton's Method, Gaussian Kernel*

## 1. Introduction

A generalization bounds were given by Bartlett and Shawe-Taylor [1]. For $\forall \delta > 0$, the learned classifier $h$ and the sample set $X = \{x \in R^n : \| x \| \leq R\}$, have the following inequality

$$R(h) \leq \hat{R}_\rho(h) + \mathrm{O}\left(\sqrt{\frac{(R^2/\rho^2)\log^2 m + \log(1/\delta)}{m}}\right)$$

where $R(h)$ is the true error of the classifier $h$, $|X| = m$ and $\hat{R}_\rho(h) = \frac{|\{x_i : y_i h(x_i) < \rho\}|}{m}$ is the ratio of margin less than $\rho$, and $\rho$ is the margin. We can find that the error dependents on the learned kernel function and sometimes the parameters determine the effectiveness of kernel methods in practical applications.

The kernel alignment can be used to measure the similarity between matrices, which was first introduced by Cristianini et al. [2]. It can be seen as the generalized cosine between matrices. When the sample set is identified, kernel alignment can measure the similarity between the kernel functions. Another definition of kernel alignment based on the feature space was given by Cortes et al. [3]. And some kernel functions were learned by the centered alignment. Some multi-class problems and regression problems can be solved by the SVMs based on alignment [4]. The kernel matrix plays a very important role in kernel methods. It contains all the topology information of the sample set. For the binary classification problems, the Gram matrix composed by the class labels vector is the optimal objective for learning the kernel function and the corresponding parameters. Then we can take the alignment as the measure to learn the kernel and parameters.

Vapnik et al. introduced the kernel function into SVMs [5], and which achieved remarkable effect. From then on the kernel method has become a mainstream algorithm in machine learning. Schölkopf et al. proposed the KPCA method [6]. Mika et al. take the kernel method into the linear Fisher discriminant criteria, and provided the nonlinear Fisher Discriminant (Kernel Fisher Discriminant,

KFD) [7]. In addition, there is Kernel Independent Component Analysis (KICA) [8] and kernel-based clustering method and so on [9-11]. In kernel methods, lots of parameters need to be determined. So far, a large part of parameters need to be set by manually, which causes a waste of time and manpower. At the same time, unreasonable parameter settings may lead to the failure of the algorithm. Some selection methods are summarized by Xu Yong et al. [12]. Bruzzone and Prieto selected the bandwidth of Gaussian function by solving the optimal problem based on the Fisher criterion [13]. Kernel methods as the non-linear techniques can be used for image processing, which are effective in image de-noising and visualization of high dimensional data [14-15].

Alignment can be used to measure the similarity between the kernel matrix and gram matrix. Using the property, take the alignment as the objective function to select the kernel and parameters. In the paper, we give the rigorous theoretical analysis of the selecting parameter method. And provide an algorithm to calculate the parameter in Gaussian kernels. Some experiments results are supplied to indicate the rationality of the algorithm.

## 2. The Kernel PCA

For the sample set $S = \{x_1, x_2, \cdots, x_N\} \subset R^n$, we denote the samples' label as $y = [y_1, y_2, \cdots, y_N]'$, where $y_i \in \{-1, +1\}, (i = 1, 2, \cdots, N)$ for the binary classification problem. Assume the samples obey the same probability distribution $D$. And the kernels $k_1$ and $k_2$ are positive semi-definite and symmetrical (PDS). The alignment can be written as [2]

$$\hat{A}_s(K_1, K_2) = \frac{< K_1, K_2 >_F}{\sqrt{< K_1, K_1 >_F < K_2, K_2 >_F}} \tag{1}$$

where $K_1, K_2 \in R^{N \times N}$ denote the kernel matrices. And the inner production between matrices is defined by

$$< K_1, K_2 >_F = \sum_{i,j=1}^{N} k_1(x_i, x_j) k_2(x_i, x_j) = trace(K_1^{'} K_2)$$

It is easy to find that alignment $\hat{A}_S(K_1, K_2) \in [-1, 1]$, especially $\hat{A}_S(K_1, K_2) \in [0, 1]$, $K_1, K_2 \geq 0$.

In the kernel methods, kernel matrix contains all the information from the samples. And the kernel matrix is determined by the sample set and the kernel function. In classification problems, the similarity between the kernel matrix and the Gram matrix of the samples' labels can be used as the standard for the selection of kernel function and the corresponding parameters. The alignment can be given by

$$\hat{A}_S(K, yy') = \frac{< K, yy' >_F}{\sqrt{< K, K >_F < yy', yy' >_F}} = \frac{< K, yy' >_F}{N \sqrt{< K, K >_F}} \tag{2}$$

The bigger of the alignment value means that the learned kernel is more appropriate to solve the problem.

## 3. Calculating the Pre-image Algorithm

In this section, we will learn the classifier function from the space $H = \{uk_1 + \sqrt{1-u^2} k_2 : u \in [0, 1)\}$, where the two kernels $k_1$ and $k_2$ is the PDS, and they are the Gaussian function with the undefined bandwidths $\sigma_1$ and $\sigma_2$. Denote the classifier function as $k = uk_1 + \sqrt{1-u^2} k_2$, which has the equivalent form with $k = u_1 k_1 + u_2 k_2$, where $u_1^2 + u_2^2 = 1$ and $u_1 \in [0, 1)$. This means that the kernels $k_1$ and $k_2$ has the symmetry position. Then we can assign the parameter $\sigma_1$ as a small fixed value. The parameter $\sigma_2 \in (\sigma_1, +\infty)$ is unknown and need to be selected for

the Gaussian kernel. Then the function in space $H$ only includes two variables $u \in [0,1)$ and $\sigma_2 \in (\sigma_1, +\infty)$. For the kernel $k$ has the following Theorem.

**Theorem 1.** Let $S = \{x_1, x_2, \cdots, x_N\} \subset R^n$ denote the sample set. Vector $y = [y_1, y_2, \cdots, y_N]' \in R^N$ is the label corresponding to the set $S$. Kernels $k_1$ and $k_2$ are the Gaussian function respectively corresponding to the bandwidths $\sigma_1$ and $\sigma_2$. Denote the kernel function as $k = uk_1 + (1-u)k_2$, $u \in [0,1)$. Then the function

$$F(u, \sigma_2) = \hat{A}_S(K, yy') = \frac{< K, yy' >_F}{N\sqrt{< K, K >_F}} \tag{3}$$

at most has two stagnation points. The matrix $K = uK_1 + (1-u)K_2$ is symmetrical and positive semi-definite. □

**Prof.** Equation (3) can write as

$$F(u, \sigma_2) = \frac{y'(uK_1 + (1-u)K_2)y}{N\sqrt{tr\left((uK_1 + (1-u)K_2)^2\right)}} \tag{4}$$

Calculate the derivative function as follows

$$\begin{aligned}
&\frac{\partial F(u)}{\partial u} \\
&= \frac{1}{N\left(tr\left((uK_1 + (1-u)K_2)^2\right)\right)^{3/2}} \cdot \left[\left(y'\left(K_1 - K_2\right)y\right) \cdot tr\left((uK_1 + (1-u)K_2)^2\right)\right. \\
&\quad \left. - \left(y'\left(uK_1 + (1-u)K_2\right)y\right) \cdot tr\left(\left(K_1 - K_2\right)\left(uK_1 - (1-u)K_2\right)\right)\right. \\
&= \frac{1}{N\left(tr\left((uK_1 + (1-u)K_2)^2\right)\right)^{3/2}} \cdot [(y'K_1y) \cdot tr(KK_2) - (y'K_2y) \cdot tr(KK_1)] \\
&= \frac{1}{N\left(tr\left((uK_1 + (1-u)K_2)^2\right)\right)^{3/2}} \cdot G(u)
\end{aligned}$$

The stagnation point $u^*$ of function $F(u)$ can be obtained by solving the equation $G(u) = 0$.

$$u^* = \frac{\left(y'K_1y\right) < K_2, K_2 >_F - \left(y'K_2y\right) < K_1, K_2 >_F}{\left(y'K_1y\right) < K_2, K_1 - K_2 >_F - \left(y'K_2y\right) < K_1, K_1 - K_2 >_F} \tag{5}$$

When $\left(y'K_1y\right) < K_2, K_1 - K_2 >_F - \left(y'K_2y\right) < K_1, K_1 - K_2 >_F = 0$, we have

$$\left(y'K_1y\right) < K_1, K_2 >_F = \left(y'K_2y\right) < K_1, K_1 >_F \tag{6}$$

$$\left(y'K_1y\right) < K_2, K_2 >_F = \left(y'K_2y\right) < K_1, K_2 >_F \tag{7}$$

From equations (6) and (7),

$$(< K_1, K_2 >_F)^2 = < K_1, K_1 >_F < K_2, K_2 >_F \tag{8}$$

Using the Cauchy inequality, the equation (8) means $K_1 = K_2$.

For the function

$$F(u, \sigma_2) = \frac{\sum_{i,j=1}^{N} y_i y_j (u \exp(-\| x_i - x_j \|^2 / 2\sigma_1^2) + (1-u)\exp(-\| x_i - x_j \|^2 / 2\sigma_2^2))}{N\sqrt{\sum_{i,j=1}^{N} (u \exp(-\| x_i - x_j \|^2 / 2\sigma_1^2) + (1-u)\exp(-\| x_i - x_j \|^2 / 2\sigma_2^2))^2}} \tag{9}$$

We have

$$\frac{\partial}{\partial\sigma_2}F(u,\sigma_2)$$
$$= \frac{1-u}{N\sigma_2^3\left(< K,K >_F\right)^{3/2}} \cdot \left[ < K,K >_F \cdot \sum_{i,j=1}^{N} y_i y_j \exp(-\| x_i - x_j \|^2 /(2\sigma_2^2)) \| x_i - x_j \|^2 \right.$$
$$- (y'Ky) \cdot \sum_{i,j=1}^{N} \exp(-\| x_i - x_j \|^2 /2\sigma_2^2) \cdot \| x_i - x_j \|^2 \cdot$$
$$\left. \left( u\exp(-\| x_i - x_j \|^2 /2\sigma_1^2) + (1-u)\exp(-\| x_i - x_j \|^2 /2\sigma_2^2) \right) \right]$$
$$= \frac{1-u}{N\sigma_2^3\left(< K,K >_F\right)^{3/2}} \cdot R(\sigma_2)$$

For the equation (10)

$$R(\sigma_2) = 0, \quad \sigma \in (0,+\infty) \tag{10}$$

at most have two roots. The detail analysis can be found in the next section.

We can find that the equation $G(u) = 0$ have following two solutions in the proof.

$$\begin{cases} u = 0,1 & k_1 = k_2 \\ u = u^* & k_1 \neq k_2 \end{cases} \tag{11}$$

**Theorem 2.** Under the same conditions in theorem 1, the alignment function $F(u,\sigma_2)$ also at most have two stagnation points for the kernel function $k = uk_1 + \sqrt{1-u^2}k_2, u \in [0,1)$. □

The proof of theorem 2 is similar to the theorem 1.

For the Gaussian kernel, the bigger bandwidth we use, the flatter function will be. Based on this property, we know that the function with smaller bandwidth will be a better measure for the samples with dramatic changes, and the function with bigger bandwidth is a good measure for the samples with slowly changes.

In practical application, the optimal bandwidth usually distributes in a small range. There is no worth for the values which beyond the range. We can set a range of bandwidth $\sigma$ in advance, and the selecting algorithm can just work on this range. The method of calculating the bandwidth $\sigma$ will be presented in the next section. In the next section we will give an algorithm to solve the equation (10).

## 4. Experiments

To verify the learning kernel functions' generalization capability, we do some numerical experiments on two data sets. The first data set is from some printed matters. We take the digits "8" and "9" as the experimental target, because they are more difficult to identify than the other digits. In Figure 5, some training samples are shown. The size of each image is $22 \times 33$. In Figure 1, shows the alignment values

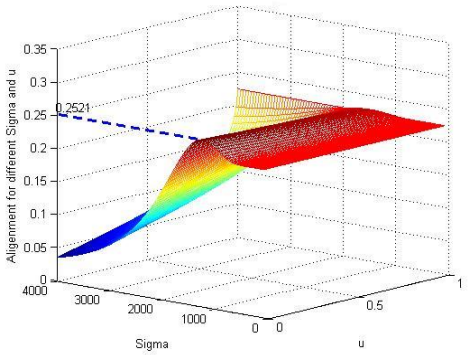


**Figure 1.** Alignment values vary with the parameters $u$ and $\sigma_2$

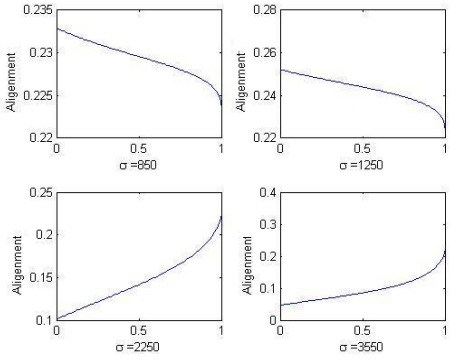


**Figure 2.** Alignment values vary with the parameters $u$

vary with the parameters $u$ and $\sigma_2$. We can find that the alignment function has a maximize value on the first data set. In Figure 2, shows the alignment values varying with the parameter $u$ at some fixed bandwidths $\sigma$. In Figure 3 and Figure 4 shows the alignment values and the recognition rates varying with bandwidth $\sigma$. The results clearly indicate that learning the kernel using the alignment is reasonable. In the experiments, we take the constant $\sigma_1 = 50$ and the vector $\sigma_2 = 100:50:4e3$.

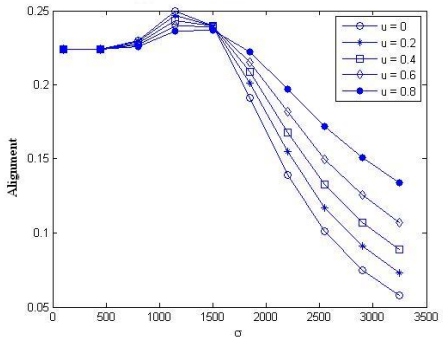


**Figure 3.** Alignment values vary with bandwidth $\sigma$

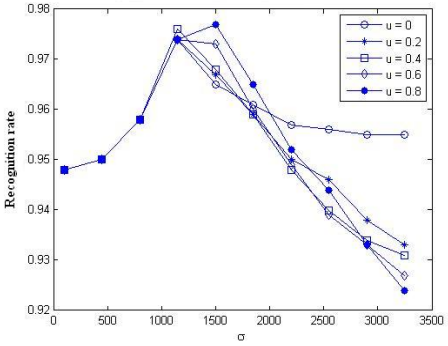


**Figure 4.** Recognition rates vary with bandwidth $\sigma$

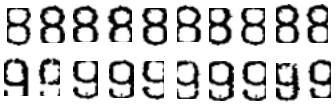

**Figure 5.** Training samples being used in the experiment

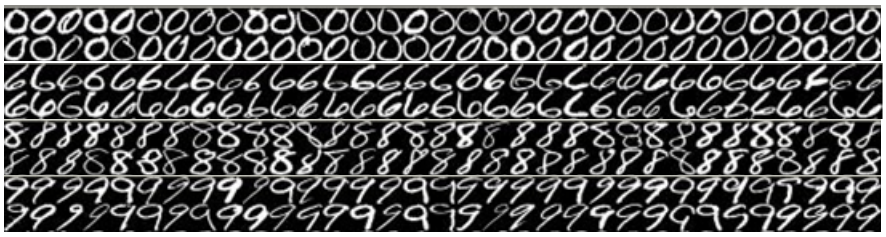

**Figure 6.** Training samples being used in the experiment

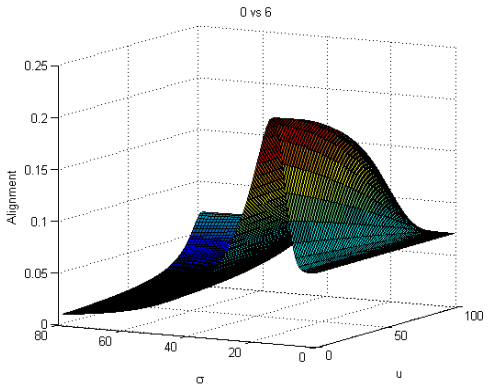


**Figure 7.** Alignment values vary with the parameters $u$ and $\sigma_2$

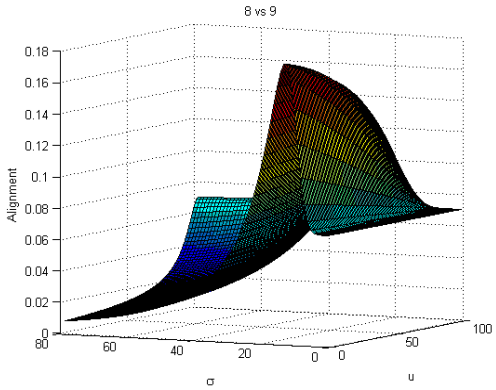


**Figure 8.** Alignment values vary with the parameters $u$ and $\sigma_2$

And then we do the same experiments on the USPS Handwritten Digits. We take the digits “0”, “6”,“8” and “9” as the experimental target. In the experiment, all the calculation is based on their gray

value. At first, we select 200 images as the training samples, which include 100 images for digital "8" and 100 images for digital "9". We take the labels of the digits "8" and "9" as 1 and -1. The test set includes 2000 images, 1000 images for digital "8" and 1000 for digital "9". Denote the set constituted by the selected 200 images as S. In Figure 6 shows some samples in the training sample set.In Figure 7 and Figure 8 show the alignment values change accompanied by the variables $\sigma_2$ and $u$. We can find that their curves are similar.

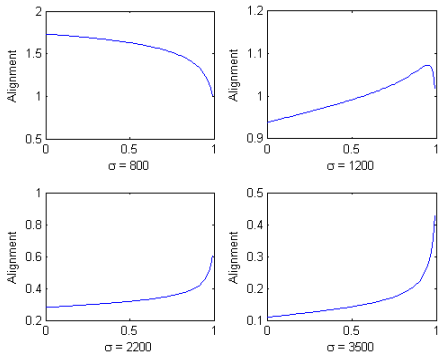


**Figure 9.** Alignment values vary with parameter $u$

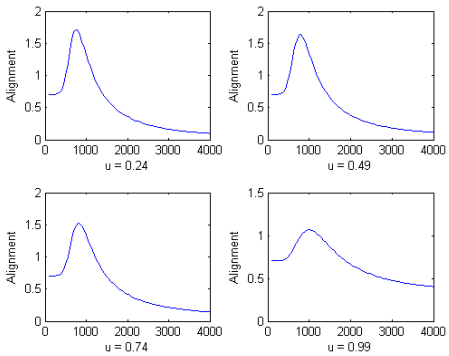


**Figure 10.** Alignment values vary with parameter $\sigma_2$

And in Figure 9, shows the alignment values change accompanied by the parameter $u$ when the bandwidth $\sigma_2$ is fixed. We can obviously find that alignment function have only one maximum point about the parameter $u$. In Figure 10, shows the alignment values change accompanied by the parameter $\sigma_2$ when the bandwidth $u$ is fixed. We also can obviously find that alignment function have only one maximum point about the parameter $u$.

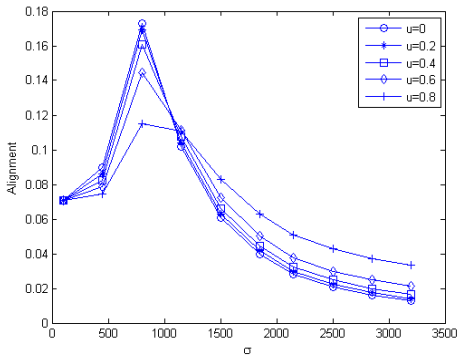


**Figure 11.** Alignment values vary with bandwidth $\sigma$

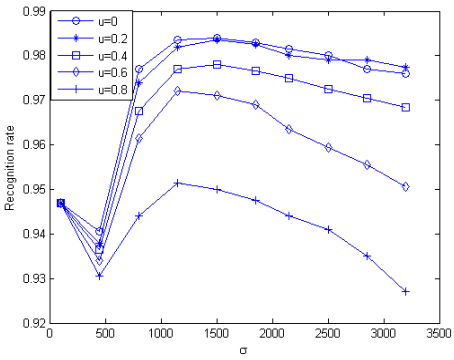


**Figure 12.** Recognition rates vary with bandwidth $\sigma$

In Table 1 and Table 2 show the alignment values and recognition rates with different parameters $u$ and bandwidths $\sigma$. From the data we can see that the recognition rate's variation is approximately consistent with the alignment. The results of numerical experiments verify the rationality to select parameters by the alignment.

From the Figure 11 and Figure 12 we can find that they have the same trends with the bandwidth $\sigma_2$. From the Figure 13 and Fig 14 we can find that parts changing trends are similar to the parameter $u$. And in Table 1 and Table 2, we provide more detailed experimental results.

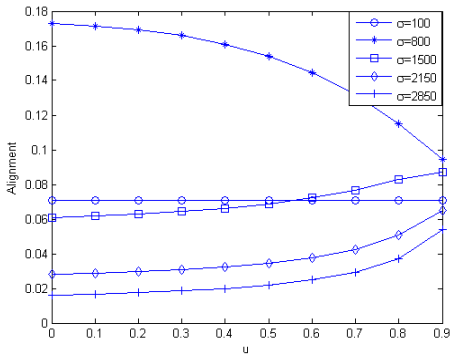


**Figure 13.** Alignment values vary with bandwidth $\sigma$

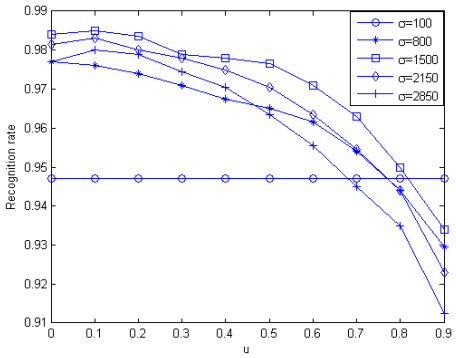


**Figure 14.** Recognition rates vary with bandwidth $\sigma$

**Table 1.** Calculated alignment values with different parameters $u$ and bandwidth $\sigma$

| $u$ \ $\sigma$ | 100 | 450 | 800 | 1150 | 1500 | 1850 | 2200 | 2550 | 2900 | 3250 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.947 | 0.090 | 0.173 | 0.102 | 0.061 | 0.040 | 0.028 | 0.021 | 0.016 | 0.013 |
| 0.1 | 0.947 | 0.088 | 0.171 | 0.103 | 0.062 | 0.041 | 0.029 | 0.022 | 0.017 | 0.013 |
| 0.2 | 0.947 | 0.086 | 0.169 | 0.104 | 0.063 | 0.042 | 0.030 | 0.022 | 0.018 | 0.014 |
| 0.3 | 0.947 | 0.084 | 0.166 | 0.106 | 0.064 | 0.043 | 0.031 | 0.024 | 0.019 | 0.015 |
| 0.4 | 0.947 | 0.082 | 0.161 | 0.108 | 0.066 | 0.045 | 0.032 | 0.025 | 0.020 | 0.017 |
| 0.5 | 0.947 | 0.081 | 0.154 | 0.110 | 0.069 | 0.047 | 0.035 | 0.027 | 0.022 | 0.018 |
| 0.6 | 0.947 | 0.079 | 0.145 | 0.112 | 0.072 | 0.050 | 0.038 | 0.030 | 0.025 | 0.021 |
| 0.7 | 0.947 | 0.077 | 0.132 | 0.113 | 0.077 | 0.055 | 0.043 | 0.035 | 0.029 | 0.026 |
| 0.8 | 0.947 | 0.075 | 0.115 | 0.111 | 0.083 | 0.063 | 0.051 | 0.043 | 0.037 | 0.034 |
| 0.9 | 0.947 | 0.073 | 0.095 | 0.099 | 0.087 | 0.074 | 0.065 | 0.059 | 0.054 | 0.051 |

**Table 2.** Calculated recognition rates with different parameters $u$ and bandwidth $\sigma$

| $u$ \ $\sigma$ | 100 | 450 | 800 | 1150 | 1500 | 1850 | 2200 | 2550 | 2900 | 3250 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.071 | 0.941 | 0.977 | 0.984 | 0.984 | 0.983 | 0.982 | 0.980 | 0.977 | 0.976 |
| 0.1 | 0.071 | 0.940 | 0.976 | 0.984 | 0.985 | 0.984 | 0.983 | 0.982 | 0.980 | 0.981 |
| 0.2 | 0.071 | 0.938 | 0.974 | 0.982 | 0.984 | 0.983 | 0.980 | 0.979 | 0.979 | 0.978 |
| 0.3 | 0.947 | 0.937 | 0.971 | 0.980 | 0.979 | 0.979 | 0.978 | 0.976 | 0.975 | 0.973 |
| 0.4 | 0.071 | 0.937 | 0.968 | 0.977 | 0.978 | 0.977 | 0.975 | 0.973 | 0.971 | 0.969 |
| 0.5 | 0.071 | 0.936 | 0.965 | 0.975 | 0.977 | 0.974 | 0.971 | 0.969 | 0.964 | 0.960 |
| 0.6 | 0.071 | 0.934 | 0.962 | 0.972 | 0.971 | 0.969 | 0.964 | 0.960 | 0.956 | 0.951 |
| 0.7 | 0.071 | 0.932 | 0.954 | 0.964 | 0.963 | 0.958 | 0.955 | 0.949 | 0.945 | 0.942 |
| 0.8 | 0.071 | 0.931 | 0.944 | 0.952 | 0.950 | 0.948 | 0.944 | 0.941 | 0.935 | 0.927 |
| 0.9 | 0.071 | 0.930 | 0.930 | 0.935 | 0.934 | 0.929 | 0.923 | 0.918 | 0.913 | 0.910 |

From the theoretical analysis and the experimental results, we can assert that it is reasonable and effective for learning the kernel and selecting parameters by the alignment in the binary classification problem.

## 5. Conclusion

We present a method of learning kernel with the alignment, including an efficient iterative algorithm. Our algorithm suggests that a higher recognition rate can be obtained based on the learning function in binary classification. The method not only can be used to select the parameters, but also can be used to learn the kernel. The sensible of selection theory is also corroborated by some of our empirical results.

## 6. References

[1] Bartlett P and Shawe-Taylor J, "Generalization performance of support vector machines and other pattern classifiers", MIT Press, USA, 1999.
[2] Cristianini N, Shawe-Taylor J, Elisseeff A, Jaz Kandola, "On kernel-target alignment ", In Proceedings of the Conference on Neural Information Processing Systems. Cambridge, pp. 367–373, 2002.
[3] Cortes C, Mohri M, Roatamizadeh A, "Two-stage learning kernel algorithms", In Proceedings of the 27th International Conference on Machine Learning , pp. 239-246, 2010.
[4] Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, Tatsuya Akutsu, "Protein homology detection using string alignment kernels", Bioinformatics, vol.20, no. 11, pp. 1682-1689, 2004.
[5] Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers", In Proceedings of 5th ACM Workshop on Computational Learning Theory, Pittsburgh, pp. 144-152, 1992.
[6] Schölkopf B, Smola A, Müller K R, "Kernel principal component analysis", In Proceedings of the 7th International Conference Lausanne, pp. 583-588, 1997.
[7] Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, Klaus-Robert Müller, "Fisher discriminant analysis with kernels", In Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing IX, pp. 41-48, 1999.
[8] Bach F R and Jordan M I., "Kernel independent component analysis", The Journal of Machine Learning Research, vol. 3, pp. 1-48, 2003.
[9] Alzate C. and Suykens J. A. K., "Hierarchical kernel spectral clustering", Neural Networks, vol. 35, pp.21-30, 2012.
[10] Karl Rohe, Sourav Chatterjee, Bin Yu, "Spectral clustering and the high-dimensional stochastic blockmodel", The Annals of Statistics, vol. 39, no. 4, pp. 1878-1915, 2011.
[11] Nguyen Lu Dang Khoa and Sanjay Chawla, "Large Scale Spectral Clustering Using Approximate Commute Time Embedding", Arxiv preprint arXiv, pp. 4541, 2011.
[12] Xu Y, Zhang D P, Yang J., "Kernel method and its application in pattern recognition", National Defence Industry Press, China, 2010.
[13] L. Bruzzone, D. F. Prieto, "A Technique for the Selection of Kernel-Function Parameters in RBF Neural Networks for Classification of Remote-Sensing Images", In Proceedings of the IEEE Transactions on geoscience and remote sensing, 1999.
[14] Yu Yi, Yang Nan, Dong Bingchao, Zheng Xiaoxiang, "Neural Decoding Based on Kernel Regression", International Journal of Digital Content Technology and its Applications, vol. 6, no. 13, pp. 427-435, 2012.
[15] Weiya Shi, "The Algorithm of Nonlinear Feature Extraction for Large-scale Data set", International Journal of Information Processing and Management, vol. 3, no. 2, pp. 45-52, 2012.