CARpLSVM: Cardiac Arrhythmia Recognition using pruned LocalSVM

Muhammad Usman Akram
Faculty of Science
University of Trento, Italy
muhammadusman.akram@studenti.unitn.it

***Abstract*—Local SVM based method is proposed which uses Pruned Fuzzy KNN for pruning and finding local training set, which is later used for training SVM and classifying query point.**

I. Introduction

Cardiac Arrhythmia (also known as Cardiac dysrhythmia or irregular heartbeats) arise as irregularity in function of cardiac muscles, and are indicators of irregularities in hearts electrical activities. Arrhythmia is a very wide range of beats which indicate different conditions or reasons in cardiac functionality. Some of these arrhythmia are indicator of life threatening situations and may result in cardiac arrest. Thus, early and fast recognition of such arrhythmia is imperative in healthcare systems. This, project proceeds with existing research on topic and analyze Support Vector Machines as main tool for recognition of arrhythmia. As, we know special conditions resulting in emergencies occur very few times as compared to normal rhythm of heart. Thus, we need a method which allows control on over fitting on normal beats and performs well on other (life threatening arrhythmia).

Selected beats from MIT-BIH Arrhythmia database [1] were used. Selected Records contained 23,200 beats only, of 6 types of cardiac arrhythmia, Normal (N), Paced Beats (PB), Atrial Premature Beats (APB), Premature Ventricular Contraction (PVC), Left and Right Bundle Branch Blocks (LBBB and RBBB).

Same feature set, as previous work [2] is used. In which features were extracted using tow-level wavelet decomposition of an ECG signal, which has noise reduction without reevaluation of wavelet coefficients. Wavelet decomposition was done using A'trous algorithm proposed by Martinez et al. [3].

II. Arrhythmia Beat Classification

Beat classification is done in various steps, including feature extraction, normalization, feature reduction, data pruning and classification using pruned weighted fuzzy kNN (pfwkNN, previously presented pruning algorithm [4]), support vector machine (SVM) and LocalSVM using combination of pwfkNN and SVM. All steps are discussed in detaill in following sections.

A. Feature Extraction

Feature extration was done using two-level wavelet as in [3] with a' trous' algorithm. As wavelet is derivative of a low pass filter, it offers noise suppression, for details, please refer to [2]. This wavelet is presented by,

$$\Psi(\Omega) = j\Omega \left(\frac{\sin\left(\frac{\Omega}{4}\right)}{\frac{\Omega}{4}} \right)^4 \tag{1}$$

The same wavelet transform is also used for detection of QRS complex. For feature extraction, wavelet coefficients of 64 point window centered at QRS fiducial point, only up to scale 2^2 are used. Following 11 features are extracted from ECG signal:

- Variance of the original QRS complex signal denoted by σ_S^2
- Variance in each sub-band denoted by $\sigma_{A2}^2, \sigma_{D2}^2, \sigma_{D1}^2$
- Variance of the autocorrelation function of wavelet coefficients in each sub-band denoted by $\sigma_{R(A2)}^2, \sigma_{R(D2)}^2, \sigma_{R(D1)}^2$
- Ratio of minimum to maximum wavelet coefficient in each sub-band denoted by r_A2, r_D2, r_D1

These features are combined with the instantaneous RR interval to produce a feature set given by $\left\{\sigma_S^2, \sigma_D^2 1, \sigma_{R(D1)}^2, r_D 1, \sigma_D^2 2, \sigma_{R(D2)}^2, r_D 2, \sigma_A^2 2, \sigma_{R(A2)}^2, r_A 2, RR\right\}$ for a single beat.

B. Normalization

As normalization of extracted data is very important due to the fact of difference in scale in different feature. Thus, we have deployed a Tangent sigmoid function for normalization of al features to standardize at a same level. The normalization function is given bellow,

$$x_{ij}^{'} = tansig\left(\frac{x_{ij} - \bar{x}_j}{\sigma_{x_j}}\right) \tag{2}$$

Where $\bar{x}_j$ and σ_{x_j} are the mean and variance of the j^{th} component of feature vector. This function normalizes features to [-1, 1] scale.

C. Feature Reduction

We have employed two feature reduction techniques Principal Component Analysis (PCA) and Linear Fisher Discrement Analysis (LFDA). With PCA considering the eigen-values of resultant principal components. Only six principal components are selected, which are later used to project 11-dimensional feature space on a six-dimensional space. LFDA is also trained locally on same k as for classification purpose and feature space is reduced to six orthonormalized features.

D. Pruned Fuzzy K-Nearest Neighbor (PFKNN) Classifier

For a training set T and label of point x in T is $C(X)$. PFKNN is explained as,

1) Fuzzy K-Nearest Neighbor Search: Fuzzy KNN is similar to KNN search except for everypoint belonging to a single class, points belong to multiple classes with different membership functions functions associated to classes. Search is described as,

1) Search K nearest neighbor x_j,$j = 1...K$ of given point x using Euclidean distance from set of stored data points (subset of training data) using Fast nearest neighbor search from ATRIA [5].
2) Evaluate membership function values of each class suing,

$$\mu_{c_i}(x) = \frac{\sum_{j=1}^{K} \mu_{c_i}(x_j)\, d_j^{-2/(m-1)}}{\sum_{j=1}^{K} d_j^{-2/(m-1)}} \tag{3}$$

Where
$d_j = \|x - x_j\|$ is Euclidean distance between x and x_j
$\mu_{c_i}(x_j)$ is membership value of point x_j for class c_i. Membership values of all points in stored data, are evaluated for each class, during training as follows,

$$\mu_{c_j}(x_j) = \begin{cases} 0.51 + 0.49\frac{k_i}{K} & \text{if } c(x_p) = c_i \\ 0.49\frac{k_i}{K} & \text{else} \end{cases} \tag{4}$$

Where,
k_i is number of nearest neighbors of x_p that belong to class as x_p.
3) Assign label to the query point x, using,

$$c_o(x) = argmax_i(\mu_{c_i}(x)) \tag{5}$$

2) Pruning Method: Pruning involes reducing data points from traing set T to obtain a much smaller prototype set P. For pruning AF Pruning Algorithm [4]is used, which is as,

1) Starting with training set T and empty prototypical $p = \phi$.
2) Find K nearest neighbors of each point x, such that $c(x_j) \neq c(x)$ and add all such point to prototype set, This will give all the border points of different clusters in data.
3) Now, classify each point in training set using prototype set through fknn (or classifier of choice). If point is misclassified, add it to prototype set and reevaluate class memberships and weights. This will accommodate any outlier cluster which were left in Step 2.

After obtaining pruned set, class weights are calculated to deal with the data imbalance problem (used in pruned weighted fuzzy knn).

E. Support Vector Machine (SVM)

Support Vector Machines belong to group of large margins classifiers. SVM tries to find a hyperplane separating two classes with maximum margin. SVM is the maximal margin hyperplane in feature space build using kernel function in hyper space (or kernel space) [6]. We have used radial biases function (rbf) as kernel for SVM, which is given by,

$$\kappa(p, q) = exp\left(-\gamma\|p - q\|^2\right) \tag{6}$$

Where, γ is width of rbf.

F. Local Support Vector Machine (LocalSVM)

As, SVM needs to lot of optimization and calculations to find right hyperplane separating two classes.The imbalanced of data points used belonging to multiple classes, makes finding generalized boundaries separating classes complex task. SVM has advantage that, though classifier takes time to train but once trained, classifying becomes easy and less time consuming than most prototype base classifiers, but imbalance in data classifier might under or over fit on certain classes. To avoid these problems, Localized SVM were introduced, these classifiers are combination of a prototype based classier (like kNN) and SVM. LocalSVM can be described as,

1) First use kNN to find K nearest neighbors
2) Train SVM on, set of k nearest neighbors (for small k it will take relatively very small amount of time)
3) Use trained SVM to classify query point.

For k approaching infinity or equal to size of prototype set Local SVM will behave like simple SVM and for too small k it will behave in much similar manner to kNN.

G. Classification

Classification involves finding k nearest neighbors from stored prototype data set and then depending on classifier (pfkNN or localSVM) calculation of the membership function values for query point or training a Local SVM (an SVM with training set including k nearest neighbors only) and assigning class label based on SVM classification result.

III. Description of Database

The data set used for testing purpose is subset of MIT-BIH Arrythmia Database. It contains two-channels ambulatory ECG recordings of 47 subjects, gathered by BIH Arrhythmia Laboratory between 1975 and 1979. ECG records were digitized at a sampling frequency of 360 Hz with 11-bit resolution. We used the annotation provided by expert cardiologist which included with database. We are only using beats belonging to six types of Arrhythmia (Normal (N), Paced Beats (PB), Atrial Premature Beat (APB), Premature Ventricular Contraction (PVC), Left and Right Bundle Branch Blocks (LBBB and RBBB)). Number of beats are as shown by plot 1,

IV. Results

Results of classification of six types of beats are presented in following sub-sections.

Classifier		N	LBBB	RBBB	PVC	PAC	PB	A(G)	R
SVM	PPV	99.90 ± 0.02	99.41 ± 0.45	99.61 ± 0.09	98.83 ± 0.53	99.36 ± 0.32	99.87 ± 0.13	99.59 ± 0.14	1
SVM (PCA)	PPV	99.82 ± 0.05	98.87 ± 0.28	99.39 ± 0.33	98.79 ± 0.27	99.05 ± 0.40	99.87 ± 0.22	99.38 ± 0.13	1
	Se	99.83 ± 0.11	98.96 ± 0.16	99.50 ± 0.18	98.65 ± 0.01	98.71 ± 1.30	99.87 ± 0.22	(99.25)	
LSVM	PPV	99.87 ± 0.00	99.17 ± 0.30	99.48 ± 0.25	99.01 ± 0.37	99.53 ± 0.20	100.00 ± 0.00	99.55 ± 0.03	1
	Se	99.89 ± 0.06	99.25 ± 0.22	99.56 ± 0.17	99.05 ± 0.25	99.07 ± 0.18	99.92 ± 0.07	(99.45)	
Pruned LSVM	PPV	99.90 ± 0.02	99.25 ± 0.34	99.50 ± 0.16	98.87 ± 0.71	99.59 ± 0.09	99.96 ± 0.07	99.56 ± 0.15	0.28
	Se	99.89 ± 0.06	99.18 ± 0.39	99.62 ± 0.17	99.34 ± 0.35	99.89 ± 0.10	99.91 ± 0.15	(99.47)	
Pruned LSVM (PCA)	PPV	99.82 ± 0.09	98.87 ± 0.32	99.50 ± 0.14	98.79 ± 0.49	99.35 ± 0.28	99.96 ± 0.07	99.44 ± 0.10	0.28
	Se	99.82 ± 0.17	99.08 ± 0.32	99.35 ± 0.22	99.21 ± 0.46	98.82 ± 0.28	99.79 ± 0.36	(99.35)	
Pruned LSVM (LFDA)	PPV	99.75 ± 0.12	99.11 ± 0.38	99.42 ± 0.20	98.71 ± 0.34	98.77 ± 0.11	99.96 ± 0.07	99.39 ± 0.12	0.27
	Se	99.68 ± 0.11	99.15 ± 0.27	99.40 ± 0.31	99.11 ± 0.59	98.46 ± 0.47	99.92 ± 0.07	(99.28)	
wFKNN	PPV	99.78 ± 0.14	98.87 ± 0.35	99.44 ± 0.15	98.92 ± 0.49	99.47 ± 0.17	99.96 ± 0.07	99.43 ± 0.00	1
	Se	99.92 ± 0.07	99.23 ± 0.19	99.38 ± 0.10	98.87 ± 0.23	98.14 ± 0.27	99.96 ± 0.07	(99.25)	
Pruned wFKNN	PPV	99.90 ± 0.05	98.85 ± 0.43	99.21 ± 0.13	99.09 ± 0.46	99.35 ± 0.19	100.00 ± 0.00	99.43 ± 0.14	0.28
	Se	99.82 ± 0.02	99.26 ± 0.39	99.48 ± 0.10	98.96 ± 0.56	98.20 ± 0.23	99.83 ± 0.08	(99.26)	
Pruned wFKNN (PCA)	PPV	99.83 ± 0.04	98.90 ± 0.34	99.13 ± 0.10	98.79 ± 0.37	99.19 ± 0.50	99.96 ± 0.07	99.35 ± 0.01	0.28
	Se	99.75 ± 0.15	99.12 ± 0.31	99.35 ± 0.11	98.91 ± 0.67	98.28 ± 0.73	99.79 ± 0.36	(99.20)	
Pruned wFKNN (LFDA)	PPV	99.79 ± 0.04	98.82 ± 0.48	98.86 ± 0.17	98.69 ± 0.16	98.51 ± 1.18	99.92 ± 0.15	99.20 ± 0.17	0.27
	Se	99.56 ± 0.13	98.88 ± 0.23	99.09 ± 0.52	98.76 ± 0.50	98.47 ± 0.17	99.92 ± 0.07	(99.11)	

TABLE I
COMPARISON OF CLASSIFIERS AND FEATURE REDUCTION ($N_{train} = 15467, N_{test} = 7733$) WITH $k = 15$ & $\gamma = 2^{-1}$

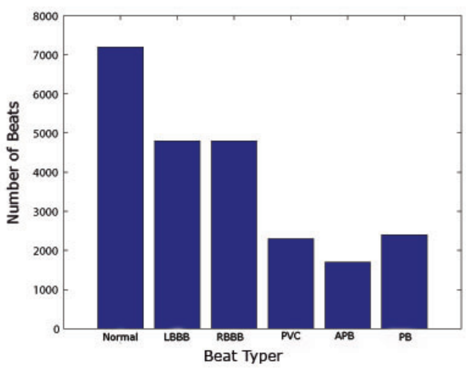

Fig. 1. Distribution of Beats

A. Performance Metrics

Following performance metrics are use to evaluate and compare performance of employed classifiers.

1) Mean & Standard Deviation of Positive Predictive Values (PPV): PPV of each class is calculated over 3 fold cross validation. With TP_c and FP_c representing the number of true and false positives for a given class c. PPV is defined by,

$$PPV_c = \frac{TP_c}{TP_c + FP_c} \tag{7}$$

2) Mean & Standard Deviation of Sensitivity Values (Se): Se of each class is calculated over 3 fold cross validation. If FN_c is number of false negatives for a class c, than its Se is given by,

$$Se_c = \frac{TP_c}{TP_c + FN_c} \tag{8}$$

3) Mean & Standard Deviation of Total Accuracy (A): Total Accuracy of each class is calculated over 3 fold cross validation. It is given by,

$$A = \left(1 - \frac{N_{error}}{N_{test}}\right) \tag{9}$$

Where, N_{error} and N_{test} are number of misclassified query points and total query points respectively.

4) Mean & Standard Deviation of Geometric Mean of Sensitivity values (G): G is calculated over 3 fold cross validation. and is given by,

$$G = \left(\prod_{k=1}^{6} Se_{c_k}\right)^{\frac{1}{6}} \tag{10}$$

For measuring performance of pruning, we use ratio of pruned data to original training data. and is given by,

$$R = \frac{\text{\# of points in set P}}{\text{\# of points in set T}} \tag{11}$$

B. Classification Results

Firstly, results for pruned Local SVM for parameters $k = 15$ and $\gamma = s^{-1}$ are presented and all classifiers were trained over 66.66% data and was tested over remaining 33.33% with 3 fold cross validation with $A = 99.55\%$.SVM performers really well with 99.59% accuracy, while pwfKNN has a accuracy of 99.43%. It is observed that pruning on average leaving 4382 prototype points out of 15467 training points, didn't have any negative effect on classification using Local SVM ($A = 99.56\%$), on the other hand standard deviation of total accuracy of pruned data was much higher in comparison to non pruned data. Table I showes the comparison of employed classifiers.

V. Conclusion

Proposed pruned LocalSVM (using Fuzzy KNN) has a comparable performance to unpruned SVM (but much better performance while comapring training and classification times) and better than pruned weighted FKNN [7]. It offers reduced time complexity and also shows very low performance degradation, if applied to data with noise (with 20 db SNR, total accuracy is 98.37%). It also shows good performance even with when features are reduced to 6 from 11 features, using either PCA or LFDA. Proposed method reduces time and space complexity for both LocalSVM and SVM. Due to incremental nature of our proposed pruning algorithm LocalSVM can incrementally learn even after deployment.

References

[1] G. B. I. Christov, I. Jekova, "Premature ventricular contraction classification by the kth nearest-neighbours rule," *Physiological Measurement*, vol. 26, no. 1, p. 123, 2005.

[2] F. A. Minhas and M. Arif, "Robust electrocardiogram (ecg) beat classification using discrete wavelet transform," *Physiological Measurement*, vol. 29, no. 5, pp. 555–570, 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18427158

[3] J. Martinez, R. Almeida, S. Olmos, A. Rocha, and P. Laguna, "A wavelet-based ecg delineator: evaluation on standard databases," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 4, pp. 570 –581, april 2004.

[4] M. Arif, M. Akram, and F. Afsar, "Arrhythmia beat classification using pruned fuzzy k-nearest neighbor classifier," in *Soft Computing and Pattern Recognition, 2009. SOCPAR '09. International Conference of*, dec. 2009, pp. 37 –42.

[5] C. Merkwirth, U. Parlitz, and W. Lauterborn, "Fast nearest-neighbor searching for nonlinear signal processing," *Physical Review*, pp. 2089–2097, 200.

[6] F. Markowetz, "Classi cation by support vector machines," Practical DNA Microarray Analysis, 2003.

[7] M. U. Akram, "Application of prototype based fuzzy classifiers for ecg based cardiac arrhythmia recognition," PIEAS, Pakistan, 2008.