SCENE CLASSIFICATION USING SUPPORT VECTOR MACHINES WITH LDA

N. VEERANJANEYULU#1, AKKINENI RAGHUNATH#2 , B. JYOSTNA DEVI #3 & VENKATA NARESH MANDHALA#4

VFSTR UNIVERISTY, VADLAMUDI, INDIA

{veeru2006n1,akkineniraghunath2,jyostna.bodapati823,mvnaresh.mca4}@gmail.com

ABSTRACT

Scene classification, the classification of images into semantic categories is a challenging and important problem nowadays. We present a procedure to classify real world scenes in eight semantic groups of coast, forest, mountain, open country, street, tall building, highway and inside city using support vector machines. Traditional classification approaches generalize poorly on image classification tasks, when the classes are non-separable. In this paper we used Support Vector Machine (SVM) for scene classification. SVM is a supervised classification technique, has its roots in Statistical Learning Theory and have gained prominence because they are robust, accurate and are effective even when using a small training sample. By their nature SVMs are essentially binary classifiers, however, they can be adapted to handle the multiple classification tasks common in scene classification. This paper shows that support vector machines (SVM's) can generalize well on difficult scene classification problems. SVMs can efficiently perform a non-linear classification using kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. In this paper 4 types of kernels (linear, polynomial, gaussian and sigmoidal kernels) are used with support vector machines. It is observed that Gaussian kernel outperform other types of kernels. Moreover, we observed that a simple remapping of the input x to x' improves the performance of linear SVM's to such an extent that it makes them, for this problem, a valid alternative to RBF kernels.

Keywords: *Support vector machine, Kernel, cross validation, dimensionality reduction, linear kernel, polynomial kernel, gaussian kernel and sigmoidal kernel.*

1. INTRODUCTION

The statement "Machines shall rule the world" is no more a mere figment of imagination, for it's the promise of Machine learning. In our quest to design and manufacture robotic replicates of humans we haven't come a long way, yet we have with us a set of techniques, algorithms and models which serve as a good starting point.

We consider here a fundamental problem of computer vision, i.e. enabling computers to see the way we see things. We in future wish our machines would match the capabilities of human vision. Its interesting to note that, every second we receive tremendous amount of visual data and almost unconsciously we process this information very quickly. Classifying an object as table, a ball, or a scene as mountain or river is pretty trivial for us. We can in fact process amazingly more complex information. It is a well-known fact that robotic vision compares miserably with our eyes. Here, we intend to make a start towards our goal by considering a very trivial problem by the standard of human vision and that is scene classification. Given an image of the scene we wish to classify it as say a mountain, forest, city, street etc.

In classic pattern recognition problems, classes are mutually exclusive by definition. When the classes are mutually exclusive (linearly separable) then classification task is trivial. Classification errors occur if the classes overlap that is when the classes are non-separable then classification becomes non trivial and classic methods for classification may not yield better performance. Classes are by definition non separable (not mutually exclusive) in semantic scene and in medical diagnosis. Such a problem poses challenges to the classic pattern recognition paradigm and demands a different treatment. Experiments show that our methods are suitable for scene classification; furthermore, our work appears to generalize to other classification problems of the same.

The theory of linear discriminants dates back to the 1930s, when Fisher proposed a procedure for classification. In the field of artificial intelligence attention was drawn to this problem by the work of Frank Rosenblatt, who starting from 1956 introduced the perceptron learning rule. Perceptron uses simple learning algorithm which can find linear patterns in data. A more powerful idea, multilayer perceptrons (MLP) are introduced in 1980s. MLP is a network of perceptrons with continuous activation functions and is very slow in learning.

In, a SVM based scene classification system is described. This approach from is compared with the Linear discriminant Analysis (LDA) system and the results favor the SVM model with higher percentage of accuracy. In experiments pertaining to our paper show results that, the SVM model consistently surpasses the other models in terms of efficiency and accuracy.

2. SUPPORT VECTOR MACHINES

SVMs are among the best (and many believe are indeed the best) "off-the-shelf" supervised learning algorithm. Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A schematic example is shown in the illustration below. In this example, the objects belong either to class GREEN or RED. The separating line defines a boundary on the right side of which all objects are GREEN and to the left of which all objects are RED. This is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective groups (GREEN and RED in this case) with a line.

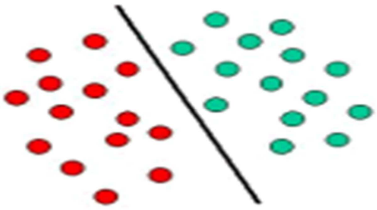

Figure 1: Linearly Separable Data

The SVM is a learning algorithm for classification. It tries to find the optimal separating hyper plane such that the expected classification error for unseen patterns is minimized. Thus, the SVM has very good generalization performance. More precisely, given a set of training samples x_i and the corresponding decision values $y_i \in \{-1, 1\}$, the SVM aims at finding the best separating hyper plane given by the equation $w^T x + b$ that maximizes the distance between the two classes. The set of points for which $w^T x + b \geq 1$ are set to belong to one class and those points for which $w^T x + b \leq -1$ are classified into another. The goal of the optimizing function is to maximize the distance between the two hyper-planes given by
$1/w^T w$. This problem can be equivalently formulated of the form minimize $w^T w$ subject to the constraint that

$$y_i (x_i^T w + b) \geq 1, \qquad \forall i$$

The solution to this optimization problem can be easily obtained when the data are linearly separable. However, complications arise when the data are non-separable.

Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). This situation is depicted in the illustration below. Compared to the previous schematic, it is clear that a full separation of the GREEN and RED objects would require a curve (which is more complex than a line). Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyper-plane classifiers. Support Vector Machines are particularly suited to handle such tasks.

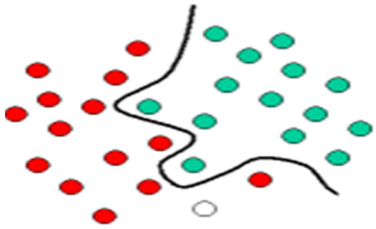

Figure 2: Non-Linearly Separable Data

In this case, the SVM handles non-separable data by introducing slack variables ξ_i. In this case, the optimization problem can be reformulated of the form minimize $(||w||^2)/2 + C(\Sigma\xi_i)^k$
subject to the constraint that

$$x_i^T w + b \geq 1 - \xi_i \,, \ \forall y_i = 1$$
$$x_i^T w + b \geq -1 + \xi_i \,, \ \forall y_i = -1$$
$$\xi_i \geq 0$$

The primal problem as stated above can be converted to its equivalent dual form maximize $\Sigma\alpha_i - (½)^* \Sigma\alpha_i\Sigma\ \alpha_j\ y_i\ y_j\ x_i^T x_j$ subject to $0 \leq \alpha_i \leq C$ and $\Sigma\alpha_i\ y_i = 0$.

In general, the term $x_i^T x_j$ can be rewritten in terms of the kernel functions as $K(x_i, x_j)$.

2.1 Kernel Methods:

When the data is non linearly separable in the original space it can be tranformed to a higher dimensional in which it can be linearly separable. Nonlinear transformation of data to a higher dimensional feature space is achieved by a Mercer kernel. Pattern analysis methods are implemented such that the kernel feature space representation is not explicitly required. They involve computation of pair-wise inner-products only. The pair-wise inner-products are computed efficiently directly from the original representation of data using a kernel function (Kernel trick). In this case, the problem can be equivalently understood in terms of projecting the input data into a higher dimensional space where they are separated using parallel hyper planes. The same is illustrated in the following figure. The original input data in two dimensional space is non linearly separable. It is transformed to three dimensional space using the following kernel function.

$$F(\mathbf{x}) = \{x^2, y^2, \sqrt{2}xy\};\ \ Z = \{z_1, z_2, z_3\}$$

After the transformation the data is linearly separable in three dimensional space.

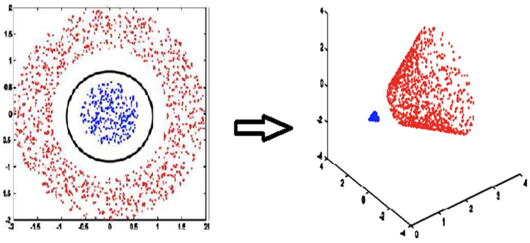

Figure 3: Transformation Of Data To Higher Dimensional Space

3. METHODOLOGY

3.1 Scene Categories Dataset as Input:

In this paper we used scene image category dataset for experimental results. This dataset contains 8 outdoor scene categories: Coast, mountain, forest, open country, street, inside city, tall buildings and highways. There are 3600 colour images each of size 256 X 256 pixels. All the images are labelled.

Figure 4: Sample Scenes Of All Classes

3.2 Feature Extraction:

Each image is divided into 36 blocks and then 23 features are extracted from each block of the image. For each image 36 X 23 dimensional features are available. The 23 dimensional features include color histogram, edge directed histograms and entropy of wavelet coefficients extracted for local blocks of an image for a particular scene.

3.3 Normalization

Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

Feature standardization makes the values of each feature in the data have zero-mean and unit-variance. This method is widely used for normalization in many machine learning algorithms (e.g., support vector machines, logistic regression, and neural networks). In general, we first calculate the mean and standard deviation for each feature, and then, subtract the mean in each feature. Then, we divide the values (mean is already subtracted) of each feature by its standard deviation.

$$X' = (X - \mu)/\sigma$$

Where ‘μ’ is the mean and σ is the standard deviation.

3.4 N-Fold Cross validation:

Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on ``new" data. This is the basic idea for a whole class of model evaluation methods called ***cross validation.***

K-fold cross validation is one way to improve over the holdout method. The data set is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times. The variance of the resulting estimate is reduced as *k* is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set *k* different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over. In our paper we experimented with 10-fold cross validation. Each time 75% data of the each class is used for training and remaining data is used for testing.

3.5 LDA:

Linear discriminant analysis (LDA) and the related Fisher's linear discriminant are methods used in statistics, pattern recognition and machine learning to find a linear combination of features which separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

Fisher's linear discriminant is a classification method that projects high-dimensional data onto a line and performs classification in this one-dimensional space. The projection maximizes the distance between the means of the two classes while minimizing the variance within each class.

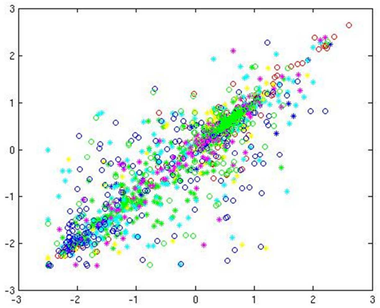

Figure 5: Original Data With 2 Features

If there are two classes transform the d-dimensional data onto 1 dimensional space (that is project onto a line). If there are c classes the data need to be projected from d-dimensional space to c-1 dimensional space. In our scene dataset we have 8 classes so the data is projected onto 7 dimensional space.

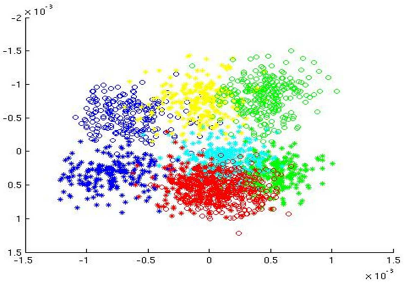

Figure 6: Original Data Projected Onto 7 Dimensional Space

3.6 SVM Training:

The output of LDA is passed for the training using SVM. By using different kernel functions, SVM can implement a wide variety of learning algorithms. It is well known that the SVM has a great potential to perform well. However, the performance of the SVM is very closely tied to the choice of the optimal kernel functions. There has been a lot of research over the last few years on algorithms to help choose the exact type of kernel for a given problem with a certain set of features. Most of these methods are based on simple heuristics based on the knowledge of the input data and there has not been any standardized method to obtain the best kernel. Hence, the choice of the optimal kernel has reduced to a trial and error procedure in most scenarios.

There exist many popular kernel functions that have been widely used for classification.

3.6.1 Linear Kernel:

The Linear kernel function is of the form

$$K(x_i , x_j) = (x_i^T x_j)$$

3.6.2 Polynomial kernel:

The polynomial kernel function is of the form

$$K(x_i , x_j) = (1 + x_i^T x_j)^p$$

3.6.3 Gaussian kernel:

This kernel is also known as Radial basis function kernel. This is a reasonable measure of x_i and x_j's similarity, and is close to 1 when x_i and x_j are close, and near 0 when x_i and x_j are far apart. This Gaussian kernel corresponds to an infinite dimensional feature mapping. The radial basis function is given by

$$K(x_i , x_j) = \exp(−\gamma\|x_j − x_i \|^2)$$

3.6.4 Sigmoidal kernel:

The sigmoidal kernel function is of the form

$$K(x_i , x_j) = \tanh(\gamma\ x_i^T x_j+C)$$

3.7 Testing:

There are number of kernels that can be used in Support Vector Machines models. These include linear, polynomial, radial basis function (RBF) and sigmoid:

$$K(\mathbf{X}_i,\mathbf{X}_j)=\begin{Bmatrix} \mathbf{X}_i\cdot\mathbf{X}_j & \text{Linear} \\ (\gamma\mathbf{X}_i\cdot\mathbf{X}_j+\mathrm{C})^d & \text{Polynomial} \\ \exp(-\gamma|\mathbf{X}_i-\mathbf{X}_j|^2) & \text{RBF} \\ \tanh(\gamma\mathbf{X}_i\cdot\mathbf{X}_j+\mathrm{C}) & \text{Sigmoid} \end{Bmatrix}$$

Where:

$$K(\mathbf{X}_i,\mathbf{X}_j)=\phi(\mathbf{X}_i)\bullet\phi(\mathbf{X}_j)$$

that is, the kernel function, represents a dot product of input data points mapped into the higher dimensional feature space by transformation Φ. All the four types of kernels are used for SVM modelling.

3.7.1 Test Results using Linear kernel

	Coast	Forest	Highway	Inside city	mountain	Open country	street	Tall building
Coast	54	1	7	2	4	20	1	1
Forest	0	75	0	0	5	2	0	0
Highway	9	0	36	1	8	6	4	1
Inside city	1	1	3	54	3	1	5	9
mountain	1	10	2	4	52	12	6	7
Open country	8	12	4	0	10	66	2	1
Street	1	0	3	8	6	0	53	2
Tall building	5	1	1	13	8	2	3	56

Accuracy: 66.27

3.7.2 Test Results using Polynomial kernel, degree-2

	Coast	Forest	Highway	Inside city	mountain	Open country	street	Tall building
Coast	56	0	2	2	19	11	0	0
Forest	2	67	1	0	8	2	0	2
Highway	5	2	23	1	22	12	0	0
Inside city	0	10	1	32	19	8	1	6
mountain	2	0	0	1	84	7	0	0
Open country	3	4	0	1	84	7	0	0
Street	1	0	1	1	27	13	30	0
Tall building	3	2	0	3	40	5	1	35

Accuracy: 58.26

3.7.3 Test results using Gaussian (RBF) kernel

	Coast	Forest	Highway	Inside city	mountain	Open country	street	Tall building
Coast	62	1	4	1	5	13	1	3
Forest	0	72	0	0	5	1	0	4
Highway	5	0	32	3	9	11	4	1
Inside	1	1	2	54	0	2	4	13

city								
mountain	1	3	0	5	67	7	4	7
Open country	2	5	1	0	15	78	0	2
Street	0	0	2	8	7	1	48	7
Tall building	6	0	1	6	8	1	2	65

Accuracy: 71.03

3.7.4 Test Results using Sigmoidal kernel

	Coast	Forest	Highway	Inside city	mountain	Open country	street	Tall building
Coast	57	1	5	4	5	17	1	0
Forest	0	75	0	0	5	2	0	0
Highway	7	0	34	3	10	8	3	0
Inside city	1	2	3	52	1	2	6	10
mountain	2	6	0	4	62	8	8	4
Open country	2	8	1	0	11	79	1	1
Street	0	0	4	9	6	1	51	2
Tall building	6	1	1	12	6	2	4	57

Accuracy: 69.39

4. COMPARATIVE ANALYSIS

The PCA takes advantage of the fact that, under admittedly idealized conditions, the variation within class lies in a linear subspace of the image space. Hence, the classes are convex, and, therefore, linearly separable. One can perform dimensionality reduction using linear projection and still preserve linear separability.

LDA seeks to reduce dimensionality while preserving as much of the class discriminatory information as possible. Let the set of D-dimensional samples x1 , x2 , ... xN , N1 of which belong to class ω1, and N2 to class ω2. We seek to obtain a scalar y by projecting the samples x onto a line y= wTx

Among all the possible lines we would like to select the one that maximizes the separability of the scalars. The basic idea behind Fisher Linear discriminant analysis is to find the best set of vectors that can minimize the intra cluster variability while maximizing the inter cluster distances FLD was able to achieve a much better performance even under a highly reduced dimensional space. Results without applying LDA and with using LDA

Following are the average Accuracies for SVM classifier with different kernels and different features

Features	Kernel type	Accuracy
Image pixels	Linear	52.27
	Polynomial	50.68
	RBF	65.76
	Sigmoidal	62.13
LDA coefficients	Linear	66.27
	Polynomial	58.26
	RBF	71.03
	Sigmoidal	69.39

5. CONCLUSION

In this paper, we consider the problem of scene classification and use the Support vector machines for classification. Initially we extracted the features, we normalized the scene images. These are then fed into the SVM with different types of kernel functions and the classification error and accuracy are noted in each case. We explored the use of four different kernels namely linear, polynomial, radial basis function and sigmoid. The performance under different types of parameters was studied and tabulated. Our results indicate that RBF kernel gives the best performance compared to all the other kernels (with different parameter settings). Next, we consider feature dimension reduction methods. More specifically, we test the performance of the SVM classifier when the FLD with RBF kernel gives the best results.

REFERENCES

[1] Mukundhan Srinivasan and Nivas Ravichandran, "*Support Vector Machine Components - Based Face Recognition Technique using 3D Morphable*", Modeling, International Journal of Future Computer and Communication, Vol. 2, No. 5, October 2013.

[2] John Shawe-Taylor, "*Kernel Methods and Support Vector Machines*", lecture notes, June 2009.

[3] Geoff Gordon, "*Support Vector Machines and Kernel Methods*".

[4] S. M. Metev and V. P. Veiko, "*Laser Assisted Microtechnology"*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[5] J. Breckling, Ed., "*The Analysis of Directional Time Series: Applications to Wind Speed and Direction"*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[6] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "*A novel ultrathin elevated channel low-temperature poly-Si TFT,"* IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[7] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "*High resolution fiber distributed measurements with coherent OFDR,"* in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.

[8] jst@cs.ucl.ac.uk

[9] M. Shell. (2007) IEEEtran webpage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/ macros/latex / contrib/IEEEtran/

[10] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[11] "*PDCA12-70 data sheet,*" Opto Speed SA, Mezzovico, Switzerland.

[12] A. Karnik, "*Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP,*" M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[13] J. Padhye, V. Firoiu, and D. Towsley, "*A stochastic model of TCP Reno congestion avoidance and control,*" Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[14] *"Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification"*, IEEE Std. 802.11, 1997.