SUPPORT VECTOR CLASSIFICATION OF REMOTE SENSING IMAGES USING IMPROVED SPECTRAL KERNELS

Mohd Noor Md Sap, Mojtaba Kohram

Faculty of Computer Science and Information Systems, University Technology Malaysia
mohdnoor@utm.my kmojtaba2@siswa.utm.my

Abstract: A very important task in pattern recognition is the incorporation of prior information into the learning algorithm. In support vector machines this task is performed via the kernel function. Thus for each application if the right kernel function is chosen, the amount of prior information fed into the machine is increased and thus the machine will perform with much more functionality. In the case of hyper-spectral imagery the amount of information available prior to classification is a vast amount. Current available kernels do not take full advantage of the amount of information available in these images. This paper focuses on deriving a set of kernels specific to these imagery. These kernels make use of the spectral signature available in images. Subsequently we use mixtures of these kernels to derive new and more efficient kernels for classification. Results show that these kernels do in fact improve classification accuracy and use the prior information available in imagery to a better degree.

Keywords: Support Vector Machines, Kernels, Spectral angle, Classification, Land Cover.

I. INTRODUCTION

Support vector machines [1] have been successfully applied to multi-spectral and hyper-spectral data classification [2], [3], [4]. Their ability to map data into very high dimensions without much added computational complexity and without falling victim to the Hughes phenomenon (*curse of dimensionality*) (Hughes 1968) have made them an ideal method for dealing with remote sensing data. In addition, the solution sparseness (only part of the database is used for classification) and few number of tuning parameters available have made this technique a very attractive classification method for researchers.

A setback of the SVM algorithm is that there are few techniques available for incorporating prior information into its formulation. In the case of remote sensing data this is very unfortunate due to the vast amount of prior information available in remote sensing imagery.

similarity measures in the SVM kernel. The typical SVM formulation utilizes either the Euclidean distance or the dot product of the data vectors for discrimination purposes. While these measures have proven effective in many cases, when dealing with remote sensing data they do not take into account the spectral signatures of different land cover classes.

In this paper two spectral similarity measures are incorporated into the SVM kernel and classification results on benchmark data are compared with the standard Euclidean distance based classification. These measures are the Spectral Angle Mapper (SAM) which essentially results in the angle between two data vectors and the Spectral Information Divergence (SID) which takes into account the probability distributions produced by the spectral signatures.

This paper is organized as follows: first we take a brief look at the SVM formulation, the Kernel trick and non-linear SVM formulization. Subsequently a brief introduction to the SAM and SID similarity measures is presented. Experiment design and results are described in the next section followed by a discussion and conclusion.

2. SUPPORT VECTOR MACHINES

This section is dedicated to a brief introduction of SVMs and their classification process. We will start with basic binary and linear classification and move on to the methods adopted for removing these restrictions. For further discussion and a deeper understanding, readers are referred to [1], [5].

Assume that the training set consists of N data points $\mathbf{x}_i$ where $\mathbf{x}_i \in \Re^d$, (i = 1, 2, ..., N). To each vector we assign a target $y_i \in \{-1,1\}$. Our task is to locate a hyper-plane $\mathbf{w} \cdot \mathbf{x}^T + b$ which correctly divides these data points into two classes, resulting in the decision function $f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x}^T + b)$ which in turn correctly classifies the data. In effect, there exist an infinite number of hyper-planes which correctly divide the data points. Among these, SVMs select the hyper-plane which *maximizes the margin*. The margin is defined as the distance between the classifier and the closest training points. The *maximal margin hyper-plane* is identified by solving the following convex optimization problem with respect to w, ξ_i, and b:

$$\textit{Minimize}: \qquad (1)$$
$$\|\mathbf{w}\| + C\sum_{i=1}^{N}\xi_i$$
$$\textit{Subject to}:$$
$$y_i(\mathbf{w}\cdot\mathbf{x}_i^T + b) \geq 1-\xi_i$$
$$\xi_i \geq 0$$
$$i = 1,\ldots,N$$

The slack variables ξ_i are added to the formulation to allow for misclassifications when the two classes are inseparable. The above formulation maximizes the margin $(1/\|\mathbf{w}\|)$ while minimizing the errors made during classification by placing a penalty term, C, on the summation of errors made by the classifier. Figure 1 illustrates the classification of non-linearly separable data.

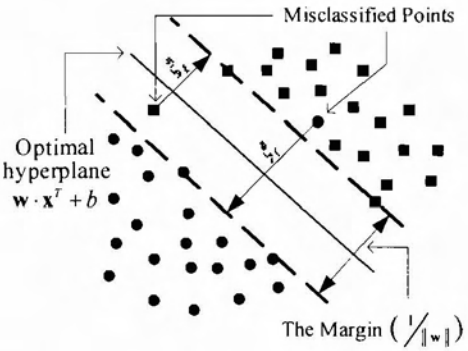

Fig. 1. Maximal Margin Classifier in linearly non-separable case

This optimization problem is solved by transforming it to its corresponding dual problem. Following this adaptation, problem (1) is converted to the following optimization problem:

$$\textit{Minimize}: \qquad (2)$$
$$W(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j^T)$$
$$\textit{Subject to}:$$
$$\sum_{i=1}^{N} y_i \alpha_i = 0$$
$$0 \leq \alpha_i \leq C, i = 1,\ldots,N$$

whereby α_i represent the lagrange multipliers and are bounded by the penalty parameter C, hence resulting in the constraints occasionally being called the *box constraints*. This formulization results in a decision function of the form:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}^T) + b^*\right) \quad (3)$$

It can be seen that only data vectors with $\alpha_i \neq 0$ play a role in finding the optimal hyper-plane. Usually most of the α_i in the optimal solution are equal to 0, and this results in a huge amount of data not being responsible for classification. Only a select group of vectors called the support vectors contribute to the solution. This feature (sparseness) of SVMs is very valuable when dealing with large datasets like hyper-spectral and multispectral images.

While the above formulation is designed for binary classification, various techniques have been devised to expand this to multi-class classification. The two most popular methods are the *one against all* and the *one against one*. The one against all technique forms a binary SVM for each class against all the other classes while the one against one method forms a binary SVM for each pair of classes. Subsequently for each testing point the most often computed label is assigned to the particular vector.

3. KERNELS AND NONLINEAR MACHINES

The problem dealt with up to now has been to linearly classify the data, but for nonlinear classifications SVMs make use of kernel methods. These methods consist of mapping data into a high dimensional feature space $\Phi(\mathbf{x})$ where the data can be smoothly classified by a linear machine (figure 2).

Taking note of equations (2) and (3), it is observed that wherever the data points appear they are in form of dot products, thus it is convenient to define a function K as:

$$K(\mathbf{x}, \mathbf{x}_i) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i) \quad (4)$$

This function is called the kernel function and results in the dot product of the training vectors in feature space. In this manner, instead of taking on the computationally expensive task of mapping all the data points into feature space, the kernel matrix can be computed with much more efficiency and speed. This results in an efficient non-linear classification of the data. By replacing the data points with the kernel function in (2), the following formulation is achieved:

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(\mathbf{x}, \mathbf{x}_i) \tag{5}$$

while the decision function can be expressed as:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}_j) + b^* \right) \tag{6}$$

For a function to be categorized as a kernel that function has to fulfill Mercer's conditions [1]. These conditions are equivalent to requiring that "for any finite subset of input space $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, the corresponding matrix is positive semi-definite" [5].

4. SPECTRAL SIMILARITY MEASURES

While the kernels introduced in table 1 have proven successful in most cases, when dealing with hyperspectral data, they do not take full advantage of the rich amount of *a priori* information available. This could be due to the fact that these Kernels do not take into account the band to band spectral signature effects. In order to effectively make use of this information, measures of similarity other than the Euclidean distance and dot product must be used. Here we introduce two measures that are specifically designed for this purpose. It is insightful to add that while one of these measures is geometrical and another is entropy-based, both these measures are insensitive to the length of the pixel vectors whose similarity they measure.

4.1 The Spectral Angle Mapper (SAM)

SAM has been abundantly used for remote sensing applications. It measures the similarity of two vectors by finding the angle between two data vectors. The SAM function is defined as:

$$\alpha(\mathbf{x}, \mathbf{x}_i) = arc\cos\left(\frac{\mathbf{x} \cdot \mathbf{x}_i^T}{\|\mathbf{x}\| \|\mathbf{x}_i\|} \right) \tag{7}$$

SAM has the ability to exploit spectral characteristics while being robust to energy differences because the angle between two vectors is not affected by the length of the particular vector [6]. However, if the angle between two data vectors is very small, SAM tends towards the Euclidean distance and the two measures achieve very similar discrimination results [7].

4.2 Spectral Information Divergence (SID)

Spectral information divergence [8] is essentially an application of the *Kullback-Lieber* (KL) divergence (Kullback-Lieber distance, relative entropy) to hyper-spectral data. The KL divergence is an information theoretic based measure that is defined as the "distance" of one probability distribution from another. It is part of a broader class of divergences called f-divergences [9] utilized to quantify "closeness" of one probability distribution from another.

For two discrete probability distributions P and Q, the f-divergence can be written as:

$$I_f(P,Q) = \sum_i q(x_i) f\left(\frac{p(x_i)}{q(x_i)}\right) \tag{8}$$

By substituting $f(t) = t \ln t$ into the above, the KL divergence is achieved:

$$D(P \| Q) = \sum_i p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right) \tag{9}$$

It can be proven that the inequality $D(P \| Q) \geq 0$ always holds and that $D(P \| Q) = 0$ if and only if $P = Q$. It can also be seen that in general the KL divergence is not symmetric, that is, in general $D(P \| Q) \neq D(Q \| P)$. This can easily be overcome by defining:

$$I(P,Q) = D(P \| Q) + D(Q \| P) \tag{10}$$

as the distance between distributions P and Q. Loosely speaking in information theoretic terms, this equation is equivalent to the amount of *uncertainty* resulting from evaluation of the *entropy* of P by use of distribution Q thus quantifying the distance between P and Q.

To employ the KL divergence as a measure of distance between two data points, a probabilistic interpretation of the data points has to be adopted. This can be achieved by viewing the spectral signature of each pixel vector as a probability distribution and subsequently using the KL divergence to measure the distance of these distributions. Each pixel vector's probability distribution can be derived by normalizing the pixel vector to unity.

For a vector $\mathbf{x} = [x_1 \ldots\ldots x_l]^t$ this is accomplished as follows:

$$p_i = \frac{x_i}{\sum_{i=1}^{l} x_i} \tag{11}$$

The above formulation results in a probability distribution vector $\mathbf{p} = [\, p_1, ..., p_d \,]^T$ for each pixel, which from here on describes the properties of the pixel vectors. This adjustment is significant since it opens many probabilistic and statistical routes for processing the data. As an example it can be seen that for each pixel, statistics of different order such as mean, variance, third and fourth order statistics can be defined and utilized in an appropriate manner [7]. Using this metric the spectral information divergence of two pixel vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ with probability vectors **p** and **q** respectively, is defined as:

$$SID(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_i \| \mathbf{x}_j) + D(\mathbf{x}_j \| \mathbf{x}_i) \tag{12}$$

Where:

$$D(\mathbf{x}_i \| \mathbf{x}_j) = \sum_{l=1}^{d} p_l \log(p_l / q_l) \tag{13}$$

To illustrate how SID takes into account spectral signatures, we can consider the extreme case of the distance between pixel vector **x** and its probability distribution vector **p**. Since both these vectors have the same spectral signature, we are hoping that the distance between them is minimal using every metric. Since SAM is also length insensitive, it does achieve the desired results, but this is not the case if Euclidean distance is applied to these two vectors. Using the Euclidean distance, these two vectors might readily be cast very far from each other.

To sum up, SID does not take into account the geometrical features of the pixel vectors; instead it takes note of the discrepancy between the probability distributions produced by each pixel vector. It is hoped that this unique feature of SID will render this measure more effective than the geometrical measures during classification.

5. MIXTURE KERNELS

By embedding the above similarity measures into an RBF function, these measures can be effectively used as the kernel function for the SVM algorithm. The three similarity measures applied in this paper are the Euclidean distance, spectral angle mapper and the spectral information divergence. The insertion of these measures into the RBF function results in the following kernels:

1) Euclidean distance based Gaussian RBF (basic RBF):

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2) \quad (14)$$

2) SAM based RBF [10], [11], [12], [13]:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma |\alpha(\mathbf{x}, \mathbf{x}_i)|) \quad (15)$$

3) SID based RBF [10], [14], [15]:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma SID(\mathbf{x}, \mathbf{x}_i)) \quad (16)$$

While the above kernels have provided individually impressive results, it has been shown that each kernel reveals individual properties that are unique to that kernel [15]. In this view, a mixture of these functions which takes advantage of the diversity and unique qualities of each kernel could prove productive. Mixtures or composites can be achieved by simple or weighted addition of the aforementioned kernels. It must be noted that all simple and weighted additions of two or more mercer kernels result in an eligible mercer kernel [16]. Composites of classical functions as in the polynomial and RBF kernels have shown acceptable results in application [17], [18] and hence it is reasonable to perceive that a mixture of functions with higher classification accuracy than the classical kernels will result in an improved and more accurate classifier.

In this paper, using the above kernels, four mixtures are produced and the results of these mixtures are compared from various perspectives. The formed kernels are derived by simple addition of the functions above. Thus for example the SID-SAM mixed kernel would simply be:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma_1 SID(\mathbf{x}, \mathbf{x}_i)) + \exp(-\gamma_2 |\alpha(\mathbf{x}, \mathbf{x}_i)|) \quad (17)$$

In this same manner the other three kernels are the SAM-RBF, SID-RBF, and the SAM-SID-RBF kernels. For ease of naming convention, and since the RBF is usually used with the Euclidean distance, the word RBF is used instead of Euclidean in the naming of the functions.

Also, note that the fourth function is a threefold mixture of all the above kernels. While the overhead of such a function might not be worth the improvement in accuracy, it could be useful to take a look at the results achieved from this kernel as opposed to the results from the two fold mixtures. Experimentation and results on these kernels are provided in the next section.

6. EXPERIMENTATION AND RESULTS

Experiments were applied to the well known hyper-spectral AVIRIS image taken over Northwest Indiana's Indian Pine in June 1992 [19]. This image consists of 145 × 145 pixels in 220 spectral bands.

Figure 2 shows band 100 along with the land cover map of the image. The image consists of sixteen classes of which seven were discarded due to insufficient data. The remaining nine classes were used for classification purposes.

Before any testing could begin the parameter tuning had to be performed on the various kernels to pinpoint exact values for C and γ of each kernel. This was achieved through 10-fold cross validation of a randomly selected training set consisting of 50 data points from each class. A useful observation made in this phase was that simultaneous tuning of the parameters essentially reaches the same results as tuning each kernel separately and using the obtained parameter in the mixture kernel. Utilization of this practical remark leads to a two dimensional grid search at all times instead of dealing with a time consuming three or four dimensional grid search in the parameter tuning phase.

Fig. 2. (*right*) band 100 of the AVIRIS image in grayscale, (*left*) thematic map of the AVIRIS image.

For the first phase of testing, 10 independently selected training and testing sets, each dividing the data into two equal parts, were acquired from the data and the seven kernels were tested on these ten sets. Table 1 depicts classification results for the first five sets while table 2 shows cumulative results for all data sets.

Table 1 – Comparative results of the various kernels on the first five data sets. In each set the most accurate classifier is set in bold.

	Data Set									
	1		2		3		4		5	
Kernel	Acc	nSV	Acc	nSV	Acc	nSV	Acc	nSV	Acc	nSV
RBF	94.26	1584	94.58	1578	94.09	1588	94.64	1584	94.77	1631
SAM	94.07	2476	93.98	2482	94.04	2514	94.30	2496	94.58	2546
SID	95.07	1860	94.88	1866	**95.16**	1900	94.82	1882	95.61	1886
RBF-SAM	95.012	1757	**95.35**	1780	94.77	1788	95.16	1765	95.29	1811
RBF-SID	95.33	1717	95.20	1707	94.82	1741	95.07	1740	95.65	1750
SAM-SID	95.11	1921	94.97	1929	95.05	1962	94.78	1934	95.63	1943
RBF-SAM-SID	**95.37**	1764	95.24	1737	94.90	1766	**95.18**	1769	**95.69**	1788

The first column of table 1 in each set depicts the accuracy of the specific kernel on that data set and the second column is the number of support vectors used for classification with this kernel. From table 1 and table 2 it can be clearly seen that among the non-mixture kernels the SID kernel outperforms the others by a comparatively large margin. This result is in line with other studies performed on this kernel [20], [15].

However by looking at table 1 it is not clearly evident which of the two-kernel mixtures have higher accuracy. RBF-SAM and RBF-SID are each the best on two data sets while SAM-SID is the best on one of the data sets. A look at table 2 will reveal that while all three of these kernels achieve close results, the RBF-SID is the more accurate classifier and moreover, this kernel has the minimum number of support vectors among the three kernels. This leads to a very sparse solution, which is more time-efficient and also has higher generalization capability. Lastly, by looking at the RBF-SAM-SID kernel, it can be viewed that as expected this kernel outperforms all the previous kernels in terms of accuracy. In table 1 it can be seen that this kernel is first in terms of accuracy on all but two data sets (set 3). In all 10 sets it came second only three times, outperforming all the other kernels on the other 7 sets. Also in terms of number of support vectors this kernel achieved a respectable second.

Another experiment undertaken was to do a 20-fold cross validation on the whole data set (9345 pixels) to mainly confirm the above results and also to get a deeper understanding of the gap between the kernels. Results of this experiment are presented in the third column of table 2. Sure enough, the RBF-SAM-SID kernel is the strongest in this part but because the volume of the training data in each fold is very large the difference between the classifiers accuracy is small. Here again the SID shows its power by coming very close to the mixture models in terms of accuracy. The SAM-SID and the RBF-SAM show exactly the same result and again the BFSID shows stronger than these two kernels.

Table 2 – Average number of support vectors and average accuracy over 10 randomly selected training and testing sets. The third column is the result of a 20-fold cross validation experiment. The highest accuracies are emboldened.

Kernel	Avg. nSV.	Avg. Acc.	20-fold CV
RBF	1593	94.42	95.73
SAM	2503	94.20	95.53
SID	1879	94.97	96.47
RBF-SAM	1778	95.09	96.55
RBF-SID	1727	95.25	96.63
SAM-SID	1924	90.02	96.55
RBF-SAM-SID	1763	**95.31**	**96.67**

In the second phase of testing we will look at how these kernels react to variation in the number of training samples provided. For this purpose the data is first divided into 5 sets, with training sets consisting of 5,10,30,50 and 70 percent of the data respectively. The results of this experiment are shown in Figure 3.

Figure 3 illustrates that the trend of classification accuracy does not significantly change with variation in training set size. However, there is a mild change of trend in the 5 percent case. In this case for the first time in experimentation the RBF-SAMSID kernel comes 3rd in accuracy. While the two kernels occupying first and second place both have a Euclidean distance particle in them. This demonstrates the high ability of the Euclidean based kernel to generalize from very low amounts of training data.

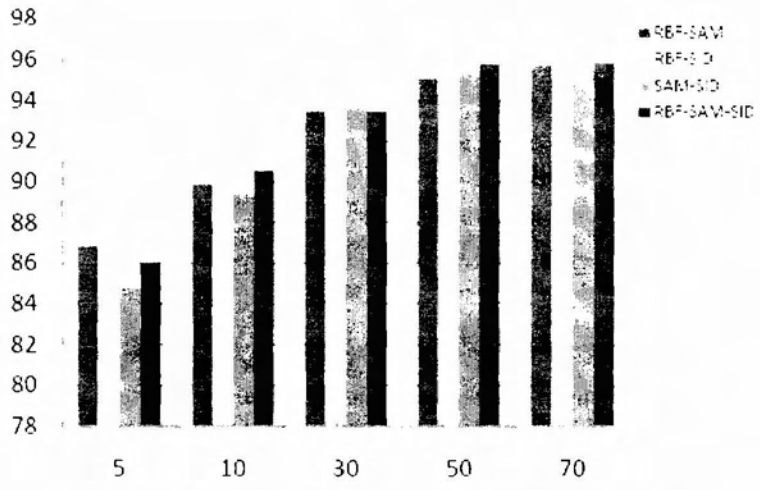

Fig. 3 - Effects of Variation of Training size on the various kernels

After this oddity, the rest of the diagram follows the trend shown before in the basic training and testing phase, meaning that the RBF-SID and RBF-SAM-SID kernels mostly dominate the other kernels.

3. Shah, C.A., Watanachaturaporn, P., Varshney, P.K., Arora, M.K.: Some recent results on hyperspectral image classification. 2003 IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data (2003)
4. Pal, M., Mather, P.M.: Assessment of the effectiveness of support vector machines for hyperspectral data. Future Generation Computer Systems **20** (2004) 1215–1225
5. Cristianini, N., Shaw-Taylor, J.: Support Vector Machines. Cambridge university Press, Cambridge, Uk (2000)
6. Kruse, F.A., Richardson, L.L., Ambrosia, V.G.: Techniques Developed for Geologic Analysis of Hyperspectral Data Applied to Near-Shore Hyperspectral Ocean Data. Fourth International Conference on Remote Sensing for Marine and Coastal Environments, Orlando, Florida (1997)
7. Chang, C.-I.: Hyperspectral Imaging: Techniques for Spectral Detection and Classification. Kluwer Academic/Plenum Publishers, New York (2003)
8. Chang, C.-I.: An information theoretic-based measure for spectral similarity and discriminability. IEEE transactions on geoscience and remote sensing **46** (2000) 1927–1932
9. Ali, S.M., Silvey, S.D.: A General Class of Coefficients of Divergence of One Distribution from Another. Journal of the Royal Statistical Society **28** (1966) 31–142
10. Mercier, G., Lennon, M.: Support Vector Machines for Hyperspectral Image Classification with Spectral-based kernels. IEEE International Geoscience and Remote Sensing Symposium (2003)
11. Fauvel, M., Chanussot, J., Benediktsson, J.A.: Evaluation of Kernels for Multiclass Classification of Hyperspectral Remote Sensing Data. IEEE International Conference on Acoustics, Speech and Signal Processing (2006)
12. Sindhumol, S., M.Wilscy: Hyperspectral Image Analysis -A Robust Algorithm using Support Vectors and Principal Components. The International Conference on Computing: Theory and Applications (2007)
13. Sap, M.N.M., Kohram, M.: Spectral Angle Based Kernels for the Classification of Hyperspectral Images Using Support Vector Machines. IEEE proceedings of the Asia Modelling Symposium 2008 (2008)

14. Nemmour, H., Chibani, Y.: Multiple support vector machines for land cover change detection: An application for mapping urban extensions. ISPRS Journal of Photogrammetry & Remote Sensing **61** (2006) 125–133

15. Kohram, M., Sap, M.N.M., Ralescu, A.L.: Spectral Information Divergence based kernels for SVM Classification of Hyper-Spectral Data. The 19th Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, OH, USA (2008)

16. Scholkopf, B., Smola, A.J.: Learning with kernels. MIT Press Cambridge, Mass (2002)

17. Smits, G.F., Jordaan, E.M.: Improved SVM Regression using Mixtures of Kernels. Proceedings of the 2002 International Joint Conference on Neural Networks, Honolulu, HI, USA (2002)

18. Jiang, T., Wang, S., Wei, R.: Support Vector Machine with Composite Kernels for Time Series Prediction. Advances in Neural Networks – ISNN 2007 (2007) 350-356

19. Landgrebe, D.: AVIRIS NW Indiana's Pines 1992 dataset. http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html (1992)