Orthogonal Decision Trees for Resource Constrained Physiological Data Stream Monitoring using Mobile Devices

Haimonti Dutta, Hillol Kargupta[*], Anupam Joshi
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County, 1000 Hilltop Circle Baltimore, MD 21250
{hdutta1, hillol, joshi}@cs.umbc.edu

Abstract

Several challenging new applications demand the ability to do data mining on resource constrained devices. One such application is that of monitoring physiological data streams obtained from wearable sensing devices. Such monitoring has applications for pervasive healthcare management, be it for seniors, emergency response personnel, soldiers in the battlefield or atheletes. A key requirement is that the monitoring system be able to run on resouce constrained handheld or wearable devices. Orthogonal decision trees (ODTs) offer an effective way to construct a redundancy-free, accurate, and meaningful representation of large decision-tree-ensembles often created by popular techniques such as Bagging, Boosting, Random Forests and many distributed and data stream mining algorithms. Orthogonal decision trees are functionally orthogonal to each other and they correspond to the principal components of the underlying function space. This paper discusses various properties of the ODTs and their suitability for monitoring physiological data streams in a resource-constrained environment. It offers experimental results to document the performance of orthogonal trees on grounds of accuracy, model complexity, and other characteristics in a resource-constrained mobile environment.

1. Introduction

Analyzing and monitoring time-critical data streams using mobile and wearale devices in a ubiquitous manner is important in many application domains. Online classification of the data streams in such resource constrained environments is a challenging task that requires light-weight classifiers that are accurate but compact in representation. One class of such applications, detailed in the next section, involve the monitoring of physiological data streams obtained from wearable sensors. These applications demand the ability to quickly classify relatively large amount of data. Decision trees (e.g., CART[3], ID3[13], and C4.5 [14]) offer one way to construct rule-based patterns and classifiers from data; the construction techniques are usually fast and scalable; therefore, they may be used for monitoring and mining data streams from ubiquitous devices like PDAs, palmtops, and wearable computers. Ensemble learning techniques are used where single decision trees do not

[*] Also affiliated with AGNIK, LLC, USA.

provide sufficient accuracy. Boosting [5, 4], Bagging[1], Stacking [16], and random forests [2] are some of the well-known ensemble-learning techniques. Many of these techniques often produce large ensembles that combine the outputs of a large number of trees for producing the overall output.

In many time-critical applications such as monitoring data streams [15], particularly for resource constrained environments [7], maintaining a large ensemble and using it for continuous monitoring is computationally challenging. A redundancy free and meaningful compact representation of large ensembles is therefore needed. We have developed[8, 6] a technique to construct redundancy-free decision trees-ensembles by constructing Orthogonal Decision Trees (ODTs). The technique first constructs an algebraic representation of trees using multi-variate discrete Fourier bases. The new representation is then used for eigen-analysis of the covariance matrix generated by the decision trees in Fourier representation. The proposed approach converts the corresponding principal components to decision trees using a technique reported elsewhere [7]. These trees are functionally orthogonal to each other and they span the underlying function space. These orthogonal trees are in turn used for accurate (in many cases with improved accuracy) and redundancy-free (in the sense of orthogonal basis set) compact representation of large ensembles. We use this compact orthogonal decision tree ensemble to implement a system for monitoring physiological health data streams that can run on resource constrained PDA/wearable devices.

The rest of the paper is organized as follows: Section 2 discusses the importance of monitoring physiological data streams using wearable devices and armbands available in the market. Section 3 presents the underlying theory for representation of decision trees using their Fourier spectra. Section 4 describes the process of removing redundancy from decision tree ensembles and Section 5 explains the method of construction of orthogonal decision trees. Sections 6 and 7 present experimental results for ODTs and compares it with a well known ensemble learning technique. Finally, Section 8 concludes this paper.

2 Physiological Data Stream Monitoring

We draw two scenarios to illustrate the potential uses of the orthogonal decision trees. Both cases involve a situation where a potentially complex decision space has to be examined, and yet the resources available on the devices that will run the decision process are not sufficient to maintain and use ensembles.

Consider a real time environment to monitor the health effects of environmental toxins or disease pathogens on humans. There are significant advances being made today in biochemical engineering to create extremely low cost sensors for various toxins[9] that could constantly monitor the environment and generate data streams over wireles networks. It is not unreasonable to assume that similar sensors could be developed to detect disease causing pathogens. In addition, most state health/environmental agencies and the federal government entities such as CDC and EPA have mobile labs and response units that can test for the presence of pathogens or dangerous chemicals. The mobile units will have handheld devices with wireless connections on which to send the data and/or their analysis. In addition, each hospital today generates reports on admissions and discharges, and often reports that to various monitoring agencies. Given these disparate data streams, one could analyze them to see if correlates can be found, alerting experts to potential cause-effect relations (Pfiesteria found in Chesapeake Bay and hostpitals report many people with upset stomach who had seafood recently), potential epedemiological events (field units report dead infected birds and elederly patients check in with viral fever symptoms, indicating tests needed

Figure 1. The Body Media SenseWear armband and The Vivometrics Life Shirt Garment

for west nile virus and preventive spraying), and more pertinent in present times, low grade chemical and biological attacks (sensors detect particular toxins, mobile units find contaminated sites, hospitals show people who work at or near the sites being admitted with unexplained symptoms). At present, much of this analysis is done “post facto” – experts hypothesize on possible causes of ailments, then gather the data from disparate sources to confirm their hypotheses. Clearly, a more proactive environment which could mine these diverse data strems to detect emergent patters would be extremely useful. This scenario, of course, has some futuristic elements.

On a more present day note, there are now several wearable sensors on the market such as SenseWear armband from BodyMedia[1] , Wearable West[2], and LifeShirt Garment from Vivometrics[3] that can be used to monitor vital signs for a person such as temperature, heartrate, heatflux, SpO_2 etc.

The figure 1[4] on the left hand side shows the SenseWear armband that was used to collect the data. The sensors in this band were capable of measuring the following:

1. Heat flux: The amount of heat dissipated by the body.
2. Accelerometer: Motion of the body
3. Galvanic Skin Response: Electrical conductivity between two points on the wearer’s arm
4. Skin Temperature: Temperature of the skin and is generally reflective of the body’s core temperature
5. Near-Body Temperature: Air temperature immediately around the wearer’s armband.

The subjects were expected to wear the armband as they went about their daily routine, and were required to timestamp the beginning and end of an activity. For example, before starting to take a jog, they could press the timestamp button, and when finished, they could press the button again to record the end of the activity. This body monitoring device can be worn continuously, and can store up to 5 days of physiological data before it had to be retrieved. The LifeShirt Garment is yet another example of an easy to wear shirt, that allows measurement of pulmonary functions via sensors woven into the shirt. The figure 1 on the right hand side shows the heart monitor. Subjects are capable of measuring symptoms, moods, activities and several other physiological characteristics.

[1] http://www.bodymedia.com/index.jsp

[2] http://www.smartextiles.info

[3] http://www.vivometrics.com

[4] The figures are obtained from http://www.cs.utexas.edu/users/sherstov/pdmc/ and http://www.vivometrics.com

Analysing these vital signs in real time using small form factor wearable computers has several valuable near term applications. For instance, one could monitor senior citizens living in assisted or independent housing, to alert physicians and support personnel if the signs point to distress. Similarly, one could monitor athletes during games or practice. Given the recent high profle deaths of athletes both at the professional and high school levels during practice, the importance of such an application is fairly apparent. Other potential applications include battlefield monitoring of soldiers, or monitoring first responders such as firefighters.

The paper offers a method for on line monitoring of physiological data using wearable or handheld (PDAs, cellphones) devices. Data streams are sent to them from sensors using short range wireless networks such as PANs. Precomputed(based on training data obtained previously) orthogonal decision trees and bagging ensembles are kept on these devices. The data streams are classified using these precomputed models, which are updated on a periodic basis. It must be noted that while the monitoring is in real time, the model computation is done off-line using stored data.

3. Fourier Transform of Decision Trees

Decision tree (e.g., CART[3], ID3[13], and C4.5 [14]) ensembles are widely used for classification and other related applications. Ensemble classifiers generate the output by combining the outputs of several base classifiers that define the ensemble. The ensemble approach often produce higher classification accuracy compared to the individual base classifiers.

Large ensembles perform well in terms of accuracy. However, they are often difficult to understand and transform into actionable knowledge. Ensembles are sometimes also redundant. Therefore, it is important to construct a redudancy-free and simple representation of such large ensembles that can be effectively used.

The rest of this paper exploits a linear algebraic representation of the trees in order to be able to construct compact, redundancy-free orthogonal decision trees ([8], [6])that are in turn used for representing the ensemble. This paper adopts multi-variate discrete Fourier representation [7] for various reasons discussed later. The following section reviews this material.

This section briefly discusses the background material [7] necessary for the development of the proposed technique to construct orthogonal decision trees. The proposed approach makes use of linear algebraic representation of the trees. In order to do that that we first need to convert the trees into a numeric tree just in case the attributes are symbolic. This can be done by simply using a codebook that replaces the symbols with numeric values in a consistent manner. Since the proposed approach of constructing orthogonal trees uses this representation as an intermediate stage and eventually the physical tree is converted back, the exact scheme for replacing the symbols (if any) does not matter as long as it is consistent.

Once the tree is converted to a discrete numeric function, we can also apply any appropriate analytical transformation if necessary. Fourier transformation is one such interesting possibility. Fourier bases are orthogonal functions that can be used to represent any discrete function. Consider the set of all ℓ-dimensional feature vectors where the i-th feature can take λ_i different categorical values. The Fourier basis set that spans this space is comprised of $\Pi_{i=0}^{\ell}\lambda_i$ basis functions. Each Fourier basis function is defined as,

$$\psi_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x}) = \frac{1}{\sqrt{\Pi_{i=1}^{l}\lambda_i}}\Pi_{m=1}^{l}\exp^{\frac{2\pi i}{\lambda_m}x_m j_m}$$

where $\mathbf{j}$ and $\mathbf{x}$ are strings of length ℓ; x_m and j_m are m-th attribute-value in $\mathbf{x}$ and $\mathbf{j}$, respectively; $x_m, j_m \in \{0, 1, \cdots \lambda_i\}$ and $\overline{\lambda}$ represents the feature-cardinality vector, $\lambda_0, \cdots \lambda_\ell$; $\psi_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x})$ is called the $\mathbf{j}$-th basis function. The vector $\mathbf{j}$ is called a *partition*, and the *order* of a partition $\mathbf{j}$ is the number of non-zero feature values it contains. A Fourier basis function depends on some x_i only when the corresponding $j_i \neq 0$. If a partition $\mathbf{j}$ has exactly α number of non-zeros values, then we say the partition is of order α since the corresponding Fourier basis function depends only on those α number of variables that take non-zero values in the partition $\mathbf{j}$.

A function $f : \mathbf{X}^\ell \to \Re$, that maps an ℓ -dimensional discrete domain to a real-valued range, can be represented using the Fourier basis functions: $f(\mathbf{x}) = \sum_{\mathbf{j}} w_{\mathbf{j}}\overline{\psi}_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x})$. where $w_{\mathbf{j}}$ is the Fourier Coefficient (FC) corresponding to the partition $\mathbf{j}$ and $\overline{\psi}_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x})$ is the complex conjugate of $\psi_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x})$; $w_{\mathbf{j}} = \sum_{\mathbf{x}} \psi_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x}) f(\mathbf{x})$. The *order* of a Fourier coefficient is nothing but the order of the corresponding partition. We shall often use terms like *high order* or *low order* coefficients to refer to a set of Fourier coefficients whose orders are relatively large or small respectively. Energy of a spectrum is defined by the summation $\sum_{\mathbf{j}} w_{\mathbf{j}}^2$. Let us also define the inner product between two spectra $\mathbf{w}_{(1)}$ and $\mathbf{w}_{(2)}$ where $\mathbf{w}_{(i)} = [w_{(i),1} w_{(i),2}, \cdots w_{(i),|J|}]^T$ is the column matrix of all Fourier coefficients in an arbitrary but fixed order. Superscript T denotes the transpose operation and $|J|$ denotes the total number of coefficients in the spectrum. The inner product, $< \mathbf{w}_{(1)}, \mathbf{w}_{(2)} >= \sum_{\mathbf{j}} w_{(1),\mathbf{j}} w_{(2),\mathbf{j}}$. We will also use the definition of the inner product between a pair of real-valued functions defined over some domain Ω. This is defined as $< f_1(\mathbf{x}), f_2(\mathbf{x}) >= \sum_{\mathbf{x}\in\Omega} f_1(\mathbf{x}) f_2(\mathbf{x})$.

Fourier transformations of bounded-depth decision trees have several properties that makes it an efficient one. More details can be found elsewhere [10, 12].

3.1. Properties of Decision Trees in the Fourier Domain

This section considers the Fourier spectrum of decision trees with finite depths, bounded by some constant. The underlying functions in such decision trees can be represented by a constant depth Boolean AND and OR circuit (or equivalently AC^0 circuit). Linial et al. [10] noted that the Fourier spectrum of AC^0 circuit has very interesting properties and proved the following lemma.

Lemma 1 ***(Linial, 1993)*** *Let M and d be the size and depth of an AC^0 circuit. Then*

$$\sum_{\{\mathbf{j} \mid o(\mathbf{j})>t\}} w_{\mathbf{j}}^2 \leq 2M2^{-t^{1/d}/20}$$

where $o(\mathbf{j})$ denotes the order of the partition $\mathbf{j}$ and t is a non-negative integer. The term on the left hand side of the inequality represents the energy of the spectrum captured by the coefficients with order greater than a given constant t. The energy captured by all high order Fourier coefficients is small. This is because the energy of the Fourier coefficients of higher order decays exponentially. This observation suggests that the spectrum of a Boolean decision tree (or equivalently bounded depth function) can be approximated by computing only a small number of low order Fourier coefficients. So Fourier basis offers an efficient numeric representation of a decision tree in the form of a linear function that can be easily stored and manipulated. The exponential decay property of Fourier spectrum also holds for non-Boolean decision trees. The complete proof is available elsewhere [12].

Let us also note that,

1. the Fourier spectrum of a decision tree can be efficiently computed [7] and

2. the Fourier spectrum can be directly used for constructing the tree [12].

In other words, we can go back and forth between the tree and its spectrum. This is philosophically similar to the switching between the time and frequency domains in the traditional application of Fourier analysis for signal processing.

Fourier transformation of decision trees also preserves inner product. The functional behavior of a decision tree is defined by the class labels it assigns. Therefore, if $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_{|\Omega|}\}$ are the members of the domain Ω then the functional behavior of a decision tree $f(\mathbf{x})$ can be captured by the vector $[f]_{x \in \Omega} = [f(\mathbf{x}_1) f(\mathbf{x}_2) \cdots f(\mathbf{x}_{|\Omega|})]^T$, where the superscript T denotes the transpose operation. The following section describes a Fourier analysis-based technique for constructing redundancy-free orthogonal representation of ensembles.

The following lemma proves that the inner product between two such vectors is identical to the same in between their respective Fourier spectra.

Lemma 2 *Given two functions* $f_1(\mathbf{x}) = \sum_{\mathbf{j}} w_{(1),\mathbf{j}} \overline{\psi_{\mathbf{j}}}^{\lambda}(\mathbf{x})$ *and* $f_2(\mathbf{x}) = \sum_{\mathbf{j}} w_{(2),\mathbf{j}} \overline{\psi_{\mathbf{j}}}^{\lambda}(\mathbf{x})$ *in Fourier representation. Then* $< f_1(\mathbf{x}), f_2(\mathbf{x}) > = < \mathbf{w}_{(1)}, \mathbf{w}_{(2)} >$.

Proof:

$$\begin{aligned}
< f_1(\mathbf{x}), f_2(\mathbf{x}) > &= \sum_{\mathbf{x} \in \Omega} f_1(\mathbf{x}) f_2(\mathbf{x}) = \sum_{\mathbf{x} \in \Omega} \sum_{\mathbf{j},\mathbf{i}} w_{(1),\mathbf{j}} \overline{\psi_{\mathbf{j}}}^{\lambda}(\mathbf{x}) w_{(2),\mathbf{i}} \overline{\psi_{\mathbf{i}}}^{\lambda}(\mathbf{x}) \\
&= \sum_{\mathbf{j},\mathbf{i}} w_{(1),\mathbf{j}} w_{(2),\mathbf{i}} \sum_{\mathbf{x} \in \Omega} \overline{\psi_{\mathbf{j}}}^{\lambda}(\mathbf{x}) \overline{\psi_{\mathbf{i}}}^{\lambda}(\mathbf{x}) = \sum_{\mathbf{j}} w_{(1),\mathbf{j}} w_{(2),\mathbf{j}} = < \mathbf{w}_{(1)}, \mathbf{w}_{(2)} > .
\end{aligned}$$

■

The fourth step is true since Fourier basis functions are orthonormal. The following section presents a way to use this representation for constructing orthogonal decision trees.

4 Removing Redundancies from Ensembles

Existing ensemble-learning techniques work by combining (usually a linear combination) the output of the base classifiers. They do not structurally combine the classifiers themselves. As a result they often share a lot of redundancies. The Fourier representation offers a unique way to fundamentally aggregate the trees and perform further analysis to construct an efficient representation.

Let $f_e(\mathbf{x})$ be the underlying function representing the ensemble of m different decision trees where the output is a weighted linear combination of the outputs of the base classifiers. Then we can write,

$$f_e(\mathbf{x}) \quad = \quad \alpha_1\tau_{(1)}(\mathbf{x}) + \alpha_2\tau_{(2)}(\mathbf{x}) + \cdots + \alpha_m\tau_{(m)}(\mathbf{x}) = \alpha_1 \sum_{j\in\mathcal{J}_1} w_{(1),\mathbf{j}}\overline{\psi}_{\mathbf{j}}^{\lambda}(\mathbf{x}) + \cdots + \alpha_m \sum_{\mathbf{j}\in\mathcal{J}_m} w_{(m),\mathbf{j}}\overline{\psi}_{\mathbf{j}}^{\lambda}(\mathbf{x}).$$

Where α_i is the weight of the i^{th} decision tree and Z_i is the set of all partitions with non-zero Fourier coefficients in its spectrum. Therefore, $f_e(\mathbf{x}) = \sum_{j\in\mathcal{J}} w_{(e),\mathbf{j}}\overline{\psi}_{\mathbf{j}}^{\lambda}(\mathbf{x})$, where $w_{(e),\mathbf{j}} = \sum_{i=1}^{m} \alpha_i w_{(i),\mathbf{j}}$ and $\mathcal{J} = \cup_{i=1}^{m}\mathcal{J}_i$. Therefore, the Fourier spectrum of $f_e(\mathbf{x})$ (a linear ensemble classifier) is simply the weighted sum of the spectra of the member trees.

Consider the matrix D where $D_{i,j} = \tau_{(j)}(\mathbf{x_i})$, where $\tau_{(j)}(\mathbf{x_i})$ is the output of the tree $\tau_{(j)}$ for input $\mathbf{x_i} \in \Omega$. D is an $|\Omega| \times m$ matrix where $|\Omega|$ is the size of the input domain and m is the total number of trees in the ensemble.

An ensemble classifier that combines the outputs of the base classifiers can be viewed as a function defined over the set of all rows in D. If $D_{*,j}$ denotes the j-th column matrix of D then the ensemble classifier can be viewed as a function of $D_{*,1}, D_{*,2}, \cdots D_{*,m}$. When the ensemble classifier is a linear combination of the outputs of the base classifiers we have $F = \alpha_1 D_{*,1} + \alpha_2 D_{*,2} + \cdots \alpha_m D_{*,m}$, where F is the column matrix of the overall ensemble-output. Since the base classifiers may have redundancy, we would like to construct a compact low-dimensional representation of the matrix D . However, explicit construction and manipulation of the matrix D is difficult, since most practical applications deal with a very large domain. We can try to construct an approximation of D using only the available training data. One such approximation of D and its Principal Component Analysis-based projection is reported elsewhere [11]. Their technique performs PCA of the matrix D, projects the data in the representation defined by the eigenvectors of the covariance matrix of D, and then performs linear regression for computing the coefficients $\alpha_1, \alpha_2, \cdots$, and α_m.

While the approach is interesting, it has a serious limitation. First of all, the construction of an approximation of D even for the training data is computationally prohibiting for most large scale data mining applications. Moreover, this is an approximation since the matrix is computed only over the observed data set of the entire domain. In the following we demonstrate a novel way to perform a PCA of the matrix D, defined over the entire domain. The approach uses the Fourier spectra of the trees, Lemma 2, and works without explicitly generating the matrix D. It is important to note that the PCA-based regression scheme [11] offers a way to find the weightage for the members of the ensemble. It does not offer any way to aggregate the tree structures and construct a new representation of the ensemble which the current approach does.

The following analysis will assume that the columns of the matrix D are mean-zero. This restriction can be easily removed with a simple extension of the analysis. Note that the covariance of the matrix D is D^TD. Let us denote this covariance matrix by C. The (i,j)-th entry of the matrix,

$$C_{i,j} \quad = \quad < D(*,i), D(*,j) > = < \tau_{(i)}(\mathbf{x}), \tau_{(j)}(\mathbf{x}) > = \sum_{\mathbf{p}} w_{(i),\mathbf{p}} w_{(j),\mathbf{p}} = < \mathbf{w}_{(i)}, \mathbf{w}_{(j)} > \qquad (1)$$

The third step is true by Lemma 2. Now let us the consider the matrix W where $W_{i,j} = w_{(j),(i)}$, i.e. the coefficient corresponding to the i-th member of the partition set $\mathcal{J}$ from the spectrum of the tree $\tau_{(j)}$. Equation 1 implies that the covariance matrices of D and W are identical. Note that W is an $|\mathcal{J}| \times m$ dimensional matrix. For most practical applications

$|\mathcal{J}| << |\Omega|$. Therefore analyzing W using techniques like PCA is significantly easier. The following discourse outlines a PCA-based approach.

PCA of the matrix W produces a set of eigenvectors which in turn defines a set of Principal Components, $V_1, V_2, \cdots V_k$. Let $\gamma_{(j),q}$ be the j-th component of the q-th eigenvector of the matrix $W^T W$.

$$V_q \;=\; \sum_{j=1}^{n} \gamma_{(j),q} D(*,j) = \left[\sum_{j=1}^{n} \gamma_{(j),q} \tau_{(j)}(\mathbf{x})\right]_{\mathbf{x}\in\Omega} = \left[\sum_{j=1}^{n} \gamma_{(j),q} \sum_{\mathbf{i}} w_{(j),\mathbf{i}} \overline{\psi}_{\mathbf{i}}^{\overline{\lambda}}(\mathbf{x})\right]_{\mathbf{x}\in\Omega} = \left[\sum_{\mathbf{i}} a_{\mathbf{i},q} \overline{\psi}_{\mathbf{i}}^{\overline{\lambda}}(\mathbf{x})\right]_{\mathbf{x}\in\Omega} .$$

Where $a_{\mathbf{i},q} = \sum_{j=1}^{n} \gamma_{(j),q} w_{(j),\mathbf{i}}$. The eigenvalue decomposition constructs a new representation of the underlying domain where the feature corresponding to column vector V_q is $v_q = \sum_{\mathbf{i}} a_{\mathbf{i},q} \overline{\psi}_{\mathbf{j}}^{\overline{\lambda}}(\mathbf{x})$ i.e., $V_q = [v_q]_{\mathbf{x}\in\Omega}$. Note that v_q is a linear combination of a set of Fourier spectra and therefore it is also a Fourier spectrum. Also note that V_q-s are orthogonal which is proved in the following.

The inner product between V_q and V_r for $q \neq r$ is,

$$< V_q, V_r > \;=\; < [v_q]_{\mathbf{x}}, [v_r]_{\mathbf{x}} >= \sum_{\mathbf{i},\mathbf{j}} a_{\mathbf{i},q} a_{\mathbf{j},r} \sum_{\mathbf{x}} \psi_{\mathbf{i}}(\mathbf{x}) \psi_{\mathbf{j}}(\mathbf{x}) = \sum_{\mathbf{i}} a_{\mathbf{i},q} a_{\mathbf{i},r} =< \mathbf{a}_q, \mathbf{a}_r >= 0.$$

Therefore, we conclude that the spectra corresponding to the orthonormal basis vectors V_q and V_r are themselves orthonormal. Let f_q and f_r be the functions corresponding to the spectra $\mathbf{a}_q$ and $\mathbf{a}_r$. In other words, $f_q(\mathbf{x}) = \sum_{\mathbf{i}} a_{\mathbf{i},q} \psi_{\mathbf{i}}(\mathbf{x})$ and $f_r(\mathbf{x}) = \sum_{\mathbf{i}} a_{\mathbf{i},r} \psi_{\mathbf{i}}(\mathbf{x})$. Then by Lemma 2 we can also conclude that, $< V_q, V_r >=< \mathbf{a}_q, \mathbf{a}_r >=< f_q(\mathbf{x}), f_r(\mathbf{x}) >$. This implies that the inner product between the output vectors of the corresponding functions are also orthonormal to each other. The following section defines orthogonal decision trees that makes use of these principal components.

4.1 Orthogonal Decision Trees

The analysis presented in the previous sections offers a way to construct the Fourier spectra of a set of functions that are orthogonal to each other and therefore redundancy-free. These functions also define a basis and can be used to represent any given decision tree in the ensemble in the form of a linear combination. Orthogonal decision trees can be defined as an immediate extension of this framework.

A pair of decision trees $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are orthogonal to each other if and only if $< f_a(\mathbf{x}), f_b(\mathbf{x}) >= 0$ when $a \neq b$ and $< f_a(\mathbf{x}), f_b(\mathbf{x}) >= 1$ otherwise. The second condition is actually a slightly special case of orthogonal functions—orthonormal condition. A set of trees are pairwise orthogonal if every possible pair of members of this set satisfy the orthogonality condition.

The principal components $V_1, V_2, \cdots V_k$ computed using the eigenvectors of the covariance matrix C are orthogonal to each other themselves. Since each of these principal components is a Fourier spectrum in itself we can always construct a decision tree from this spectrum using technique noted in Section 5. Although the tree looks physically different from the Fourier spectrum, they are functionally identical. Therefore, the trees constructed from the principal components $V_1, V_2, \cdots V_k$ also maintain the orthogonality condition. These orthogonal trees now can be used to represent the entire ensemble in a very compact and efficient manner. The following section reports some experimental results.

The orthogonality condition guarantees that the representation is not redundant. These orthogonal trees form a basis set that spans the entire function space of the ensemble. The overall output of the ensemble is computed from the output of these orthogonal trees. Specific details of the ensemble output computation depends on the adopted technique to compute the overall output of the original ensemble. However, for most popular cases considered here boils down to computing the average output. If we choose to go for weighted averages, we may also compute the coefficients corresponding to each V_q by simply performing linear regression.

The next section reports experimental results for orthogonal decision trees.

5. Experimental Results

This section first illustrates the performance of orthogonal decision trees on a physiological data set. It demonstrates the construction of ODTs using four C4.5 trees and reports the structure of an orthogonal decision tree obtained from projecting the trees along the first principle component. In resource constrained environments building a large number of trees is usually not an option. Our experiments show that aggregated orthogonal decision trees have comparable accuracy to that of large bagging ensembles. Therefore, aggregated ODT may be a good solution for classification problems on pdas, pocket pcs or cell phones. Finally, the section describes an application on a pocket pc, which can be used to keep track of the physiological conditions of people exposed to hazardous environments, such as fire-fighters trying to douse a fire, soldiers exposed to chemical or biological warfare, and disaster rescue and response workers.

6 Physiological Data Monitoring

This section documents the performance of orthogonal decision trees on a physiological data set. It makes use of publicly available data set in order to offer benchmarked results. This dataset[5] was obtained from the Physiological Data Modeling Contest[6] held as part of the International Conference on Machine Learning, 2004. It comprises of several months of data from more than a dozen subjects and was collected using BodyMedia[7] wearable body monitors.

In our experiments, the training set consisted of 50,000 instances and 11 continuous and discrete valued attributes[8]. The test set had 32,673 instances. The continous valued attributes were discretized using the WEKA software[9]. The final training and test data sets had all discrete valued attributes. A binary classification problem was formulated, which monitored whether an individual was engaged in a particular activity(class label=1) or not(class label=0) depending on the physiological sensor readings.

C4.5 decision trees were built on data blocks of size 150 instances and the classification accuracy and tree complexity was noted. These were then used to compute their Fourier spectra and the matrix of the Fourier coefficients was subjected

[5] Obtained from http://www.cs.utexas.edu/users/sherstov/pdmc/

[6] http://www.cs.utexas.edu/users/sherstov/pdmc/

[7] http://www.bodymedia.com/index.jsp

[8] The attributes used for the classification experiments were gender, galvanic skin temperature, heat flux, near body temperature, pedometer, skin temperature, readings from the longitudinal and transverse accelerometer and time for recording an activity called session time

[9] http://www.cs.waikato.ac.nz/ml/weka/

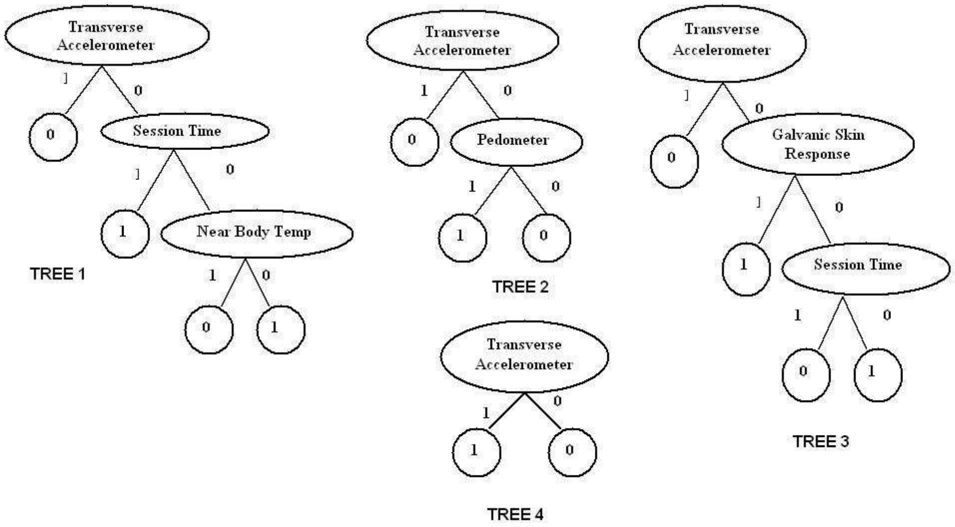

Figure 2. Decision Trees built from four different samples of the physiological data set

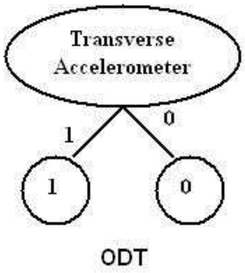

Figure 3. An Orthogonal Decision Tree

to principle component analysis. Orthogonal trees were built, corresponding to the significant components and they were combined using an uniform aggregation scheme. The accuracy and size of the orthogonal trees are noted and compared with the corresponding results generated by Bagging using the same number of decision trees in the ensemble.

The Figure 2 illustrates four decision trees built on the uniformly sampled training data set(each of size 150). The first decision tree, has a complexity 7 and considers attribute transverse acceleromenter reading, session time and near body temperature as ideal for splits. Before pruning, only two instances are mis-classified giving an error of 1.3(%). After pruning, there is no change in structure of the tree. The estimated error percentage is 4.9(%). The second, third and fourth decision trees have complexities 5, 7, and 3 respectively. An illustration of an orthogonal decision tree obtained from the first principle component, is shown in Figure 3.

Figure 4 illustrates the distribution of tree complexity and error in classification for the original C4.5 trees used to construct an ODT ensemble. The total number of nodes in the original C4.5 trees varied between three and thirteen. The trees had an

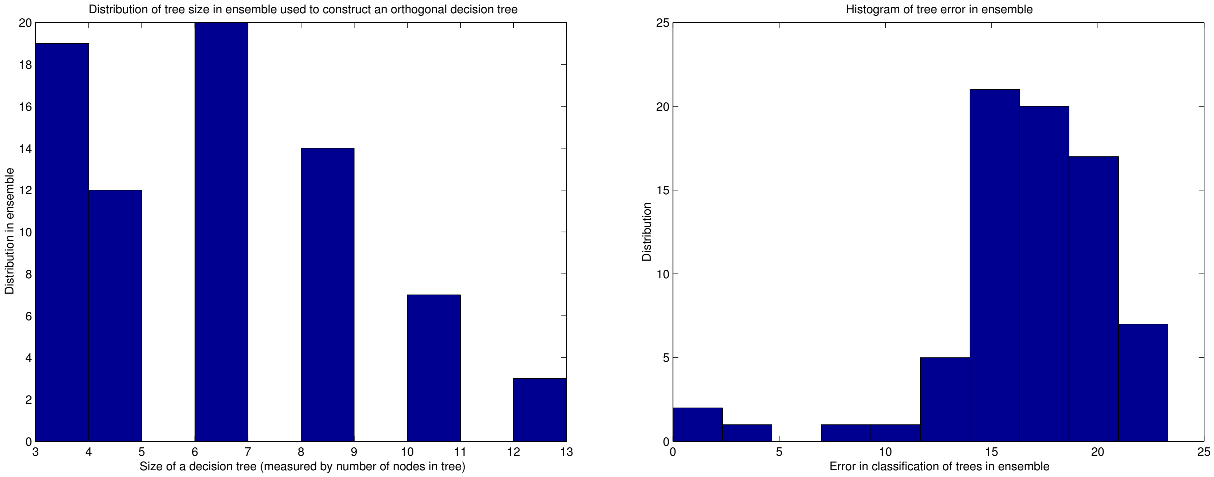

Figure 4. Histogram of tree complexity (left) and error (right) in classification for the original C4.5 trees.

error of less than 25(%). In comparison, the average complexity of the orthogonal decision trees was found to be 3 for all the different ensemble sizes. In fact, for this particular dataset, the sensor reading corresponding to transverse accelerometer attribute was found to be the most interesting. All the orthogonal decision trees used this attribute as the root node for building the trees. The Figure 5 illustrates the distribution of error in classification for an ODT ensemble of 75 trees.

We compared the accuracy obtained from an aggregated orthogonal decision tree to that obtained from a bagging ensemble(using the same number of trees in each case). Figure 6 plots the error in classification of the aggregated ODT and bagging versus the number of decision trees in the ensemble. We found that the classification from an aggregated orthogonal decision tree was better than bagging when the number of trees in the ensemble was smaller. With increase in number of trees in the ensemble bagging provided a slightly better accuracy. It must be noted however, that in constrained environments such as in pocket pcs, personal assistants and sensor network setting, increasing the number of trees in the ensemble arbitarily may not be feasible due to memory constraints.

In resource constrained environments it is often necessary to keep track of the amount of memory used to store the ensemble. In the current implementation storing a node data structure in a tree requires approximately 1 KB of memory. Consider an ensemble of 20 trees. If the average number of nodes in the trees in the ensemble is 7, then we are required to store 140 KB of data. Orthogonal decision trees on the other hand are smaller in size, with less redundancy. In the experiments we performed they typically have a complexity of 3 nodes. This means that we need to store only 3 KB of data.

We define Tree Complexity Ratio (TCR) as the total number of nodes in the ODT versus the total number of nodes in the bagging ensemble. Figure 6 plots the variation of the TCR as the number of trees in the ensemble increases. It may be noted that in resource constrained environments one can opt for meaningful trees of smaller size and comparable accuracy as opposed to larger ensembles with a slightly better accuracy.

An orthogonal decision tree also helps in the feature selection process and indicates which attributes are more important

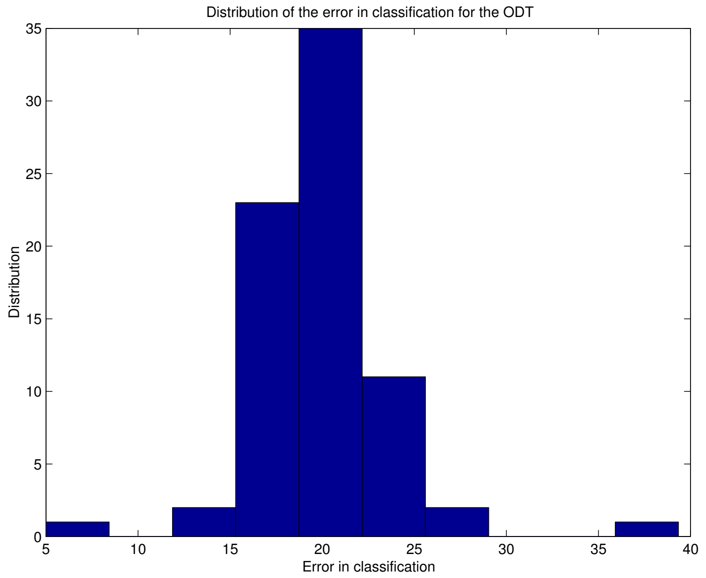

Figure 5. Histogram of error in classification in the ODT ensemble.

than others in the data set. The Figure 8 indicates the variance captured by the first principle component as the number of trees in the ensemble was varied from 5 to 75 trees. As is expected, as the number of trees in the ensemble increases, the first principle component captures most of the variance and those occupied by the second and third components gradually decreases.

The following section illustrates the response time for classification on a pocket pc using a bagging ensemble and an equivalent orthogonal decision tree ensemble.

7 Monitoring in Resource Constrained Environments

Resource Constrained environments such as personal digital assistants, pocket pcs, cell phones are often used to monitor the physiological conditions of subjects. These devices present additional challenges in monitoring owing to the limited battery power, memory restrictions and small displays that they have.

The previous section indicated that an aggregated orthogonal decision tree was small in size, and captured an accuracy better or comparable to that of bagging when the ensemble size was small. Although bagging was found to perform better in larger ensembles, the number of trees that needed to be stored was considerably larger and clearly not an option in the resource constrained environments. Therefore a tradeoff exists between the memory usage and accuracy.

In order to test the response time for monitoring, we performed classification experiments on an HP iPAQ Pocket PC. We assumed that physiological data blocks of size 40 instances were sent to the handheld device. Using training data obtained previously, we precomputed C4.5 decision trees. The Fourier spectra of the trees were evaluated(preserving approximately 99(%) of the total energy) and the coefficient matrix was projected onto the most significant principal components.

Since the time required for computation is of considerable importance in resource constrained environments, we estimated

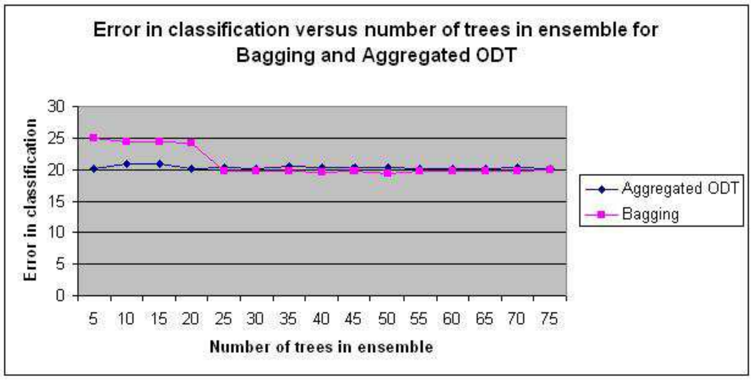

Figure 6. Comparison of error in classification for trees in the ensemble for aggregated ODT versus Bagging.

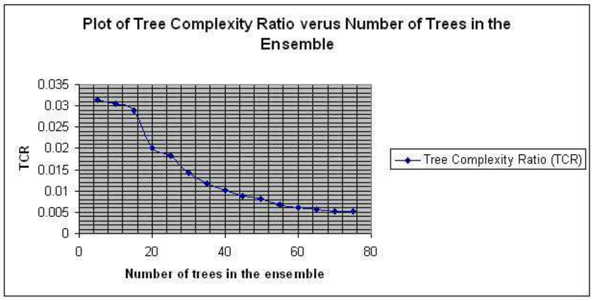

Figure 7. Plot of Tree Complexity Ratio versus number of trees in the ensemble.

the response time for Bagging ensemble versus the equivalent ODT ensemble. We define response time as the time required to produce an accuracy estimate from all the instances available by the specified classification scheme. The Figure 9 illustrates the response time for a bagging ensemble and an equivalent ODT ensemble. Clearly the equivalent orthogonal decision tree produces classification results faster than a bagging ensemble and this may be attributed to the fact that much of the redundancy in bagging ensemble has been removed in the ODT ensemble. Our method thus offers a computationally efficient method for classification on resource constrained devices.

8 Conclusions

Orthogonal decision trees offer an effective way to construct redundancy-free ensembles that are easier to understand and apply. They are particularly useful in monitoring data streams using resource constrained platforms where storage and CPU computing power are limited but fast response is important. ODTs are constructed from the Fourier spectra of the decision trees in the ensemble. Redundancy is removed from the ensemble by performing a PCA of these Fourier

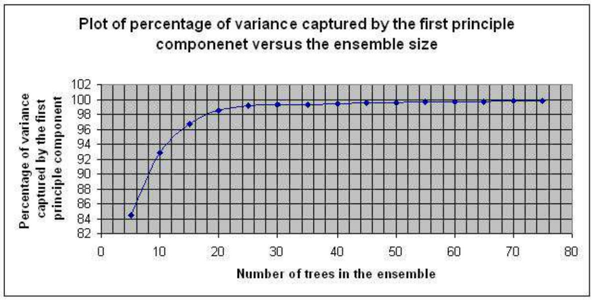

Figure 8. Variance captured by the first principle component versus number of trees in ensemble.

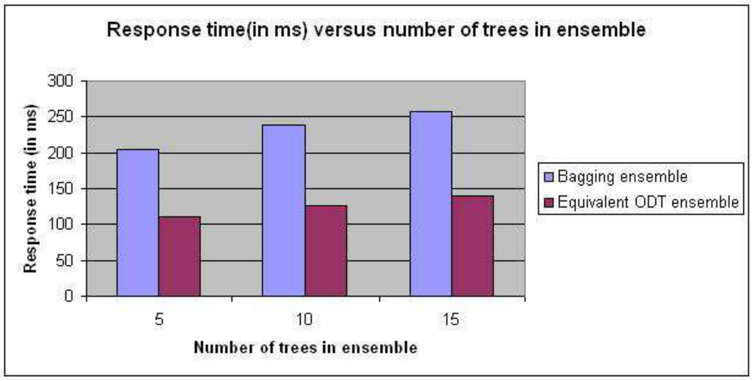

Figure 9. Plot of Response time for Bagging and equivalent ODT ensemble versus the number of trees in the ensemble.

spectra. This offers an efficient representation of the ensemble, often needed for fast response in many real-time data mining applications. This also allows a meaningful way to visualize the trees in a low dimensional space. This paper described an application of orthogonal decision tree ensembles for monitoring physiological data streams in time-critical resource-constrained environments. The current work is an extension of our earlier work [8], [6] in this area. We plan to explore further applications of ODTs in other domains. We are also working on developing techniques that makes use of the spectral representation of an ensembles for identifying its various functional and structural properties (e.g. stability).

Acknowledgments

The authors acknowledge supports from NSF CAREER award IIS-0093353 and NSF grant IIS-0203958.

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.

[4] H. Drucker and C. Cortes. Boosting decision trees. *Advances in Neural Information Processing Systems*, 8:479–485, 1996.

[5] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[6] H. Kargupta and H. Dutta. Orthogonal Decision Trees. In *Fourth IEEE International Conference on Data Mining (ICDM)*, pages 427–430, 2004.

[7] H. Kargupta and B. Park. A Fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):216–229, 2002.

[8] H. Kargupta, B. Park, and H. Dutta. Orthogonal Decision Trees. In *In Communication*, 2004.

[9] Y. Kostov and G. Rao. Low-cost optical instrumentation for biomedical measurements. *Review of Scientific Instruments*, 71(12):4361–4373, December 2000.

[10] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM*, 40:607–620, 1993.

[11] C. J. Merz and M. J. Pazzani. A principal components approach to combining regression estimates. *Machine Learning*, 36(1–2):9–32, 1999.

[12] B. H. Park and H. Kargupta. Constructing simpler decision trees from ensemble models using Fourier analysis. In *Proceedings of the 7th Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM SIGMOD*, pages 18–23, 2002.

[13] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kauffman, 1993.

[15] W. N. Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classificaiton. In *Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2001.

[16] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.