SVM-based Classification of Suerpixels of Images into Labelled Groups

Gaurav Kshirsagar
(Person #: 50134944)
University at Buffalo
Buffalo, NY, USA
gkshirsa@buffalo.edu

Abstract

Image segmentation has been one of the key fields of work and much research as people rely on multimedia data for a lot of purposes. From photo sharing between friends to security camera footage, there is an attempt to make sense of images and classify them based on their features. Thus, there is now a shift from general image segmentation to a much wider and collective analysis of large image datasets. Such approaches usually involve Machine Learning techniques since the need for automated processing is clear with such large amount of data. This project uses the Support Vector Machine classifier to train a model based on the features observed in a training dataset and then predict such similar features that occur in other datasets. We use a subset of superpixels, elicit features corresponding to color, texture and geometric information, and train a model. Then with one or more fold validation of the results, we determine the accuracy of prediction.

Keywords-support vector machine, classification, superpixel, accuracy, radial basis function

1. Introduction

Classification of images into semantically related regions has been a challenging problem in computer vision and image processing. Machine learning models have been found to be more useful in this scenario to learn features that exist in a given set of images and then the models can be used to classify test data. [1]

The aim of this experiment and research based project is to classify superpixels in 143 test images into classes

C = {sky, tree, road, grass, water, building, mountains, foreground objects}. The approach is to train a classifier based on support vector machine (SVM) with different features extracted from a separate dataset of 572 images. Extracting important features and labelling them correctly are the major factors that determine the accuracy of the classification.

A SVM based classifier sample implementation is provided to start. However, for systems running MATLAB versions previous to 2014, the function 'fitcsvm()' used to train the classifier in the sample code does not work. Thus, it is required to use LIBSVM [5] which is an SVM implementation for MATLAB.

The results based on this basic sample are analyzed and documented in this project. Different parameters of the SVM classifier are tested and conclusions are drawn based on the outputs. Other ways to improve the accuracy of classification are considered and effort is made to implement them. The results of these experiments are all documented and discussed as to why or why not the intended results were obtained.

2. Related Work

The classification problem in Computer Vision and Image Processing is a very open one and various approaches have been developed and are being debated about in this field. Using SVM is a novel one of them which uses features of existing images dataset to train and generate a model. This model is then used it to classify a separate set of superpixels from test images dataset into a set of labels as explained in A Practical Guide to Support Vector Classification [1]. This was used as a basis for understanding the concept of SVMs and their application to image processing.

Another such paper is A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods [2]. This paper exclusively deals with sigmoid kernel based classification scheme. It suggests that it was previously pointed out that sigmoid kernels are may not be positive semi-definite (PSD). A kernel is a symmetric continuous function $K:[a,b]\times[a,b]\rightarrow R$ so that $K(x,s)=K(s,x)$.
K is said to be non-negative definite (or positive semidefinite) if and only if

$$\sum_{i=1}^{n}\sum_{j=1}^{n}K(x_i,x_j)c_ic_j\geq 0$$

for all finite sequences of points x1,...,xn of [a,b] and all choices of real numbers c1,...,cn. [3]. So, it validates that

sigmoid kernel is conditionally positive definite (CPD). However, it is not better than the RBF kernel in general. It may give the local minimum for some parameters but its not a viable option.

Decomposing a Scene into Geometric and Semantically Consistent Regions [4] takes into consideration the geometric properties of the images in consideration. Individual region potentials are considered like buildings tend to be straight and trees tend to be green and textured. Inter-region potentials consider the fact that the typical foreground objects like people, vehicles are in front of the typical background objects like mountains, buildings, etc. It takes into account what regions are typically adjacent and where the typical boundaries between foreground and background objects would appear. Based on these conditions there is an energy function

$$E(R, S, G, A, vhz, K \mid I, \theta) =$$
$$+ \theta_{horizon} \psi_{horizon}(vhz)$$
$$+ \theta_{regionPr} \psi_{region\ r}(Sr, Gr, vhz; Ar, Pr)$$
$$+ \theta_{pairPrs} \psi_{pair\ rs}(Sr, Gr, Ss, Gs; Ar, Pr, As, Ps)$$
$$+ \theta_{boundaryPpq} \psi_{boundary\ pq}(Rp, Rq; \alpha_p, \alpha_q).$$

The observations made are evaluated with this energy function and the global energy of the resulting configuration is then evaluated, and the move is accepted only if this energy improves, ensuring that our inference is continuously improving a coherent global objective.

3. Algorithm

The approach in classifying the image superpixels is to train a model based on SVM and then use it to label superpixles in other images into the classes in C. So the two main tasks are Training and Testing. For the purpose of experiment and with the given time constraints for this project, superpixels in the range [1000, 100000] are chosen at random. The training and testing is done on this same set of superpixles.

Given implementation takes 38 features which describe color and texture in the images. Color takes into account the RGB and HSV models. Maximum values of the texture responses to applied filters are considered.

LIBSVM functions are used for training and testing. 5-fold cross validation is done on the superpixels. For training, the function is:

svm-train [options] training_set_file model_file

For testing the function is:

svm-predict [options] test_file model_file output_file

The performance of the classification is judged on the basis of confusion matrix. Confusion matrix is a graph shows the distribution of the pixels into classes and the accuracy of the prediction per class. Average accuracy of the prediction is displayed on the top of the graph.

The start of the experiments was with Linear kernel on with C-SVC. Results ttok a very long time and were not acceptable. Optimization took too many iterations. Then, the next step was a Polynomial kernel which was much faster. RBF provided much faster optimization and thus it's the best kernel used with the results so far. Values of cost parameter tested were 100-10000.

Then the nu-SVC type of SVM was used which has the nu parameter which approximates the fraction of training errors and support vectors [9]. Values ranging 0.1 to 0.0001 were used with different kernels.

All the test cases and the results obtained are included a separate 'outputs.txt' file for reference.

4. Experimental analysis and results

The focus of this project is on the experiments run and results obtained which are then used to fine tune the training and testing process for the given dataset. The tests were run on a set of superpixles in range [1000, 100000] and with [2, 5] fold cross validation. Tests are run using C_SVC for C-SVM classification or NU_SVC for nu-SVM classification which are the two types in LIBSVM for multi-class classification. Kernel functions tested are:

Name of function	Function
Linear	u'*v
Polynomial	(gamma*u'*v + coef0)^degree
Radial Basis Function:	exp(-gamma*|u-v|^2)
Sigmoid	tanh(gamma*u'*v + coef0)

Table 3.1.1: Kernel functions used

All the tests are run on a user machine (no server) with the following configuration:
Matlab Version : R2013a
Processor: 4th Generation Intel Core i7-4510U 2.00GHz Quad-core
Operating system: Windows 8.1 64 bit
Graphics: AMD Radeon R5 M230 2GB
Memory: 8.0GB DDR3L SDRAM 1600 MHz
In the case of this project and Matlab being the platform for implementation, the graphics card is useless for the floating point calculations as there is no support for Matlab to run optimally on ATI cards [6]

4.1. Different multi-class types used

Classification was done using C-SVC and nu-SVC with different kernels and each with a different set of parameters. Results are for a single fold and in some cases, if the output was initially good, that is, classification was into more classes than one, more folds were run to check for better output. Otherwise the basic tests were run on the single fold.

With nu parameter in nu-SVC the output was found to be much faster than the C-SVC type. Initial tests had nu values starting from 0.1 which produced errors. So, further less values were tested. 0.001 and less values were found to be most optimum for the RBF kernel as it produced the fastest output. Sigmoid kernel also produced similar results but not the fastest.

4.2. General Observations

The results observed are mainly dependent on the classification kernel used, meaning, the results of Linear kernel are similar in the case of C-SVM and nu-SVM. Following are the general observations for each kernel:

Kernel used	Statistics of output		
	Number of superpixels	Iterations to converge	Maximum runtime (in sec)
Linear	1000-5000	Order of 10^7	2400
Polynomial	1000-3000	Order of 10^7	600
RBF	1000-50000	12000	600
Sigmoid	1000-50000	2000	450

Table 4.2.1: Kernel-wise statistics (The iterations and runtime are for the maximum no. of superpixels in the table)

The runtime for linear kernel generally higher than others for the accuracy of classification it gives. Also, it is much higher for high number of superpixels than stated in the above table to the extent that Matlab freezes. Also, linear kernel takes a very high number of iterations for optimization for each class.

The runtime for polynomial kernel is also generally less than others but gives a poor accuracy of classification. It takes a very high number of iterations for optimization for each class.

Maximum number of experiments have been carried out on the Radial Basis Function kernel. It gives the ability to test on a much larger set of superpixels than linear or polynomial since its runtime is much lower. Tests were run on 1000 to 50000 superpixles. Also, the iterations taken for each class are less.

Sigmoid function takes less runtime as well and more number of superpixles can be tested. Tests were run on 1000 to 50000 superpixles. The iterations taken for each class are less than Radial Basis Function.

However, the RBF is the more favorable since it's the fastest in producing results and this matches some other experimental findings [10].

4.3. Peculiar output cases

This section lists a few output cases which were different from the rest.

Classifier and Kernel	svmtrain function parameters	Super pixels	Result and comments
Nu-SVC RBF	'-s 1 -t 2 -n 0.05 -g 1/44 -h 1'	50000	Matlab Froze
C-SVC RBF	'-s 0 -t 2 -g 0.001 -c 10000.0 -r 0.0 -d 3 -h 1 -b 1 -e 0.001 '	1000	Classification in the range 70-88% Avg Accuracy 12.5% in class 1 Runtime 100s
nu-SVC RBF	'-s 1 -t 2 -n 0.001 -g 0.0001 -h 0'	5000	Classification in the range 34-42% Avg Accuracy 12.5% in class 1 Runtime 80s
C-SVC LINEAR	'-s 0 -t 0 -c 100.0 -d 3 -h 1 -e 0.1 '	5000	Classification in the range 62-88% Avg Accuracy 12.5% in class 1 Runtime 2400s
OP24 nu-SVC POLY	'-s 1 -t 1 -c 1000.0 -n 0.003 -d 2 -g 1/44 -r 0.0 -h 1 -e 1.0 '	3000	Classification in the range 34-44% Avg Accuracy 12.5% in class 1 Runtime 600s
OP27 nu-SVC POLY	'-s 1 -t 1 -c 1000.0 -n 0.003 -d 3 -g 0.001 -r 0.0 -h 1 -e 1.0 '	100000	Classification in the range 54-64% Avg Accuracy 12.5% in class 1 Runtime 11400s
C-SVC RBF	'-s 0 -t 2 -g 0.0001 -n 0.001 -c 1.0	50000	Classification in the range 23-36%

	-e 100.0 -h 0'		Avg Accuracy 12.5% in class 1 Runtime 70s
C-SVC RBF	'-s 0 -t 2 -g 0.0002 -n 0.0001 -c 4000.0 '	10000	Classification in the range 72-88% Avg Accuracy 12.5% in class 1 Runtime 1400s
OP14 nu-SVC RBF	'-s 1 -t 2 -n 0.05 -g 1/44 -h 1'	1000	Classification in the range 77-98% Avg Accuracy 12.5% in class 1 Runtime 150s

Table 4.3.1: Output statistics

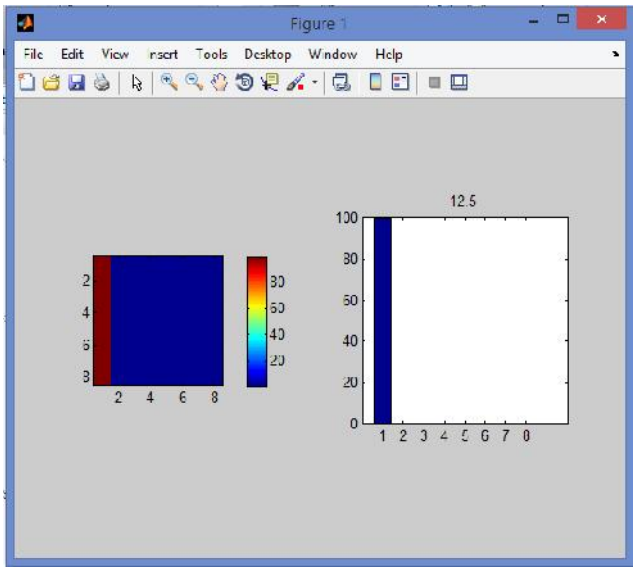

Figure 4.3.1: Output for all the cases, all superpixels labelled into a single class

4.4. Other approaches considered and tried

5. Discussions and conclusions

The general trend of the output is the same and its gets categorized into one single class i.e. class 1. This output could be improvised by inclusion of more pixels. Attempts have been to try and increase the number of superpixels considered but without much change in results. There have also been problems on Matlab freezing for long periods of time of about 4 hours. Even this could not generate an acceptable output if Matlab crashed. Thus, improving the accuracy is not possible with the available time and processing power.

VlFeat [8] and Pascal2 [7] are some other implementations of image processing algorithms and tools. However, the accuracy of the basic SVM model is limited. These algorithms use features much more than the color models, textures, geometric details. Many more such approaches also exist like global energy estimation as discussed in this paper previously.

6. References

[1] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, A Practical Guide to Support Vector Classification, http://www.csie.ntu.edu.tw/~Cjlin , 2010

[3] Hsuan-Tien Lin and Chih-Jen Lin, A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods

[4] Stephen Gould, Richard Fulton, Daphne Koller Decomposing a Scene into Geometric and Semantically Consistent Regions

[5] LIBSVM, an external library for Support Vector Machine implementation in Matlab http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[6] Matlab GPU configuration documentation http://www.mathworks.com/discovery/matlab-gpu.html

[7] Pascal2 analysis and modeling tools http://pascallin.ecs.soton.ac.uk/challenges/VOC/

[8] VlFeat SIFT feature detector and descriptor http://www.vlfeat.org/overview/sift.html

[9] Support Vector Machines http://scikit-learn.org/stable/modules/svm.html

[10] RBF SVM parameters http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

7. Graphs for the table of observations

The graphs describing the table of results being too big to accommodate in half a page are included at the end of this report

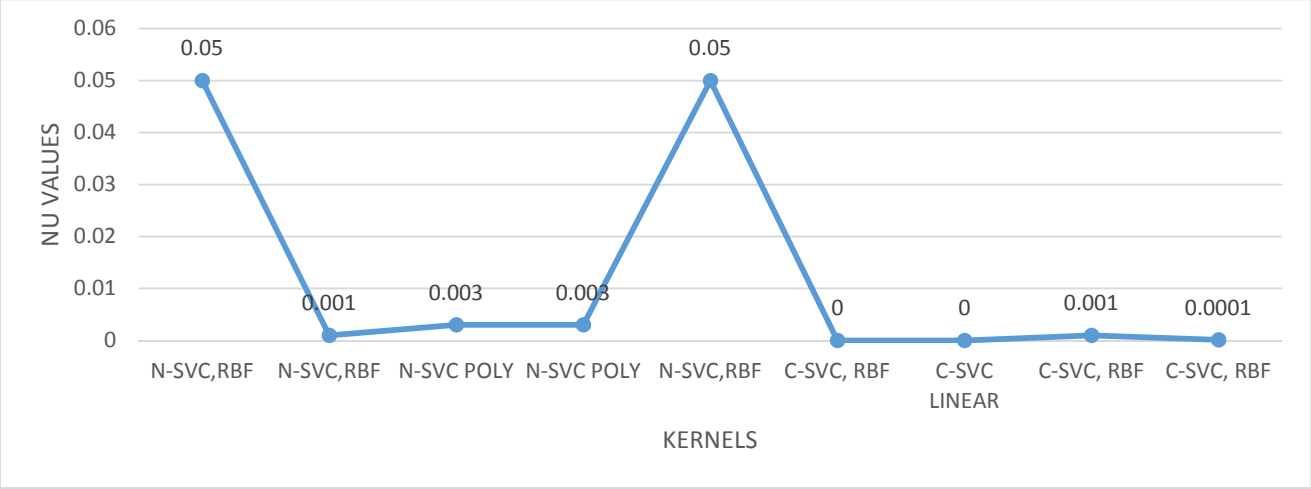

Figure : nu Values for Kernels

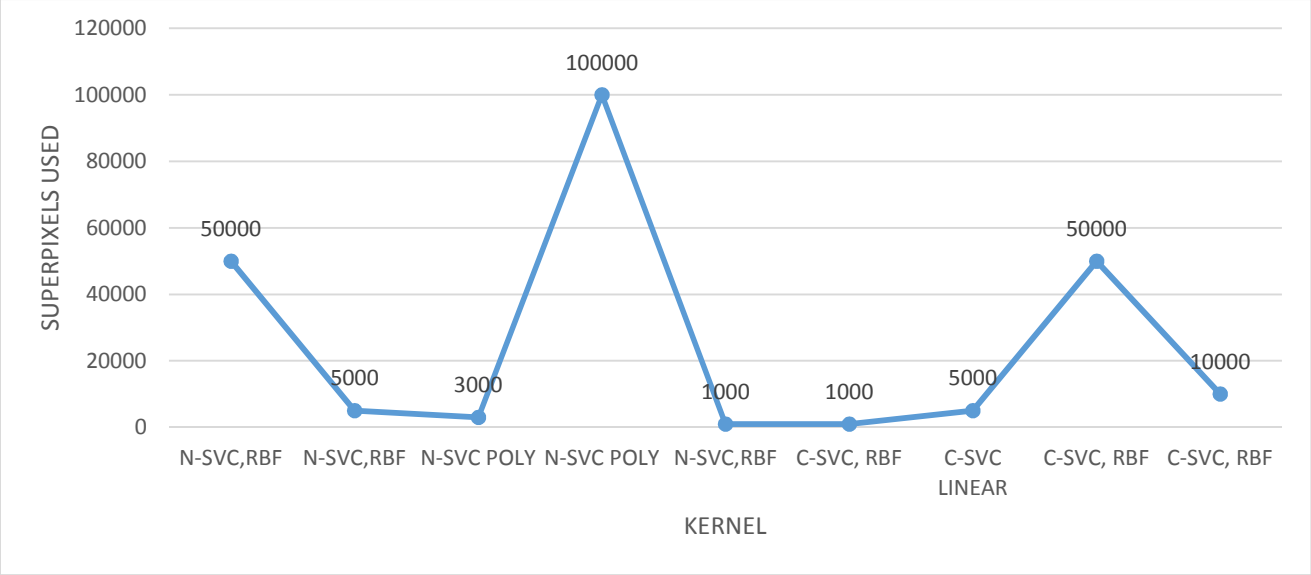

Figure : No of superpixels used for Kernels

Notes : For running the code, use file runfile.m