Detecting Cardiac Abnormalities From 12-lead ECG Signals Using Feature Selection, Feature Extraction, and Machine Learning Classification

Garrett Perkins[1], Chase McGlinn[1], Muhammad Rizwan[2], Bradley M Whitaker[1]

[1]Electrical & Computer Engineering Department, Montana State University, Bozeman, MT, USA
[2]Electrical Engineering Department, University of Management and Technology, Lahore, Pakistan

Abstract

This work represents an entry to the 2020 PhysioNET Computing in Cardiology Challenge for the team named "Whitaker's Lab." The algorithm we developed can be divided into three main components: feature extraction, dimensionality reduction, and classification. In the feature extraction stage, we process the provided 12-lead ECG signals to determine various features. We consider 12 time-domain statistical features per lead, as well as sparse coding features obtained from frequency information that is extracted from each ECG lead. After computing the features, we reduce the dimensionality of the statistical features using principal component analysis in an attempt to ease the computational requirements of the classifier. After feature extraction and dimensionality reduction, we classify each 12-lead ECG signal using a random forest classifier. The classifier is trained using a cross-validated grid search algorithm to help select hyperparameters. In an attempt to avoid overfitting, the classifier and unsupervised feature extraction algorithms are trained on disjoint subsets of the Challenge data. We were unable to rank and score in the test set, but using a holdout portion of the training set we achieved a validation score of -0.744. This result is likely to be over-optimistic.

1. Introduction

This work represents an entry to the 2020 PhysioNet Computing in Cardiology Challenge [1, 2] for the team named "Whitaker's Lab." The goal of the Challenge is to accurately detect several different types of abnormal heart rhythms along with a healthy rhythm. Specific details of the Challenge can be found in [1].

2. Methods

While we were unable to submit a final Challenge entry, we believe we are using two strategies that may be combined with other algorithms to develop a more robust classifier. The first is our use of sparse coding to extract features. We have previously used sparse coding to help detect abnormalities in time series cardiac data, considering both EGC signals and phonocardiogram (PGC) signals [3, 4], and believe it to be an underutilized tool.

The second is our method of data stratification for training the classifier. It is well known that using N-fold cross validation or reserving a holdout set in the learning process can help avoid overfitting [5]. Both strategies effectively split the data into two groups: one used to train the algorithms, and one used to report a validation score. However, our approach uses learning-based algorithms in two capacities. We used two unsupervised learning algorithms, principal components analysis (PCA) and sparse coding, to extract features from the data and we used a supervised learning algorithm, a random forest classifier, as the final classifier. Because of the possibility of overfitting when using the same set of training data to train both types of learning algorithms, we split our data into three groups: one used to train the feature extraction algorithms, one used to train the classifier, and one reserved for validation.

In addition to these two methods, we use fairly common algorithms for ECG analysis and prediction. We use a traditional R-peak detection algorithm [6], PCA, and a random forest classifier [7]. All of these tools have been successfully applied to ECG analysis in previous work [8–10].

Fig. 1 illustrates a block diagram of our algorithm. We extract features from each lead independently. We then combine the features into a single vector that is used in the classification stage, where we use a random forest classifier trained with a cross-validated random search for tuning hyperparameters [11]. The output of the classifier is the predicted disease class associated with the 12-lead ECG signal. The remainder of this section explains the details of how we implemented our algorithm on the 12-lead ECG signals provided by PhysioNet.

2.1. Preprocessing and Data Separation

Prior to performing any machine learning techniques, we first analyzed each 12-lead ECG file and replaced the

ISSN: 2325-887X DOI: 10.22489/CinC.2020.217

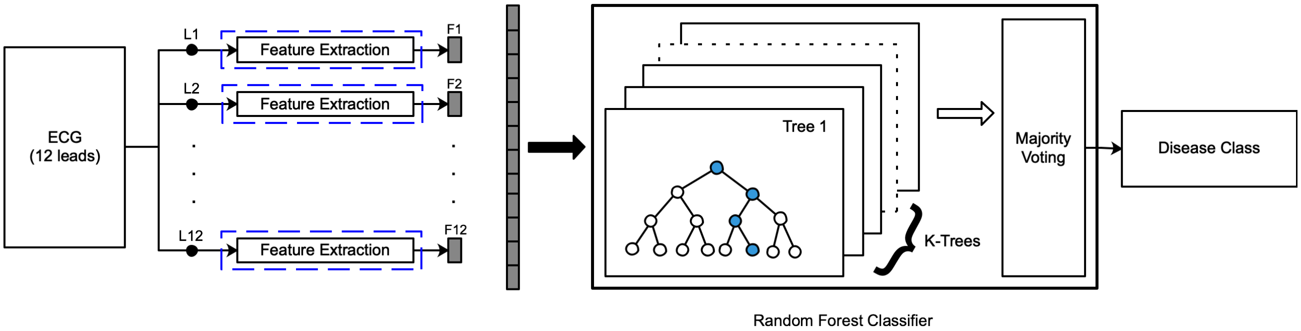

Figure 1. Block diagram outlining our approach. After basic preprocessing, each lead of the 12-lead ECG signal is sent to our feature extraction algorithm. The features are then combined and used as the input to a random forest classifier.

Not-a-Number (NaN) values with zeros. We then split the data into 3 groups: 25% of data was assigned to train the models for feature extraction (PCA and sparse coding), 50% of the data was assigned to train the random forest classifier, and the remaining 25% of the data was reserved as a holdout validation set.

The Challenge data included ECG signals taken from a variety of clinical locations. In order for an ECG classification algorithm to be practical, a classifier trained in one location should be able to generalize to another location. For this reason, we included an entire subset of the Challenge data (the unused portion of the CPSC2018 data) in the holdout set. The remaining ECG signals were randomly assigned to their respective group.

2.2. Feature Extraction

One purpose of feature extraction is to reduce the size of the input data to a smaller dimension. For example, if the data for one patient is a 15-second signal consisting of 12 ECG leads sampled at 300 Hz, then the total dimension of the input signal is 54,000. There are many ways to reduce the dimensionality of this signal: we can use just one of the 12 leads, we can use just the first 10 seconds, or the signal can be downsampled to 100 Hz. Each of these methods (or a combination of all three) would significantly reduce the input dimension. However, the goal of feature extraction is not just to truncate the data. Rather, it is to find a feature set that summarizes the discriminating information that may be present in the original ECG signals, allowing this information to be preserved and used in the classification stage.

Fig. 2 shows a block diagram of the feature extraction approach we used in our algorithm. As can be seen in the figure, our algorithm extracts two different types of features. The first type are PCA features that are trained on statistical values associated with the R-peaks and RR-intervals of the ECG signal. The second type of features are sparse coding features obtained from the frequency-domain representation of each ECG lead.

R-peak Analysis & Statistical Features: In the top branch of our feature extraction algorithm, we first perform a R-peak analysis to find the R-peak values and the RR-interval times associated with the input lead. One R-peak is associated with each heartbeat, and the value of the R-peak corresponds to the strength of the electrical signal produced by the heart to stimulate the heartbeat. The RR-intervals can be interpreted as the time between consecutive heartbeats. We then calculate statistics for the R-peaks and RR-intervals. The statistics we use are mean, median, standard deviation, variance, skewness, and kurtosis, resulting in in 12 statistics per lead. While not explicitly depicted in the figure, the statistics from each lead are concatenated to form a vector of 144 statistical features associated with each patient. We then use PCA to reduce the dimension of this data from 144 to 20.

PCA: PCA is a method of reducing dimensionality while retaining the maximum amount of variance in the data. The goal of PCA is to find the principal components of the data, which are new vectors that are made from linear combinations of the input variables. These new variables are uncorrelated and selected so that most of the information from the input training data is contained within the first. The second component contains the second most amount of information, and so on. The first step to PCA is to standardize the data so the data is all transformed on the same scale. This standardization of the data is done to prevent some input features from dominating over other values, which helps to preserve the information. In PCA, eigenvalue decomposition is performed on the covariance matrix associated with the input training data. The eigenvectors create a set of orthogonal vectors that spans the input feature space. The eigenvalues are associated with the variance for each eigenvector. Thus ranking each eigenvector by its eigenvalue from highest to lowest effectively sorts them in order of significance. The dimensionality reduction is done by choosing how many eigenvectors to

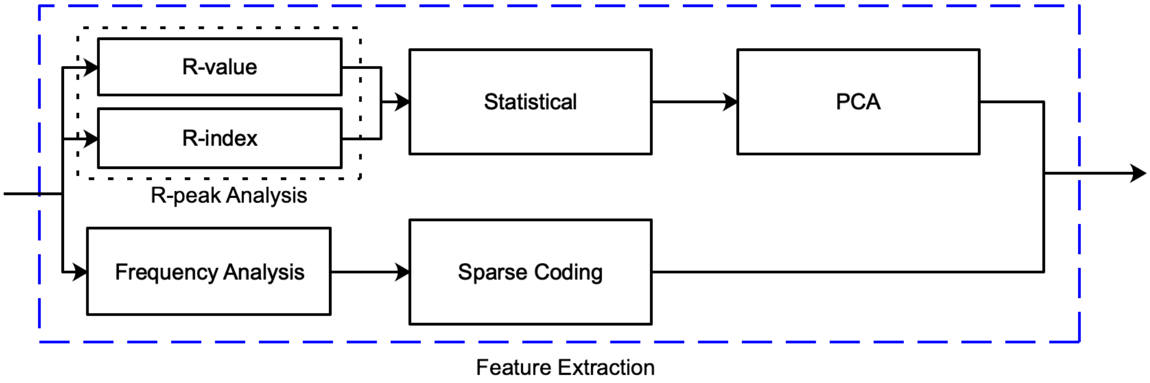

Figure 2. Block diagram of the feature extraction portion of our algorithm. The R-peak values and RR-intervals are detected, and various statistics are collected for each lead. The statistics from all leads are combined, and the dimensionality is reduced using PCA. The ECG signals also undergo a frequency analysis, and the frequency information is encoded as features using sparse coding. The PCA and sparse coding features are then sent to the classifier.

keep. In our implementation, each of the original 144-length vectors is represented (with minimal global error) as a linear combination of 20 principal components. The 20 coefficients become the PCA features associated with that particular signal. Note that the covariance matrix is formed by using only the training data; for new input signals, the 20 features are determined without altering the covariance matrix.

Frequency Analysis: In the bottom branch of our feature extraction algorithm, we perform a frequency analysis on the ECG data using an 8192-point fast Fourier transform (FFT). The length 8192 was chosen because it corresponds to an ECG signal of approximately 30 seconds (rounded to the nearest power of two), and most of the Challenge signals were shorter than this. Longer signals are truncated and shorter signals are zero-padded to create the FFT. After computing the FFT, we consider only the magnitude spectrum, and since the magnitute spectrum for real signals is symmetric, we consider only the first 4096 samples in the frequency domain. The purpose of using the FFT is to line up the input data on a common axis. When analyzing ECG signals in the time domain, the interesting information (QRS complex, R-peak locations, etc.) can occur sporadically throughout the signal. In the frequency domain, the first sample always corresponds to 0 Hz and the 4096th sample always corresponds to 300 Hz. This formats the data nicely for sparse coding, which is a matrix factorization technique.

Sparse Coding: Sparse coding is a learning method that aims to find a sparse representation of the input data from linear combinations of unknown basic elements. These basic elements are called atoms, which make up a dictionary. By definition, the sparse vectors contain a majority of zeros, which helps to speed up computations and reduce the theoretical classification complexity. In our algorithm we choose a dictionary size of 100 features and a sparsity parameter of 5, indicating that 95 of features are zero. A common dictionary is trained using data from all 12 leads from all patients in the feature extraction training set.

Output: The final output produced by our feature extraction algorithm is a vector of length 1220. The first 20 features correspond to the PCA features determined from the statistical analysis the R-peak results. The last 1200 features correspond to 100 sparse coding features for each of the 12 leads. However, it should be noted that only 60 of the 1200 sparse coding features are nonzero.

2.3. Classification

After feature extraction, we used a random forest classifier to determine the class label for each ECG signal. The random forest classifier uses a combination of multiple decision trees in a probabilistic manner. The classification accuracy using the ensemble of decision trees is better than using the individual decision trees due to the majority voting being used to make the final decision regarding the output class. The random forest classifier uses a random selection of features at each decision split to avoid correlation between the decision trees [10]. In order to avoid over-fitting, we used cross-validation to select the hyperparameters. The best parameters obtained using grid search approach for the random forest classifier are to use eight estimators with a maximum depth of two, and to use the 'Gini' criterion for calculating feature importance [12].

3. Results

We were unable to train our model on the full Challenge dataset, so the model we used for validation was trained only on the originally posted Challenge data. Table 1 presents a summary of the results we were able to collect. In the table, the Training column shows the results our classifier obtained on the 25% holdout set from the original data (following the 25%, 50%, 25% data split procedure). The Validation column shows the results obtained from the 25% holdout set from the updated Challenge data. Most metrics decrease, but we are confident that training the model on a more complete dataset would give better results.

As mentioned previously, we were unable to submit a successful model in the Official Challenge phase and therefore we are unable to present testing results. The training and validation scores reported in Table 1 are likely to be over-optimistic. However, we are hopeful that training an algorithm on the full set of training data will improve the classifier.

Table 1. Summary of training and validation results.

Metric	Training	Validation
AUROC	0.476	0.511
AUPRC	0.134	0.054
Accuracy	0.238	219
F-measure	0.078	0.002
F-beta	0.108	0.002
G-beta	0.044	0.001
Challenge Score	0.216	-0.744

4. Conclusion

Because we were unable to submit a working Challenge entry, we cannot in good faith draw meaningful conclusions about the performance of our algorithm. However, we are hopeful that our unique contributions may prove useful in future work. In particular, we have introduced our methods of (1) sparse coding for unsupervised feature extraction and (2) three-way data splitting for training feature extraction and classification algorithms. In future work, we plan to incorporate these two methods with other successful algorithms to investigate possible performance increases in the cross-validation score and develop an algorithm that can be successfully tested on the hidden dataset.

Acknowledgments

Research reported in this publication was supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number P20GM103474. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

The authors also thank the PhysioNet Challenge organizers and the Computing in Cardiology Conference committee for the work they performed to host this competition and conference.

References

[1] Perez Alday EA, Gu A, Shah A, Robichaux C, Wong AI, Liu C, Liu F, Bahrami Rad A, Elola A, Seyedi S, Li Q, Sharma A, Clifford GD, Reyna MA. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. Physiological Measurement ;(In Press).

[2] Goldberger A, Amaral L, Glass L, Hausdorff J, Ivanov P, Mark R, Mietus J, Moody G, Peng C, Stanley H. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation Online 2000;101(23):e215–e220.

[3] Rizwan M, Whitaker BM, Anderson DV. AF detection from ECG recordings using feature selection, sparse coding, and ensemble learning. Physiological Measurement 2018;39(12):124007.

[4] Whitaker BM, Suresha PB, Liu C, Clifford GD, Anderson DV. Combining sparse coding and time-domain features for heart sound classification. Physiological Measurement 2017;38(8):1701.

[5] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Statist Surv 2010;4:40–79.

[6] Pan J, Tompkins WJ. A real-time QRS detection algorithm. IEEE Transactions on Biomedical Engineering 1985;BME-32(3):230–236.

[7] Breiman L. Random forests. Machine learning 2001; 45(1):5–32.

[8] Jambukia SH, Dabhi VK, Prajapati HB. Classification of ECG signals using machine learning techniques: A survey. In 2015 International Conference on Advances in Computer Engineering and Applications. 2015; 714–721.

[9] Castells F, Laguna P, Sörnmo L, Bollmann A, Roig JM. Principal component analysis in ECG signal processing. EURASIP Journal on Advances in Signal Processing December 2007;2007(1):074580. ISSN 1687-6180.

[10] Zabihi M, Rad AB, Katsaggelos AK, Kiranyaz S, Narkilahti S, Gabbouj M. Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier. In 2017 Computing in Cardiology (CinC). 2017; 1–4.

[11] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research February 2012;13:281–305. ISSN 1532-4435.

[12] Menze BH, Kelm BM, Masuch R, Himmelreich U, Bachert P, Petrich W, Hamprecht FA. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinformatics 2009;10(1):213.

Address for correspondence:
Bradley Whitaker
630 Cobleigh Hall
Bozeman, MT 59714, USA
bradley.whitaker1@montana.edu