SENSITIVITY ANALYSIS OF SUPPORT VECTOR MACHINE IN CLASSIFICATION OF HYPERSPECTRAL IMAGERY

F. Samadzadegan, H. Hasani*, T. Partovi

Department of Geomatics Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran
(samadz, hasani, tpartovi)@ut.ac.ir

Commission I, WG I/2

KEY WORDS: Classification, Hyperspectral Imagery, Hughes Phenomenon, Feature Extraction, Support Vector Machine

ABSTRACT:

Nowadays by developing hyperspectral sensor technology, it is possible to simultaneously capture image with hundreds of contiguous narrow spectral bands. Increasing spectral bands provide more information and seem to improve classification accuracy. Nevertheless limited training samples lead to poor parameter estimation of statistical classifiers which is called Hughes phenomena. Recently Support Vector Machines (SVMs) are applied successfully for classification of hyperspectral imagery because they characterize classes by a geometrical criterion, not by statistical criteria. However, accuracy and performance sensitivity of SVMs in classification of hyperspectral imagery are affected by three different factors. The first one is the type of input data space which can be spectral space or feature space. In this paper three feature extraction methods, include: Principle Component Analysis (PCA), Independent Component Analysis (ICA) and Linear Discriminate Analysis (LDA) are used. Another effective factor is spectral similarity measures. Most of studies use Euclidean distance as a metric for measuring similarity between samples. By using Euclidean distance, geometric behaviour of data is evaluated and spectral meaning is not considered. This paper evaluates the effect of different metrics such as Spectral Angle Mapper (SAM) and Spectral Information Divergence (SID) on accuracy of classification. The last factor is training sample size that effect of this factor on SVMs classification accuracy is evaluated and results were compared with K-Nearest Neighbour (KNN) classifier. For evaluating sensitivity analysis of SVMs respect to these factors, polynomial and Gaussian kernels and two usual multiclass classification strategies include one against one and one against all are applied. Also experiments are carried out on the AVIRIS dataset.

1. INTRODUCTION

Hyperspectral imaging sensors are able to acquire several hundreds of spectral information from the visible to the infrared region (Chi and Bruzzone, 2007). These sensors provide very high spectral resolution image data and make it possible to discriminate among land cover classes that are spectrally very similar (Chi and Bruzzone, 2007). Nevertheless classification of hyperspectral data with conventional parametric classifiers such as maximum likelihood suffers from Hughes phenomena or *curse of dimensionality* (Hughes, 1968; Landgrebe, 2002). Referring to assumption of parametric classifier about class distribution, it is required to estimate distribution parameters. For this purpose, by increasing spectral dimension, more training data is needed. In the most application, training samples are limited, so it is not possible to estimate parameters accurately and classification accuracy decrease after increasing dimension more than a threshold (Fauvel et all, 2004)

Recently, SVMs as a non-parametric classifiers are applied successfully for classification of hyperspectral imagery (Melgani and Bruzzone, 2004; Camps and Bruzzone, 2005; Guo et all, 2008). Because they don't need to assume about class distribution and characterization of classes are based on geometrical criteria not by statistical criteria (Melgani and Bruzzone, 2004). SVMs work based on finding an optimum hyperplane that maximized the margin between two classes (Du et all, 2008). If training data are not separated linearity, a kernel method is used to project data to higher dimension space where data are separated linearly (Mercier and Lennon, 2003). For finding optimal hyperplane, it uses only support vectors which are the nearest data to hyperplane. As SVMs use small training samples (only support vectors), they are less sensitive to space dimensionality and hence it overcomes the Hughes' Phenomenon and is an effective tool in classification of hyperspectral data (Wang et all, 2008). It should be considered that SVMs are binary classifiers and can separate two classes. Classification of data with more than two classes, called multiclass classification, is frequent in remote sensing applications. In these cases, there are two usual strategies to classify data: one against all and one against one (Varshney and Arora, 2004). In both strategies, computational complexity depends on number of classes. However, SVMs are efficient in compare of other classifiers in high dimensional space but classification accuracy by SVMs strongly depends on kernel type and parameters setting (Pal and Mather, 2004).

This paper evaluates the sensitivity of SVMs in classification of hyperspectral imagery regarding to three different criteria. The type of input space is the first factor which can be original space or feature space (Cao et all, 2003; Kuo and Cheng, 2005). In original space, bands spectral values are used as an input. The advantage of using original space is using directly spectral information and also it doesn't need to feature extraction step. Feature space is obtained by feature extraction methods which transform data from original space to feature space. The advantage of using feature space is possibility of improving

* Corresponding author

classification performance in some classification techniques (Zhang and Huang, 2010). The second effective factor in classification is spectral similarity measures which consider spectral meaning, such as: SAM and SID. (Mercier and Lennon, 2003; Fauvel et all, 2006; Kohram and Sap, 2008). And the last factor is training sample size (Pal and Mather, 2004).

2. SUPPORT VECTOR MACHINES

SVMs are classification systems derived from statistical learning theory and they are kernel based methods. SVMs are binary classifiers. For two-class classification problem can be stated the following way (Varshney and Arora, 2004): N training sample are available and can be represented by the set pairs $\{(y_i, x_i), i = 1,2,...,N\}$ with y_i is a class label of value ± 1 and $x_i \in \Re^k$ feature vector with k components. The classifier is represented by the function $f(x;\alpha) \to y$ with α is the parameters of classifier. The SVMs method consist in finding the optimum separating hyperplane so that: 1) Samples with labels $y = \pm 1$ are located on each side of the hyperplane; 2) the distance of the closest samples to the hyperplane in each side become maximum. These samples are called support vectors and the distance is optimal margin (Figure 1).

The hyperplane is defined by $w.x + b = 0$ where (w, b) are the parameters of the hyperplane. The vectors that are not on this hyperplane lead to: $w.x + b \neq 0$ and the classifier is defined as: $f(x;\alpha) = \text{sgn}(w.x + b)$. The support vectors lie on two hyperplanes, which are parallel to the optimal hyperplane, have equations: $w.x + b = \pm 1$.

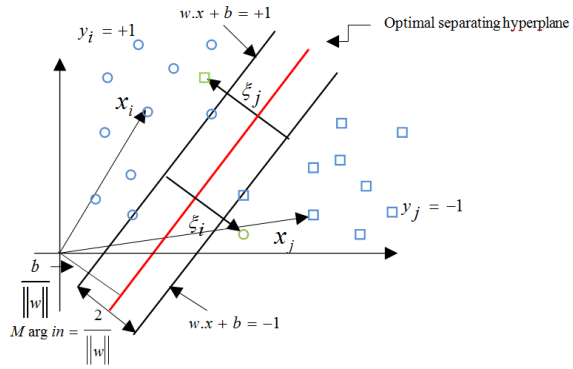


Figure 1. Classification of a non-linearly separable case by SVMs.

Sometimes, due to the noise or mixture of classes introduced during the selection of training data, variables $\xi_i > 0$, called slack variables, are used to consider effects of misclassification. Then the hyperplanes for two classes become $w.x + b = \pm(1 - \xi_i)$. Optimal hyperplane is located where the margin between two classes of interest is maximized and the error is minimized. This can be achieved by solving the following constrained optimization problem:

$$\text{Minimization: } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{k}\xi_i \qquad (1)$$

$$\text{Subject to: } y_i(w.x + b) \geq 1 - \xi_i, i = 1,...,k$$

The constant $0 < C < \infty$, called the penalty value or C value, is a regularization parameter. It defines the trade-off between the number of misclassification in the training data and the maximization of margin. In practice, the penalty value is selected by trail and error. The constrained optimization in Eq(1) is solved by the method of Lagrange multipliers. The equivalent optimization problem becomes,

$$\text{Maximize: } \sum_{i=1}^{k}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j y_i y_j (x_i.x_j)$$

Subject to: (2)

$$\sum_{i=1}^{k}\alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C, \text{ for } i = 1,2,...,k$$

In Eq(2), $\alpha_i \geq 0$ are the Lagrange multipliers. The solution of the optimization problem given in Eq(2) is obtained in terms of the Lagrange multipliers α_i. Only for support vectors, these multipliers are non-zero. The result from the optimizer, called an optimal solution, is the set $(\alpha_1^o,...,\alpha_k^o)$. The value of w and b are calculated from $w^o = \sum_{i=1}^{k} y_i\alpha_i^o x_i$ and $b^o = \frac{1}{2}\left[w^o.x_{+1}^o + w^o.x_{-1}^o\right]$ where x_{+1}^o and x_{-1}^o are the support vectors of class labels +1 and −1 respectively. The decision rule is then applied to classify the dataset into two classes.

$$f(x) = sign\left(\sum_{\text{support vector}} y_i\alpha_i^o(x_i . x) + b^o\right) \qquad (3)$$

Where $sign(\bullet)$ is the signum function. It returns +1 if the element is greater than or equal to zero and -1 if it is less than zero. There are instances where a linear hyperplane cannot separate classes without misclassification; however, those classes can be separated by a nonlinear separating hyperplane. In this case, data may be mapped to a higher dimensional space with a nonlinear transformation function. In the higher dimensional space, data are spread out, and a linear separating hyperplane may be found.

Nonlinear transformation function ϕ maps the data into a higher dimensional space. There exists a function k, called a kernel function, such that, $k(x_i, x_j) \equiv \phi(x_i)\cdot\phi(x_j)$ a kernel function is substituted for the dot product of the transformed vectors, and the explicit form of the transformation function ϕ is not necessarily known. Further, the use of the kernel function is less computationally intensive. The optimization problem then becomes,

$$Maximize: \sum_{i=1}^{k}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \qquad (4)$$

$$Subject\ to: \sum_{i=1}^{k}\alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C \quad for\ i = 1,2,...,k$$

The decision function becomes,

$$f(x) = sign\left(\sum_{\text{support vector}} y_i\alpha_i^o K(x_i, x) + b^o\right) \qquad (5)$$

A great number of kernels exist which can be divided into two categories: local and global kernels (Mercier and Lennon, 2003). In local kernels only the data that are close or in the proximity of each others have an influence on the kernel values. Basically, all kernels that are based on a distance function are local kernels. In global kernels samples that are far away from each others still have an influence on the kernel value. All kernels based on the dot-product are global.
For classification of hyperspectral images, two local and global kernels are widely used respectively are: the inhomogeneous polynomial function and the Gaussian radial basis function (Fauvel et all, 2006).

$$K_{Polynomial}(x_i, x_j) = \left[(x_i . x_j) + 1\right]^p \quad (6)$$

$$K_{Gausss}(x_i, x_j) = \exp\left[-\gamma \left\| x_i - x_j \right\|^2\right] \quad (7)$$

SVMs were originally developed to perform binary classification. However, classification of data into more than two classes, called multiclass classification, is more practical in remote sensing applications. Two usual methods are one against all and one against one. One against all is also known as *winner-take-all* classification. For an *M* class classification, *M* binary SVMs classifiers are created. Each classifier is trained to discriminate one class from the remaining *M-1* classes. During the testing or application phase, data are classified by computing the margin from the linear separating hyperplane. Data are assigned to the class labels of the SVMs classifiers that produce the maximal output. One against one in this strategy, SVMs classifiers for all possible pairs of classes are created. For an *M* class classification, *M* (*M*−1)/2 binary classifiers are created. Each binary classifier is trained to classify two classes of interest. During the testing phase, the output from each binary classifier in the form of a class label is obtained. The class label that occurs the most is assigned to that data.

3. SENSITIVITY ANALYSIS OF SVM

For sensitivity analysis of SVMs three different criteria are evaluated. The first one is the type of input data space. It can be original space or feature space. In original space, bands spectral values are used as an input. The advantage of using original space is using directly spectral information and also it doesn't need to feature extraction step. Feature space is obtained by feature extraction methods which transform data from original space to feature space. The advantage of using feature space is possibility of improving classification performance in some classification techniques. Three feature extraction methods that are used for sensitivity analysis are: PCA, ICA and LDA. Second factor is spectral similarity measures. Most of studies use Euclidean distance as a metric for measuring similarity between samples. By using Euclidean distance, geometric behaviour of data is evaluated and spectral meaning is not considered. In order to effectively make use of information intrinsically available in remote sensing imagery, other metrics can be used. For this purpose two metrics are evaluated on SVMs performance: SAM and SID. And the last factor is training sample size. For this purpose four different subset of training set is used and sensitivity of SVMs according to these training subsets are evaluated and results compare with KNN classifier result.

3.1 Feature Space

In this paper, three different feature extraction methods of PCA, ICA and LDA are evaluated:
*a) Principal Component Analysi*s: PCA is a usual unsupervised feature extraction which is based on selection features with higher variance in original space. Given a set of centered input vectors $x_t (t = 1,..., l \text{ and } \sum_{t=1}^{l} x_t = 0)$, each one has m dimension. PCA linearity transforms each vector x_t into a new one s_t by

$$s_t = U x_t^T \quad (8)$$

Where U is the $m \times m$ orthogonal matrix whose i^{th} column u_i is the i^{th} eigenvector of the sample covariance matrix

$$C = \frac{1}{l} \sum_{t=1}^{l} x_t x_t^T \quad (9)$$

In other words, PCA firstly solves the eigenvalue problem:

$$\lambda_i u_i = C u_i, \quad i = 1,..., m \quad (10)$$

Where λ_i is one of the eigenvalues of C and u_i is the corresponding eigenvector. Based on the estimated u_i the components of s_t are then calculated as the orthogonal transformation of x_t :

$$s_t = u_i x_t^T, \quad i = 1,..., m \quad (11)$$

The new components are called principal components. By using only the first several eigenvectors sorted in descending order of the eigenvalues, the number of principal components in s_t can be reduced. So PCA has the dimensional reduction characteristic (Cao et all, 2003).

b) *Independent Component Analysis*: the goal of ICA is to recover independent and unknown source signals from their linear mixtures without knowing the mixing coefficients. Let x_t and s_t denote the linear mixtures and original source signals respectively; the aim of the ICA is to estimate s_t by

$$s_t = U x_t \quad (12)$$

Where U is unmixing matrix. For estimating s_t, ICA assumes s_t components are independent statistically and all of them with possible exception of one component must be non-Gaussian. Hence it needs higher order information of the original inputs rather than the second-order information of the sample covariance as used in PCA.
A large amount of algorithms have been developed for performing ICA (Bell and Sejnowski, 1995). One of the best methods is the fixed-point-FastICA algorithm. FastICA algorithm is based on minimization of mutual information which is used as the criterion to estimate s_t as it is a natural measure of the independence between random variables. Minimization of mutual information is corresponding to maximization of negentropy which is approximated by:

$$J_G(u_i) = [E\{G(u_i^T x_t)\} - E\{G(v)\}]^2 \quad (13)$$

Where u_i is an m-dimensional vector, comprising one of the rows of the matrix U. v is a standardized Gaussian variable and G is a non-quadratic function. Maximizing $J_G(u_i)$ leads to estimating u_i by:

$$u_i^+ = E\{x_t g(u_i^T x_t)\} - E\{g'(u_i x_t)\}u_i, \quad (14)$$

$$u_i^* = \frac{u_i^+}{\left\|u_i^+\right\|} \quad (15)$$

Where u_i^* is a new estimated of u_i and g, g' are first and second derivative of G. After every iteration the vectors u^*x are decorrelated using a symmetric decorrelation of the matrix U:

$$U = (UU^T)^{-1/2} U \quad (16)$$

With U matrix $(u_1, u_2, ..., u_n)^T$ of vector u_i and $(UU^T)^{-1/2}$ is obtained by the eigenvalue decomposition of U. This step avoids a direction to be estimated several times and do not privilegiate a vector among others (Cao et all, 2003).

c) *Linear Discriminant Analysis:* LDA is one of the most popular supervised feature extraction techniques. LDA seeks an optimal set of discriminant projection vectors $w = [\phi_1, ..., \phi_d]$, to map the original data space onto a feature space, by maximizing the Fisher criterion: $J_F(w) = \left\|w^T s_b w\right\| / \left\|w^T s_w w\right\|$. Here, S_b and S_w are between-class and within-class scatter matrices of the training sample group respectively, and estimated as follows:

$$s_b = \sum_{i=1}^{c} p_i (m_i - m)(m_i - m)^T = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} p_i p_j (m_i - m)(m_i - m)^T \quad (17)$$

$$s_w = \sum_{i=1}^{c} p_i s_i \quad (18)$$

where c, p_i, m_i, m and S_i represent the number of classes, a *priori* probability of class ω_i, the mean vector of all samples and the covariance matrix of samples in class ω_i, respectively.

By transforming to feature space with maximum separation between classes, higher classification accuracy is obtained (Kuo and Landgrebe, 2004).

3.2 Similarity measures

Classical kernels have proven successful in several applications, but for hyperspectral data, they do not consider full advantage of the rich amount of *a priori* information which is available. This could be due to the fact that these kernels do not take into account the band to band spectral signature effects. Depending on their localism or globalism, classical kernels mostly use the either the Euclidean distance (local) or dot product (global) of two vectors as their similarity measure. In order to effectively make use of information intrinsically available in remote sensing imagery, other metrics except Euclidean distance can be used (Kohram and Sap, 2008). In this paper two similarity measures are introduced as a metric space that are designed for this purpose.

a*) Spectral Angle Mapper* (SAM): it defines similarity between two vectors by measuring angle between them:

$$\alpha(x, x_i) = Arc\cos\left(\frac{x.x_i}{\left\|x\right\|\left\|x_i\right\|}\right) \quad (19)$$

The angle between two vectors is not affected by length of vectors, so SAM is robust to energy difference and is able to exploit spectral characteristics.

b) *Spectral Information Divergence* (SID): This metric considers the discrepancy between probability distributions produced by each pixel vector. It is defined as:

$$SID(x, x_i) = D(x\left\|x_i\right.) + D(x_i\left\|x\right.) \quad (20)$$

With

$$D(x\left\|x_i\right.) = \sum_{l=1}^{n} p_x(l) \log\left(\frac{p_x(l)}{p_{x_i}(l)}\right) \quad (21)$$

Where p_x is a probability distribution vector for each pixel. For $x = [x_1, x_2, ..., x_d]^T$, it is computed as follow:

$$p_i(x) = \frac{x_i}{\sum_{l=1}^{d} x_i} \quad (22)$$

Where x is pixel vector and p is its probability distribution vector. Since both these vectors have the same spectral signature, it is expected to have a minimal distance using every metric. It is not a case of Euclidean distance, for this metric these two vectors might readily be cast very far from each other but SAM and SID are length insensitive, so they can reach desired result. It shows how SID takes into account spectral signature (Kohram and Sap, 2008).

3.3 Training Sample Size

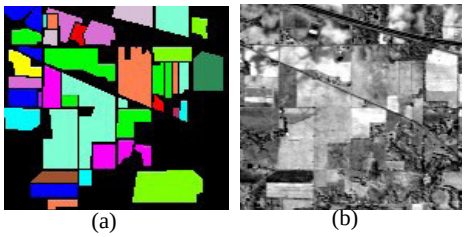
In high dimension space, training sample size has strong influence on classification accuracy which due to Hughes phenomena, this actor for parametric classifiers is more important. As mentioned in section 2, SVMs use only support vectors for training; hence they are stable by changing training sample size. For investigating effect of training sample size on SVMs classification accuracy, different training sample sizes are used and obtained results are compared with KNN classifier.

4. EXPERIMENTS

4.1 Dataset

The hyperspectral image used in the experiments acquired by AVIRIS sensor on June 12, 1992 over the northern part of Indiana which is known for the complexity of the conveyed classification problem. It covered an area of mixed agriculture and forestry landscape in the Indian Pine. Because of similarity between classes, discrimination of classes is difficult. So this data can be an appropriate choice for sensitivity analysis of SVMs to mentioned factors. A field-surveyed map consists of

sixteen classes and one unclassified class. Also the availability of reference data makes this hyperspectral image an excellent source for conducting experimental. The size of image is 145 × 145 pixels with 220 bands. In our experiments, similar to the other studies water absorption bands and noisy bands were discarded (Watanachaturaporn et all, 2005). For sensitivity analysis of SVMs, five classes include: Corn, Grass, Hay, Soybean, and Wood are used.



(a) (b)

Figure 2. (a) Reference image (b) single band image (100th band)

4.2 Results

Sensitivity of SVMs according three mentioned situations: feature space, metrics and training size result investigated based on AVIRIS dataset. For this purpose, two usual multiclass strategies, one against one and one against all and two kernels, polynomial and Gaussian used. Kappa coefficient applied as an accuracy indicator. Two dimensional grid search was utilized for the parameter tuning phase which the range of parameters γ and p for Gaussian and Polynomial kernels is respectively $[2^{-2}, 2^{10}]$ and [1, 10]. Also range of penalty value (C) is considered $[2^{-2}, 2^{10}]$. Table 1 presents the obtained accuracy in each situation of one against one and one against all by two kernels of polynomial and Gaussian RBF.

Table 1. Sensitivity analysis of SVMs respect to three factors by using *Kappa coefficient*

Method	Kernel	Factors	techniques	Kappa
One against One	Gaussian	Feature Space	PCA	96.34
			ICA	96.02
			LDA	96.76
			Original Space	96.24
		Metrics	Euclidean	96.24
			SAM	96.34
			SID	88.16
		Training Sample size (SAM)	10%	92.73
			40%	95.3
			70%	96.13
			100%	96.24
	Poly	Feature Space	PCA	95.30
			ICA	93.3
			LDA	96.29
			Original Space	95.29
One against All	Gaussian	Feature Space	PCA	96.34
			ICA	95.87
			LDA	96.50
			Original Space	96.24
		Metrics	Euclidean	96.24
			SAM	96.34
			SID	88.96
		Training Sample Size (SAM)	10%	92.78
			40%	94.68
			70%	96.03
			100%	96.24
	Poly	Feature Space	PCA	95.97
			ICA	94.4
			LDA	95.87
			Original Space	96.13

4.2.1 Effect of Feature Space

PCA, ICA and LDA feature extraction methods used in order to transformation of spectral space into feature space. As shown in Figure 3, classification with LDA features and Gaussian kernel has the highest accuracy in both multiclass classification strategies. Moreover, PCA improved classification accuracy slightly. But ICA features degrade accuracy especially when the polynomial kernel was used (Figure3).

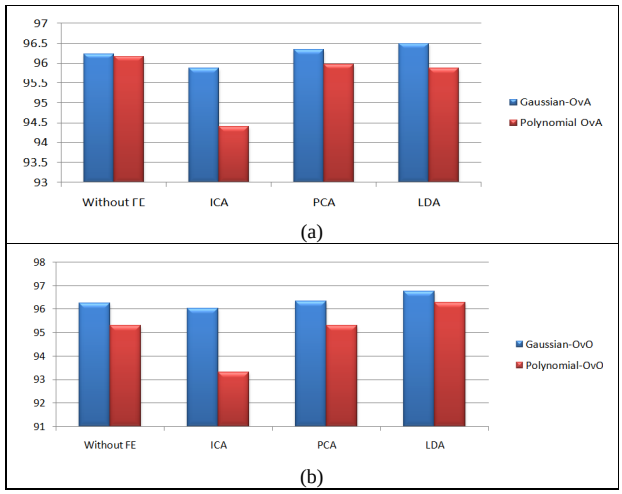


Figure 3. Effect of input space on classification accuracy for (a) one against all strategy (b) one against one strategy

4.2.2 Effect of Similarity Measures

Three different similarity measures were presented which have acceptable potential; Euclidean, SAM and SID used as metrics of SVMs classifiers. As result is shown in Figure 4, SAM can improve classification accuracy slightly. In contrast SID couldn't present acceptable accuracy in both of two strategies of one against one and one against all.

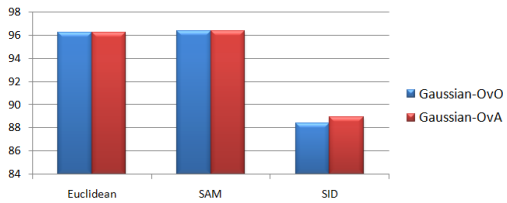


Figure 4. Effect of metrics on classification accuracy

4.2.3 Effect of training sample size

For evaluating the effect of training sample size on SVMs performance, four training sample size were used as the input dataset of classification. Regarding to high potential of Gaussian kernel with SAM metrics, they were used for evaluating the sensitivity of SVMs to training sample size. As it appears from Table 1 and Figure 5, the Kappa coefficient was always greater than KNN in both of one against one and one against all strategies. However there was not any meaningful behaviour of SVMs regarding to decreasing of training data size in comparison of KNN method.

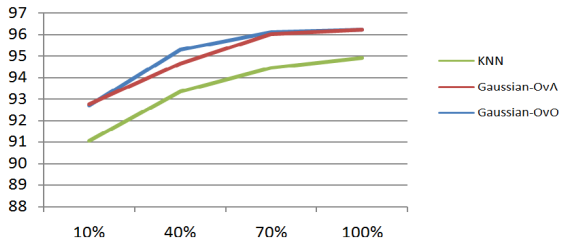


Figure 5. Effect of training sample size on classification accuracy

5. CONCLUSION

In this paper sensitivity analysis of SVMs in respect to different situations of feature space, metric and training sample size investigated. Obtained results about feature extraction methods proved that LDA presents higher classification accuracy rather than other feature spaces such as PCA and ICA. Evaluation about space metrics, shows that the SAM has better performance in comparison of other metrics such as SID and Euclidian. Assessing the training sample size in our investigation, shows that although in same situation, SVMs have better performance than KNN classifier, but the potential of them still depends on the size of training data set. Consequently, although there are good potentials in SVMs, the performance of these classifiers directly is dependent on the decision about different factors such as kernel type and its parameters. Manual determination of optimum value of these parameters, are generally time consuming and needs an expert operator. Further investigations, should be done in direction of automatic determination of these parameters.

6. REFERENCE

Bell, A., Sejnowski, T.J., 1995. An information maximization approach to blind separation and blind deconvolution, Neural Comput. 7, pp 1129–1159.

Camps, G., Bruzzone, Lorenzo., 2005. Kernel-Based Methods for Hyperspectral Image Classification. IEEE Transactions on Geoscience and Remote Sensing, pp. 1-12.

Cao, L.J., Chua, K.S., Chong, W.K., Lee, H.P., and Gu, Q.M., 2003. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. Neurocomputing 55, pp. 321 – 336.

Chi, M., Bruzzone, L., 2007. Semisupervised Classification of Hyperspectral Images by SVMs Optimized in the Primal. IEEE Transactions on Geosciences and Remote Sensing, pp. 1870-1880.

Du, P., Wang, X., Tan, K., and M.Foody, G., 2008. Impacts of Noise on the Accuracy of Hyperspectral Image Classification by SVM. Proceedings of the 8th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, pp. 138-144.

Fauvel, M., Chanussot, J., and Benediktsson, J. A., 2006. Evaluation of Kernels for Multiclass Classification of Hyperspectral Remote Sensing Data. IEEE, pp. 813-816.

Guo, B., Gunn, S., and Damper, R. I., 2008. Customizing Kernel Functions for SVM-Based Hyperspectral Image Classification. IEEE Transactions on Image Processing, pp. 622-629.

Hughes, G. F., 1968. On the mean accuracy of statistical pattern recognition. *IEEE Trans. Inform. Theory*, pp. 55-63.

Kohram, M., Sap, M.N.M., 2008. Composite Kernels for Support Vector Classification of Hyper-Spectral Data. pp. 360–370.

Kuo, B.C., Chang, K.Y., 2005. Regularized Feature Extractions and Support Vector Machines for Hyperspectral Image Data Classification. Springer, Verlag Berlin Heidelberg, pp. 873.879.

Kuo, B.C., Landgrebe, D.A., 2004. Nonparametric Weighted Feature Extraction for Classification. IEEE, Transactions on Geoscience and Remote Sensing, pp. 1096-1105

Landgrebe, D., 2002. Hyperspectral image data analysis. *IEEE Signal Process. Mag*., pp. 17-28.

Melgani, F., Bruzzone, L., 2004. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. IEEE Transactions on Geoscience and Remote Sensing, pp. 1778-1790.

Mercier, G., Lennon, M., 2003. Support Vector Machines for Hyperspectral Image Classification with Spectral-based Kernels. IEEE.

Pal, M., Mather, P.M., 2004. Assessment of the Effectiveness of Support Vector Machines for Hyperspectral data. Elsevier, Future Generation Computer Systems, pp. 1215-1225.

Varshney, P. K., Arora, M. K., 2004. Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data. Springer, Berlin Heidelberg New York, pp. 133-155.

Watanachaturaporn, P., K.Arora, M., and Varshney, Pramood., 2005. Hyperspectral Image Classification Using Support Vector Machines: A Comparsion with Decision Tree and Neural Network classifiers. ASPRS, Baltimore, Maryland.

Zhang, L., Huang, X., 2010. Object-oriented Subspace Analysis for Airborne Hyperspectral Remote Sensing Imagery. Elsevier, Neurocomputing, pp. 927-936.