Constructing Eigenvalue/Coefficient Feature Vectors for Heart Attack Classification

Utkars Jain, B.S.
Department of Bioengineering
University of Pittsburgh
Pittsburgh, Pennsylvania
Email: UTJ1@pitt.edu

Rich Tsui, Ph.D.
Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, Pennsylvania
Email: TSUI2@pitt.edu

***Abstract*—We describe a method of constructing feature vectors from EKG data for use in classical machine learning algorithms. This method uses eigenvalues from transforming the data set into a system of linear differential equations. We find that this method is efficient, easy to implement, and results in accuracy over 90%.**

***Keywords*—EKG, Heart Attack, Machine Learning, Differential Equations, Classification**

I. Introduction

An EKG (or ECG) is a test that delineates the electrical activity of your heart through leads [1]. EKGs are used to diagnose afflictions and disorders of the heart. Eigenvalues can be roughly understood as the distortion caused by a linear system, while eigenvectors can be roughly understood as the orientation of that distortion. We believe we can use eigenvalues and eigenvectors to classify EKGs.

II. The Data

Our data-set consisted of 222 samples: 52 healthy controls, 103 heart attack sufferers, and 67 sufferers of other heart disorders. Each sample contained 10 seconds of activity from 15 leads, sampled at 1000 Hz.

III. The Method

Consider a matrix, A, where the rows (m) describe the of leads (and other predictors), and the columns (n) describe points in time. The immediate goal is to create a system of linear differential equations where all the leads are related to each other, as well as an 'error' function:

$$X_i'(t) = F_i(t) + \sum_{l=1}^{m} \alpha_{i,l} X_l(t)$$

In this case, $X_i'(t)$ represents the behavior of the derivative of the ith predictor (in our case, strictly leads). $F_i(t)$ represents the error function which compensates for the disparity between the derivative and the linear combination of the lead values at t.

In order to create this system, we need to calculate all the α values for each lead. We then create another matrix, A^D, by numerically calculating the derivative using the following equation:

$$A^D(t) = \frac{A(t+2h) + 8A(t+h) - 8A(t-h) - A(t-2h)}{12h}$$

Resulting in a m x $(n-4)$ matrix. Each row of A^D represents the derivative of each time point (except the first and last two).

In order to solve for all the α values, we truncate A by the first and last two columns. We then solve the linear system, $A^T\vec{\alpha}_i = A_i^{D,T}$, where T and i signify the transpose of the i-th row, m times.

$$\begin{bmatrix} A_{1,3} & A_{2,3} & \dots & A_{m,3} \\ A_{1,4} & A_{2,4} & \dots & A_{m,4} \\ \dots & \dots & \dots & \dots \\ A_{1,(n-2)} & A_{2,(n-2)} & \dots & A_{m,(n-2)} \end{bmatrix} \begin{bmatrix} \alpha_{i,1} \\ \alpha_{i,2} \\ \alpha_{i,3} \\ \dots \\ \alpha_{i,m} \end{bmatrix} = \begin{bmatrix} A_{i,3}^D \\ A_{i,4}^D \\ A_{i,5}^D \\ \dots \end{bmatrix}$$

Then we combine and layer α values from each lead to give us an mxm matrix. We proceed to take the eigenvalues of this matrix, to get a vector of eigenvalues: $E = \{\lambda_1, \lambda_2, ..., \lambda_M\}$. We can then fit a polynomial to the the result of:

$$F_i(t) = X_i'(t) - \sum_{l=1}^{m} \alpha_{i,l} X_l(t)$$

The coefficients from the polynomial are then concatenated to the eigenvalue vector. We create a new matrix with these feature vectors, scaling each column (each predictor) from 0 to 1. Now it is ready for input into machine learning algorithms.

IV. Results

Best Performing Algorithms, Ternary Classification w/o poly	
Algorithm	Accuracy
Subspace Discriminant	**82.9 %**
Linear SVM	**81.5 %**
Quadratic SVM	**80.2 %**
Bagged Trees	**79.3 %**

TABLE I. Accuracy w/o polynomial coefficients

Best Performing Algorithms, Ternary Classification w poly	
Algorithm	Accuracy
Bagged Trees	**96.4 %**
RUS Boosted Trees	**91.0 %**
Simple Tree	**88.7 %**
Complex/Medium Tree	**88.3 %**

TABLE II. Accuracy w polynomial coefficients

References

[1] Goldberger et al. PhysioBank. *Circulation*, 101(23):e215–e220, 2000.