Testing of the Support Vector Machine for Binary-Class Classification

Matthew Scholten
California State University Long Beach – 001139

NASA Undergraduate Student Research Program (USRP)
Mentor: Dr. Thomas Lu
Jet Propulsion Laboratory, California Institute of Technology*, Pasadena, California,* 91109

ABSTRACT

The Support Vector Machine is a powerful algorithm, useful in classifying data in to species. The Support Vector Machines implemented in this research were used as classifiers for the final stage in a Multistage Autonomous Target Recognition system. A single kernel SVM known as SVMlight, and a modified version known as a Support Vector Machine with K-Means Clustering were used. These SVM algorithms were tested as classifiers under varying conditions. Image noise levels varied, and the orientation of the targets changed. The classifiers were then optimized to demonstrate their maximum potential as classifiers. Results demonstrate the reliability of SMV as a method for classification. From trial to trial, SVM produces consistent results.

1. INTRODUCTION

Computer vision is a field of research, which develops algorithms that allow computers to draw useful information from images, or video feeds. The information encoded in an image can be valuable in many applications, allowing a computer to understand its environment visually, recognize objects, and perform operations autonomously. Applications of computer vision range from medical image analysis to autonomous guidance and maneuvering of spacecraft for docking or hazard avoidance.

Autonomous target recognition (ATR) is one of the biggest challenges facing a computer vision system. Finding and identifying a particular type of target can be difficult. Targets must be identified under varying conditions, and in various orientations. Furthermore, ATR algorithms must be computationally efficient if they are to be done in real time. If an automated system uses a continuous video feed, the algorithm must complete calculation on one frame of the video before the next frame can be processed.

2. BACKGROUND

OT-MACH Correlation for Target Detection

To address the issue of speed versus accuracy in a target recognition system, the Jet Propulsion Laboratory has developed a multistage autonomous target recognition system. This method uses a grayscale optical correlator (GOC) to instantaneously scan an image, and identify subsets of the image data that contain targets. These subsets are considered regions of interest (ROIs) and serve to reduce the amount of data that is extracted and passed to the final stage of the ATR system.

The GOC filter is fast, but tends to produce many false-positive targets. To filter out erroneous ROIs, the GOC is used in combination with an Optimum Trade-off Maximum Average Correlation Height (OT-MACH) filter. The OT-MACH filter limits the number of erroneous features passed from GOC, and extracts the ROIs, which are passed to the classification stage. To perform this task, OT-MACH parameters are optimized though training by an operator to minimize Output Noise Variance, Average Correlation Energy and maximize Average Correlation Height. Each parameter is weighted independently, and is determined by an adaptive step gradient descent algorithm.

ROI Classification with Machine Learning Methods

The second stage of the ATR system uses a machine learner to classify each previously extracted feature as a target, or non-target. The ROIs are considered vectors, in which each pixel value is a dimension, and can be simplified using Principal Component Analysis (PCA). PCA selects the top 18 features of the ROI vector to produce features of smaller dimensionality. Each ROI is then projected into an 18-dimensional feature space. Machine learners are trained on a set of ROI feature vectors, and each is identified as a target or a non-target. Data represented in feature space is subsequently classified according to the machine learner algorithm. This paper documents the use of two types of support vector machines that are used as final classifiers in an autonomous target recognition system.

Free-Response operating characteristic curve (FROC curve)

A Free-Response operating characteristics curve (FROC curve) describes the accuracy of a classification method. SVMs output a ‘score’ for each potential target. This ‘score’ represents the classification accuracy of each sample; if the score is high the sample is positive or true, whereas if it is low the sample is false or negative. Picking a threshold, which defines what “high” and “low” are changes the quantity of samples classified as true versus false. Varying this threshold and plotting the results in an ROC curve gives a measure of the performance of the classification method.

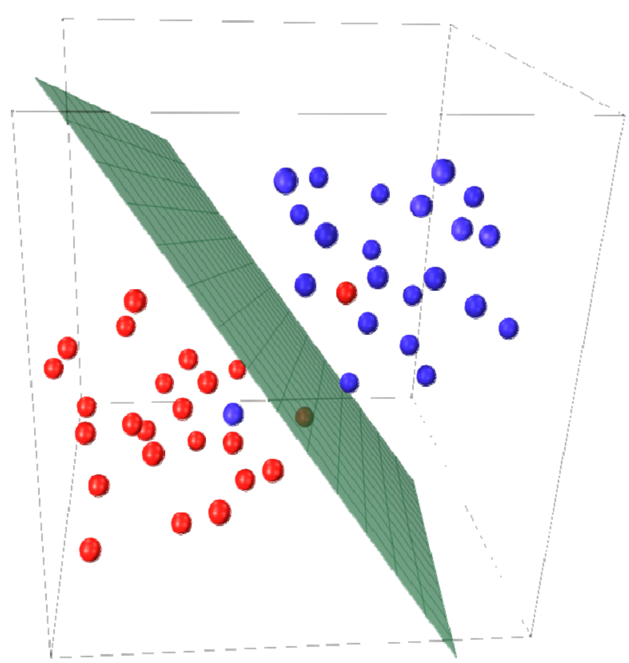
Along the x-axis is the average percent of false positives (negative samples classified as positive). The y-axis represents the percent of true positives (positive samples classified as positive). A curve, which increases along y rapidly, is best; with an ideal scenario being a vertical line along the y-axis.

3. TRAINING METHODS

Support Vector Machines

In order to distinguish classes in highly enmeshed data, as is the case with ATR, Support Vector Machines examine a set of labeled instance pairs in a multidimensional feature-space. SVM attempts to separate the data in to classes. It does so by creating a separating hyperplane along the maximal margin of class separation. In other words, it creates a hyperplane between classes where they are the most different.

The number of features representing each labeled data point determines the dimensionality of feature-space. In this case, 18 feature vectors are selected through PCA. Mapping of feature-space is determined by the SVM kernel, and is optimized by maximizing the class separation of data. The kernel parameters are optimized for the best ‘fit’, and the kernel is selected based on the conditions of input data.



This research was done using the Gaussian Radial Basis Function as a SVM kernel function according to the given optimization problem.

Given a training set of instance-label pairs (xi , yi), i = 1, . . . , l where xi ∈Rn and y ∈{1, −1}^l
SVM requires optimization of the problem:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0.$$

Where ξi is the error term, C > 0 is the penalty parameter for the error term

Training vectors xi are mapped to higher dimensional space by the function φ

Furthermore, the kernel function: K(xi , xj) ≡ φ (xi)T φ (xj)

RBF: K(xi , xj) = exp(−γ xi − xj 2), γ > 0

Linear: K(xi , xj) = xT xj

polynomial: K(xi , xj) = (γxi T xj + r)d , γ > 0.

Sigmoid: K(xi , xj) = tanh(γxi T xj + r)

With kernel parameter γ (gamma) optimized to set the kernel width. Tightening the spread of instances in regions of dense species population, and expanding the spread where population is sparse.

k-Means Clustering with SVM Classification

k-Means clustering with SVM (k-SVM) classification uses supervised and unsupervised learning methods to enhance SVM classification. k-Means clustering places a set of data points in different classes depending on the distance between them. The user specifies a number of classes, and the algorithm attempts to form sets of points form the best 'clusters'. This method uses a priori knowledge about the distribution of the data to attempt to enhance classification. With sufficient samples, a SVM classifier will yield an adequate separation of data, however, with insufficient samples the division will not be maximized. k-SVM addresses this by dividing the training samples among a set of independent SVM classifiers. The algorithm uses a point as a 'mean' to define the classes. Each point in the data set is classified according to which data point it is closest too. k-Means clustering randomly places means in the feature space and then sorts the data into clusters based on distance from the mean. Each cluster's mean is then moved to the average of all the points in that cluster. This is repeated until the means no longer move. Because of the random initial placement of means, the algorithm can converge to different clusters. To counteract this, the process is repeated several times and the clustering of data which best reduces entropy in the clusters is chosen.

The appearance of target objects can vary greatly. They often are portrayed in several orientations throughout a dataset, the scale of a target can vary, or it can belong to multiple subclasses of targets. k-SVM can simplify the problem of identifying varying targets with a single SVM. By distributing target data to a set of SVMs that are trained to identify only one variation of target appearance results can be improved. Learning each of the subclasses separately may increase the classifiers ability to generalize to new data. If image sets are similar but vary only in resolution, or if only the target's scale differs, k-SVM may yield more generalized results.

Controlling the 'fit' of data is of concern during target classification. That is to say, the classifier can be too specialized to its training data to be useful in classifying independent target data. Over-fitting the data means that the classifier emulates too closely the pattern that training data portrays. The classifier may also lose generality when applying the classifier to test data. Under-fitting has the opposite effect – where the classifier only loosely follows the training data.

The 'fit' of data is controlled by kernel parameters, and the outlier error penalty parameter c. Kernel Parameters vary by kernel selection, and a consideration to make during kernel selection is the number of kernels required. In the case of this research project the Gaussian RBF kernel was used, therefore the optimization of the kernel parameter γ (gamma) was required. The SVM classifier also requires the optimization of the parameter 'c' which sets the penalty for misclassifying a point. The parameter c relocates a sample point closer to the center of its class.

The particular c, and γ (gamma) values that are best for a given application are not known, so they must be set by the operator. It is common for arbitrary values to be used as kernel parameters; however a grid-search is an analytical alternative that seems to produce better results. A common strategy is to split the data into training, validation and testing sets. The training set is a set with unknown class-labels, and the validation set is a control, whose class-labels are known. Given a set of parameters, the SVM classifier is trained on the training set, and tested on the validation set. The prediction accuracy obtained from the 'unknown' training set reflects the classification performance of an independent 'testing' set. After several iterations, the set of values that gives the best performance is chosen, and the corresponding classifier is evaluated on the testing set.

4. METHODOLOGY

Baseline performance data was gathered from a SVM, and a k-SVM classifier. Both were trained on a single set of extracted features from long-range sonar image data. A grid-search was performed while sweeping for optimal parameters on image data during this research project. The following grid search process was used for sweeping.

> The interval [a,b] was swept, where the SVM parameter c, and γ (gamma) – here represented as p where $p \in [a,b]$, and [a,b] is divided in to n steps on a log scale starting from the center of [a,b].

At each of the subsequent n steps, parameter values were noted, and the SVM classifier was tested. The process was repeated beginning at the best p on the interval. This time the search interval was shortened, and the number of steps was increased. This was done to increase the resolution of the search around the new p, and provide the best possible parameter for the SVM Classifier.

This thorough sweeping of the kernel parameters was performed on both SVM, and Kmeans Clustering with SVM classifiers. It was done to determine the best possible fit during classification. The sensitivity of the above classifiers was then tested to determine how precise the kernel parameter selection needed to be. The optimal kernel parameters were tested on an independent testing set, and the optimal performance scores were recorded to provide a baseline for further testing. Then, Parameter values were adjusted according to the following table to determine necessary parameter accuracy.

5. RESULTS

SVM						
test number	c	accuracy @ 2FP	#FP @ 90% accuracy	γ	accuracy @ 2FP	#FP @ 90% accuracy
baseline	4.717008585	61.04	24	0.962276766	61.04	24
1	9.43401717	66.78	21.33	1.924553532	66.09	21.65
2	18.86803434	70.09	21.41	3.849107064	65.19	21.96
3	37.73606868	69.57	18.72	7.698214129	61.04	31.44
4	75.47213736	70.43	23.39	15.39642826	56.7	34.22
5	150.9442747	69.04	26.18	30.79285651	47.83	31.14
6	2.358504293	64.87	25.04	0.481138383	56.52	26.75
7	1.179252146	55.3	28.99	0.240569192	54.26	31.44
8	0.589626073	51.65	32.34	0.120284596	46.61	37.38
9	0.294813037	47.48	42.01	0.060142298	42.26	40.44
10	0.147406518	44.17	50.01	0.030071149	39.13	48.1

Kmeans Clusters						
test number	c	accuracy @ 2FP	#FP @ 90% accuracy	γ	accuracy @ 2FP	#FP @ 90% accuracy
baseline	4.717008585	65.19	27.25	0.962276766	66.96	23.29
1	9.43401717	64.17	22.89	1.924553532	65.57	38.15
2	18.86803434	66.26	26.63	3.849107064	62.43	35.68
3	37.73606868	66.96	24.04	7.698214129	57.57	41.01
4	75.47213736	62.09	29.16	15.39642826	54.61	36.44
5	150.9442747	64.52	27.08	30.79285651	45.57	30.91
6	2.358504293	65.04	20.37	0.481138383	63.48	23.41
7	1.179252146	66.96	25.38	0.240569192	63.3	23.27
8	0.589626073	64.35	21.3	0.120284596	63.83	23.87
9	0.294813037	60.35	28.3	0.060142298	58.09	24.94
10	0.147406518	64	22.44	0.030071149	57.22	29.44

SVM parameters adjusted independently by a factor of 2 with each iteration

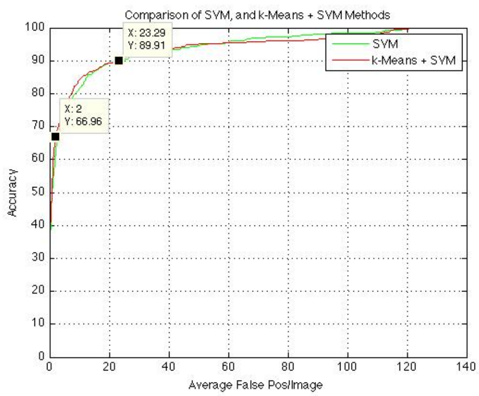
SVM parameters of Kmeans clusters adjusted by a factor of 2 with each iteration

Final Kmeans SVM Classifier						
test number	c	accuracy @ 2FP	#FP @ 90% accuracy	γ	accuracy @ 2FP	#FP @ 90% accuracy
baseline	46.41588834	63.13	26.97	1	63.13	26.97
1	92.83177667	58.43	29.15	2	64.52	27.78
2	185.6635533	58.26	25.61	4	64.17	24.22
3	371.3271067	61.57	23.99	8	35.13	37.1
4	742.6542134	64.87	29.2	16	21.04	28.1
5	1485.308427	59.83	31.7	32	13.39	28.53
6	23.20794417	66.09	22.72	0.5	65.57	19.8
7	11.60397208	67.13	20.03	0.25	65.22	27.14
8	5.801986042	64.7	22.92	0.125	64.17	23.68
9	2.900993021	64.87	23.1	0.0625	60.87	25.44

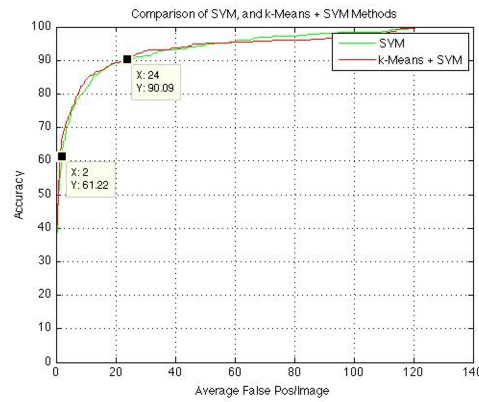
10	1.450496511	66.43	23.22	0.03125	60.35	30

SVM parameters of final Kmeans cluster classifier adjusted by a facot of 2 with each iteration

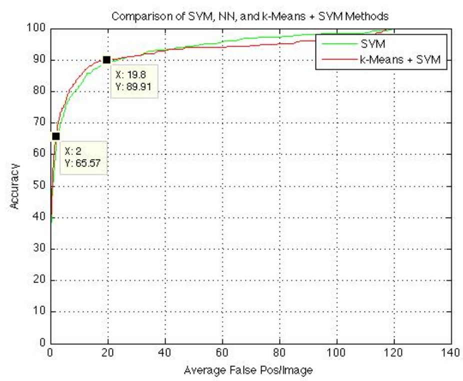
Data was recorded in the form of a FROC curve, including overall accuracy, classification accuracy at 2 false-positives per image, and false-positives per image at 90% accuracy.



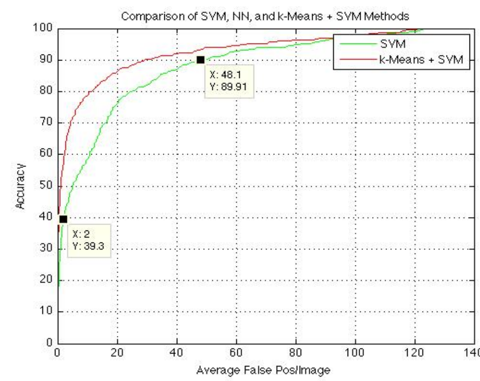
Baseline performance of k-SVM



Baseline performance of SVM



Performance of k-SVM: $\gamma = 0.5$



Performance of SVM: $\gamma = 0.03125$

Results of these tests show that generally an increase in parameter value caused the SVM classifiers to become more accurate (most likely over-fitting), and less accurate when parameters are made smaller. It is clear that kernel parameters must be swept accurately to achieve a good fit to testing data. It is clear also that a grid search is a much better method for parameter selection than selecting arbitrary numbers.

The optimal values of these test results all appear to be relatively small, so at least with this dataset, sweeping to extremely high values seems unnecessary.

6. CONCLUSIONS

SVM, and k-SVM classifiers are fairly robust. They produce consistent results, and are easy to optimize. Optimization of kernel parameters increases performance, and should be done using a two-stage method where search resolution is increased in the region around the best selection of first sweep. When tested multiple times with the same kernel parameters SVM produced identical classification accuracy, while k-SVM results tend to vary. This is possibly due to the method used by k-SVM to cluster data points.

6. FURTHER RESEARCH

Further research is required to test the ability of SVM classifiers to generalize to new data. This research was done using only long-range sonar data. A test of generalization was planned using long-range sonar data in which the targets are oriented differently, and short-range sonar data, in which the target scale is changed. The classifiers would have probably performed well under this scenario change – especially k-SVM, due to its ability to use several classifiers trained on sub-classes of targets.

ACKNOWLEDGEMENTS

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration and was sponsored by the USRP program through NASA, Jet Propulsion Laboratory and California Institute of Technology.

REFERENCES

[1] Dan Z., Jingdong W.2 Fei W., Changshui Z., “Semi-Supervised Classi¯cation with Universum,” State Key Laboratory on Intelligent Technology and Systems 2007.

[2] Qun C., Qingcai C. Xiaolong W., “Scaling Gaussian RBF kernel width to improve SVM Classifications” IEEE 2005.

[3] Lin, T. H., Lu, T., Braun, H., Edens, W., Zhang, Y., Chao, T.H., Assad, C., Huntsberger, T., "Optimization of a multi-stage ATR system for small target identification," SPIE Conference, 2010.

[4] Lu, T., Hughlett, C., Zhou, H., Chao, T.H., Hanan, J., “Neural Network Post-Processing of Grayscale Optical Correlator,” SPIE Conference, Aug. 04, 2005.

[7] Hsu C., Chang C., Lin C., "A Practical Guide to Support Vector Classification," http://www.csie.ntu.edu.tw/~cjlin, 04/15/2010.