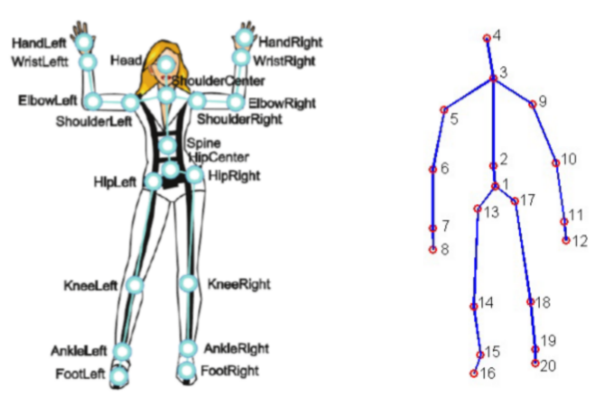
Project report on Skeleton Based Representation and Their Classifications using SVMs

Abhilesh Borode [1]

***Abstract*— MSR daily activity 3D dataset was used in this project for activity classification using support vectors machines. In SVMs the radial basis function kernel was used for classification purposes. Experimental analysis was done with the gamma and C values required by SVM algorithm and corresponding changes on the algorithm accuracy are observed. It was usually observed that the accuracy of the HJPD representation was highest among all three and that of HOD was lowest in comparison. Varying the C-SVM model parameters its was observed that the accuracy of the specific representation was increasing with the increase in the C parameter value. In the specific skeleton representations, the number of bins for the histograms computed were varied and its effect on the accuracy of the final algorithm was studied. It was observed that there was no constant trend line between accuracy and number of bins. Each representation had a different trend line. To conclude the accuracy of the C-SVM model was manly dependent on how good your distribution of the data for each skeleton representation was, that is your decision on selection of number of bins which covers all data points precisely.**

I. INTRODUCTION

The main goal of this project was to build machine learning model using C-SVM models to correctly classify the specific activity label with maximum accuracy possible. Initially we started off with computing the histograms for all the skeleton based representation models.This dataset in this project is a subset of original dataset, which only contains six (6) activity categories: CheerUp (a08) TossPaper (a10) LieOnSofa (a12) Walk (a13) StandUp (a15) SitDown (a16) We considered three main skeleton-based representation models. The first was Relative angles and distances of star skeleton. In this representation we take five joints into considerations, the head, left arm extreme, right arm extreme and the extremes of both the legs. We compute the Euclidean distance between each of these points to the center of the body. Also, the relative angles between the successive vectors joining the center of the body and the extreme joints is also computed. We then compute the histogram for each of these quantities separately while tuning the parameters for the histograms like the number of bins. The frequencies for both the histograms are recorded into a single feature vector for each instance of the RAD data representation file. The length of the final vector comes out to be 5(M+N), Where M is the number of bins for the distance vector and N is the number of bins for the angle vector. The second skeleton-based representation taken into consideration is the histogram of joint position differences (HJPD). In this representation the 3D location of all the joints in Human representation are taken into consideration and its difference is computed from that of the reference joint in the center. The histograms are computed for the differences in all three directions that is x, y and z. Parameter tuning for the number of bins is done for this similar to the RAD representation. Then the computed histograms are computed into a single vector file for each data instance of the HJPD representation. The third representation was the histogram of oriented displacements(HODs) . The HOD representation was an approach to describe the 3D trajectory of each joint separately with respect to itself. This trajectory was from the initial position of the joint to the final position of the same joint. The trajectory computation was carried out by replacing the 3D trajectory of each joint with three 2D trajectories. These three 2D trajectories represent the three projections that joint holds on the orthogonal Cartesian planes (xy, yz and xz). The 3D descriptor was the concatenation of the three 2D trajectories. The computation is continued by the construction of temporal pyramid for the 2D trajectories. Dealing with the trajectory as whole misses the temporal in- formation. In order to capture the temporal evolution, we apply a temporal pyramid approach. In the first level, the whole trajectory is used to construct a part of the descriptor. At the second level, the trajectory is divided into two halves and each one of them is used separately to obtain the second two parts of the descriptor. If the number of levels is 3, the descriptor is made of 7 parts: 1 for the first level, 2 for the second level and 4 for the third. The final descriptor is the concatenation of the seven parts. In other words, a histogram at a specific level will be split into two histograms in the next level.



(a) Names (b) Indices

Fig. 1. Skeleton Representation of Joints

SVMs (Support Vector Machines) is a very useful technique for supervised and unsupervised data classifications.

Any classification technique always involves splitting the data into three sets which are training, cross validating and testing data sets. The goal is to classify each of the datasets to its specific labels with maximum accuracy possible. SVMs works in such a way that it creates a decision boundary of n-1 dimensions, where n are the dimensions of the feature space. The decision boundary tries to classify the data points into their belonging labels. In case of many features the training vectors are mapped into higher dimensional spaces by the kernel function phi. SVM finds a linear separating hyperplane with the maximum margin in this higher dimensional space. We also take into consideration the parameter C which is the penalty parameter to penalize the error terms. SVMs consists of four basic kernels: Linear, Polynomial, Radial basis function and Sigmoid Where gamma, r and d are the kernel parameters. This project uses the RBF kernel.

II. Experiments

The RBF kernel is implemented for the classification of the activities. We implemented the skeleton based representations algorithmically on python. Thus, finding the vectors of histograms containing the distances and relative angles for each data instance file(RAD), finding the vectors of the histograms for all three coordinate displacements from the reference joints for each data instance(HJPD) and similarly for HOD. Once we had proper generated files for these representations we started to format these files into the Libsvm format such that they can be used for classification using Libsvms application programming interface. The formatting of data into Libsvm was also done. After generating the files with Libsvm format implementation of the Libsvms APIs was done like svm_trian was used to train the given model. The parameters supplied for training were $-t2$-c 8 . Where t represents the kernel we want to use, 2 denotes we are using the rbf kernel and c is the penalty parameter which is usually assigned values in the power of two. The values in the Libsvm format files were scaled from -1 to 1 which produced better prediction results. This was done using the script on the terminal window as there is no such interface for python. Then svm_predict API was used in which we feed in the same model which was used for training. The resulting values returned by this function were the accuracy of the model and the predicated label vectors for the testing set. The corresponding grid graphs for each of these representations were also computed. My architecture for the project was like a pipeline. The files of the representation is feed to the code for formatting the data. The formatted files then goes on for the accuracy calculation. Similar experiments were repeated while changing the penalty parameter values and while changing the number of bins while computing the histogram for all three skeleton based representations. However, during all the experiments, the value of t was always taken as 2 thus comparisons were made for the same RBF kernel. For RAD representations the number of bins considered were 10, 11 and 20. The values of C for RAD were taken to be a constant of 8. For HJPD the number of bins taken were 7 and 8. Even for HJPD representation the value of C was taken to be 8. For HOD the number of bins taken were 8 and 10, although for HOD representation the value of C considered was 1024.

III. Results

A. RAD

For RAD representation a total of three experiments were performed and the comparisons among them were made. It was noted that the accuracy for the 1st experiment was 53.06%, for the 2nd experiments was 54.16% and for the final experiment was 52.36%. It was observed that highest accuracy was obtained when the number of bins were 11. The following figures denote the grid graphs and variation of accuracy with number of bins.

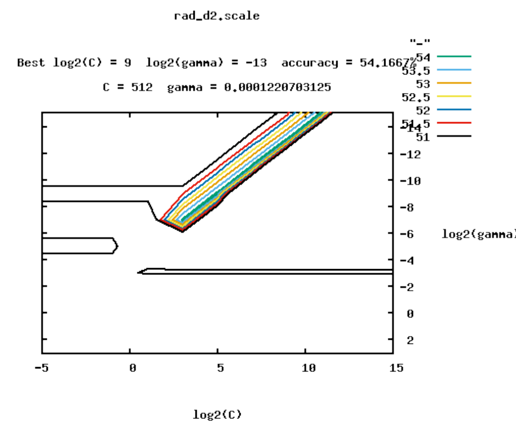


Fig. 2. RAD Grip Graph

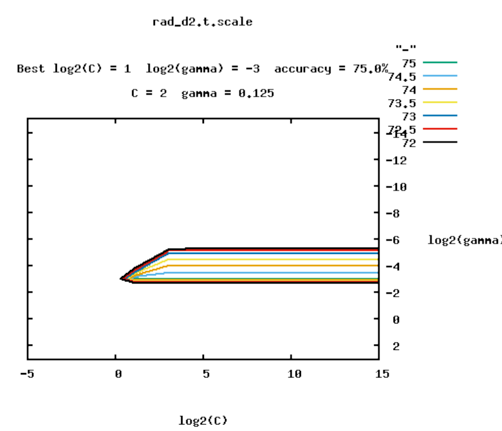


Fig. 3. RAD Grip Graph

B. HJPD

For HJPD representation total of two experiments were performed and comparisons were made between them. When the number of bins were 7 the accuracy of the model

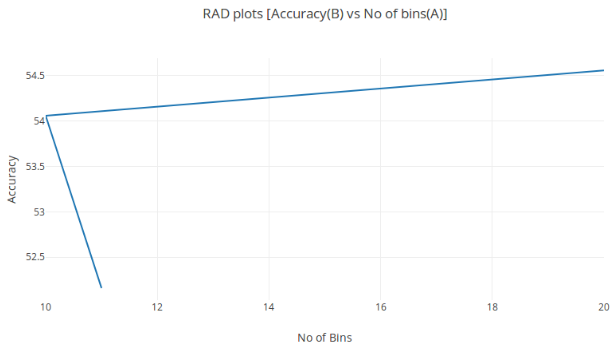


Fig. 4. Number of bins variation with accuracy

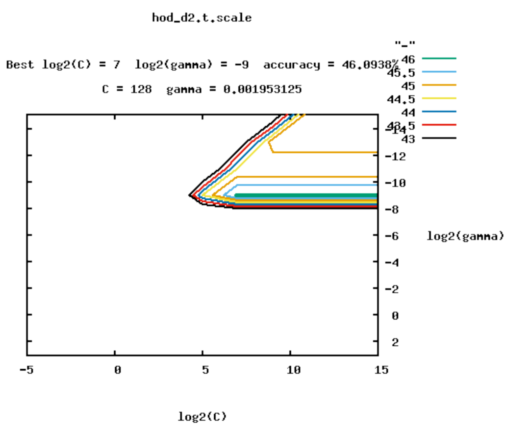


Fig. 5. HOD grid graph

came out to be the highest among all three representations which was 83.33%. In the second trial the number of bins were taken as 8 and the accuracy of the following model dropped to 81.13%. Although the difference in both the trials accuracies were not much the model with 7 bins was proven to be the best model among all others. The following figures denote the grid graphs and variation of the accuracy with the number of bins.

Best Accuracy for the different representations.

RAD = 54.16%

HJPD = 83.333%

HOD = 53.09%

C. HOD

For HOD representations a total of 2 experiments were performed when taking the number of bins as 8 and 10. The accuracy was 52% the highest when the number of bins taken were 8. The following figures denote the grid graphs and variation of the accuracy with the number of bins.

The best values for C and gama are:

RAD: C = 2 gama= 0.125

HJPD: C = 2 gama= 0.00781

HOD: C = 128 gama = 0.00195

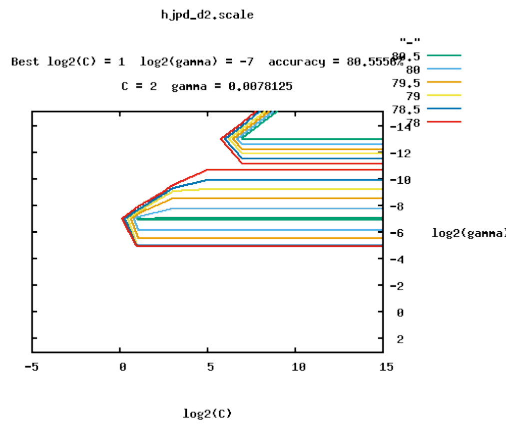


Fig. 6. HJPD grid graph

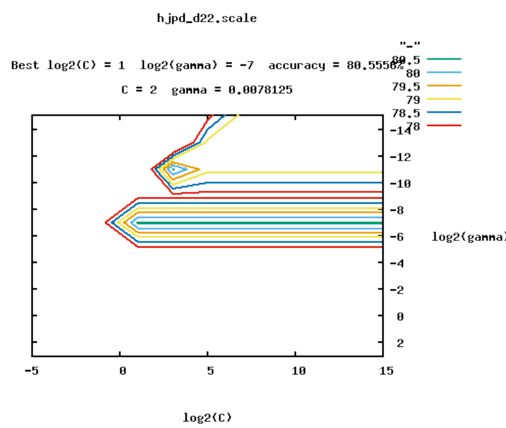


Fig. 7. HJPD grid graph

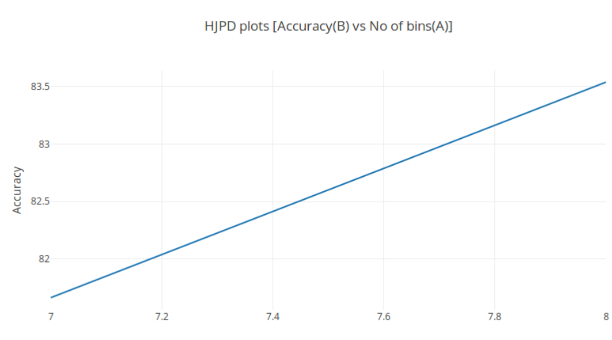


Fig. 8. Number of bins variation with accuracy

		Predicted value					
		a8	a10	a12	a13	a15	a16
Actual value	a8	6	1	0	1	0	0
	a10	2	3	0	0	2	1
	a12	0	1	4	1	0	2
	a13	0	1	1	4	1	1
	a15	0	0	2	1	5	0
	a16	0	2	1	1	0	4

Fig. 9. Confusion Matrix for HOD Representation

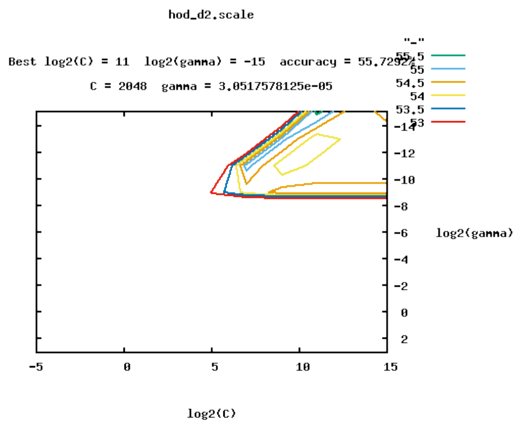


Fig. 10. HOD grid graph

IV. CONCLUSIONS

To conclude among all three representations HJPD was the one who stood out with the maximum accuracy of 83.33%. HODs accuracy was the least among all three. We found out that there is no pattern which tells us how the histogram tuning should be done for achieving the maximum accuracy. We just must visualize the histograms for several bins and then check whether those bins give us the best representation of the data points. The number of bins considered in the histogram computation is the sole dominant parameter that governs the accuracy of the model. You can increase the accuracy of your model to an extent by experimenting with the penalty parameter C, which usually increases the accuracy on increasing the value of C in the power of 2. However, the change in results are note noteworthy. You can also scale you model to a specific range before feeding it into the Libsvms predict which always resulted in an increase in accuracy by around 7%. The implementation of the Libsvm library for classification purposes could be extended to other projects as well which requires classifying the testing dataset and regression problems as well.

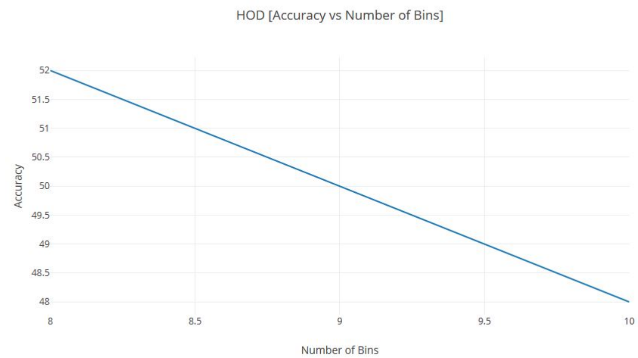


Fig. 11. number of bins variation with accuracy

REFERENCES

[1] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian Real time action recognition using histograms of depth gradients and random decision forests in IEEE Winter Conference on Applications of Computer Vision (WACV), 2014.

[2] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition., in International Joint Conference on Artificial Intelligence (IJCAI), 2013.

[3] C.-C. Chang and C.-J. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, p. 27, 2011.

[4] @articleCC01a, author = Chang, Chih-Chung and Lin, Chih-Jen, title = LIBSVM: A library for support vector machines, journal = ACM Transactions on Intelligent Systems and Technology, volume = 2, issue = 3, year = 2011, pages = 27:1–27:27, note = Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm