SVM and Dimensionality Reduction in Cognitive Radio with Experimental Validation

Shujie Hou, Robert C. Qiu*, Senior Member, IEEE, Zhe Chen, Zhen Hu, Student Member, IEEE

Abstract—There is a trend of applying machine learning algorithms to cognitive radio. One fundamental open problem is to determine how and where these algorithms are useful in a cognitive radio network. In radar and sensing signal processing, the control of degrees of freedom (DOF)—or dimensionality—is the first step, called pre-processing. In this paper, the combination of dimensionality reduction with SVM is proposed apart from only applying SVM for classification in cognitive radio. Measured Wi-Fi signals with high signal to noise ratio (SNR) are employed to the experiments. The DOF of Wi-Fi signals is extracted by dimensionality reduction techniques. Experimental results show that with dimensionality reduction, the performance of classification is much better with fewer features than that of without dimensionality reduction. The error rates of classification with only one feature of the proposed algorithm can match the error rates of 13 features of the original data. The proposed method will be further tested in our cognitive radio network testbed.

Index Terms—Degrees of freedom (DOF), cognitive radio, support vector machine (SVM), dimensionality reduction.

I. INTRODUCTION

Intelligence and learning are key factors to cognitive radio. Recently, there is a trend of applying machine learning algorithms to cognitive radio [1]. Machine learning [2] is a discipline to design algorithms for computers to imitate human being’s behaviors, which includes learning to recognize the complex patterns and making decisions based on experience automatically and intelligently. The topic of machine learning is precisely that needs to be introduced to cognitive radio. One fundamental open problem is to determine how and where these algorithms are useful in a cognitive radio network. Such a systematical investigation is missing in the literature. It is the motivation of this paper to fill this gap.

In radar and sensing signal processing, the control of degrees of freedom (DOF)—or intrinsic dimensionality—is the first step, called pre-processing. The network dimensionality, on the other hand, has received attention in information theory literature. One naturally wonders how network (signal) dimensionality affects the performance of system operation. Here we study, as an illustrative example, state classification of measured Wi-Fi signal in cognitive radio under this context. Both linear such as principal component analysis (PCA) and nonlinear methods such as kernel principal component analysis (KPCA) and maximum variance unfolding (MVU) are studied, by combining them with support vector machine (SVM) [3]–[8]—the latest breakthrough in machine learning. Dimensionality reduction methods of PCA, KPCA and MVU are systematically studied in this paper which can meet the needs of all kinds of data owning different structures. The reduced dimension data can retain most of the useful information of the original data but have much fewer dimensions.

SVM is both a linear and a nonlinear classifier which has been successfully applied in many areas such as handwritten digit recognition [9]–[11] and object recognition [12]. In cognitive radio, SVM has been exploited to do channel and modulation selection [13], signal classification [14]–[16] and spectrum estimation [17]. In [14], a method for combining feature extraction based on spectral correlation analysis with SVM to classify signals has been proposed. In this paper, SVM method will be explored as a classifier for measured Wi-Fi signal data.

Dimensionality reduction methods are innovative and important tools in machine learning [18]. The original dimensionality data collected in cognitive radio may contain a lot of features, however, usually these features are highly correlated and redundant with noise. Thus the intrinsic dimensionality of the collected data is much fewer than the original features. Dimensionality reduction attempts to select or extract a lower dimensionality expression but retain most of the useful information.

PCA [19] is the best-known linear dimensionality reduction method. PCA takes the variance among data as the useful information—it aims to find a subspace $\Omega$ which can maximally retain the variance of the original dataset. On the other hand, although linear PCA can work well (when such a subspace $\Omega$ exists), it always fails to detect the nonlinear structure of data. A nonlinear dimensionality reduction method called KPCA [20] can be used for this purpose. KPCA uses the kernel tricks [21] to map the original data into a feature space $F$, and then does PCA in $F$ without knowing the mapping explicitly. The leading eigenvector of the sample covariance matrix in the feature space has been explored to spectrum sensing in cognitive radio [22].

Manifold learning has become a very active topic in nonlinear dimensionality reduction. The basic assumption for these manifold learning algorithms is that the input data lie on or close to a smooth manifold [23], [24]. The cognitive radio network happens to lie in this category. A lot of promising methods have been proposed in this context, including isometric mapping (Isomap) [25], local linear embedding (LLE) [26], Laplacian eigenmaps [27], local tangent space alignment (LTSA) [28], MVU [29], Hessian eigenmaps [30],
manifold charting [31], diffusion maps [32] and Riemannian manifold learning (RML) [33]. The MVU approach will be applied to our problem. MVU exploits semidefinite programming method—the latest breakthrough in convex optimization—to solve a convex optimization model that maximizes the output variance, subject to the constrains of zero mean and local isometry.

As aforementioned, the collected data often contains too much redundant information. The redundant information not only complicates the algorithms but also conceals the underlying reason of the performance of the algorithms. Therefore, the intrinsic dimensionality of the collected data should be extracted. In this paper, the combination of dimensionality reduction with SVM is also proposed apart from only applying SVM for classification. The intrinsic structure of the collected data in cognitive radio determines the uses of the corresponding linear or nonlinear dimensionality reduction methods.

The contributions of this paper are as follows. First, SVM is explored to classify the states of Wi-Fi signal successfully. Second, the combination of SVM and dimensionality reduction is the first time proposed to cognitive radio, with the motivation stated above. Third, the measured Wi-Fi signal is explored to classify the states of Wi-Fi signal successfully. The contributions of this paper are as follows. First, SVM is applied to our problem. MVU exploits semidefinite programming to implicitly map the original data into a higher dimensional space in which the separating hyperplane with the largest margin [7] satisfying the following constraints:

\[ w \cdot x + b \geq 1 \quad \text{for} \quad l_{i_t} = 1 \]
\[ w \cdot x + b \leq -1 \quad \text{for} \quad l_{i_t} = -1 \]

for linear separable case, in which \( w \) is the normal vector of the hyperplane and \( b \) for stands for inner product. The constraints (4) can be combined into:

\[ l_{i_t}(w \cdot x_{i_t} + b) \geq 1. \]

The requirements for separating hyperplane with the largest margin can formulate the problem into the following optimization model:

\[ \text{minimize} \quad ||w||^2 \]
\[ \text{subject to} \quad l_{i_t}(w \cdot x_{i_t} + b) \geq 1 \]

in which \( t = 1, 2, \cdots, T \). The dual form of (6) by introducing Lagrange multipliers

\[ \alpha_{i_t} \geq 0, \quad t = 1, 2, \cdots, T \]

is:

\[ \text{maximize} \quad \sum_{i_t} \alpha_{i_t} - \frac{1}{2} \sum_{i_t, j_t} \alpha_{i_t} \alpha_{j_t} l_{i_t} l_{j_t} x_{i_t} \cdot x_{j_t} \]
\[ \text{subject to} \quad \sum_{i_t} \alpha_{i_t} l_{i_t} = 0 \]
\[ \alpha_{i_t} \geq 0, \]

in which

\[ w = \sum_{i_t} \alpha_{i_t} l_{i_t} x_{i_t}. \]

Those \( x_{i_t} \) with \( \alpha_{i_t} > 0 \) are called support vectors. By substituting (9) into (3), the solution of separating hyperplane is:

\[ f(x) = \sum_{i_t} \alpha_{i_t} l_{i_t} x_{i_t} \cdot x + b. \]

The brilliance of the (10) is that it just relies on the inner product between training points and testing point. It allows SVM to be easily generalized to nonlinear SVM. If \( f(x) \) is not a linear function about the data, the nonlinear SVM can be obtained by introducing a kernel function:

\[ k(x_{i_t}, x) = \varphi(x_{i_t}) \cdot \varphi(x) \]

(11) to implicitly map the original data into a higher dimensional feature space \( F \), where \( \varphi \) is the mapping from original space to feature space. In \( F \), \( \varphi(x_{i_t}) \) are linearly separable. The separating hyperplane in feature space \( F \) is easily generalized into the following form:

\[ f(x) = \sum_{i_t} \alpha_{i_t} l_{i_t} \varphi(x_{i_t}) \cdot \varphi(x) + b = \sum_{i_t} \alpha_{i_t} l_{i_t} k(x_{i_t}, x) + b. \]

(12)

By introducing the kernel function, the mapping \( \varphi \) need not be explicitly known which reduces much of the computational complexity. For much more details, refer to [6] [7].

II. SUPPORT VECTOR MACHINE

Supervised learning is one of the three major learning types in machine learning. The dataset of supervised learning consists of \( M \) pairs of inputs/outputs (labels)

\[ (x_{i_t}, l_{i_t}), \quad i = 1, 2, \cdots, M. \] (1)

Suppose the training samples and testing samples are \( x_{i_t} \) with indexes \( i_t, t = 1, 2, \cdots, T \) and \( i_s, s = 1, 2, \cdots, S \), respectively. A supervised learning algorithm seeks mapping from inputs to outputs by training set and predicts any outputs based on the mapping. If the output is categorical or nominal value, then this will become a classification problem. Classification is a common task in machine learning. SVM [6] [7] methods will be employed in two categories’ classification experiments later.

Consider, for example, a simplest two classes classification task with the data that are linearly separable. Under this case, \( l_{i_t} \in \{-1, 1\} \) (2) indicates which class \( x_{i_t} \) belongs to. SVM attempts to find the separating hyperplane

\[ w \cdot x + b = 0 \]

with the largest margin [7] satisfying the following constraints:

\[ w \cdot x_{i_t} + b \geq 1 \quad \text{for} \quad l_{i_t} = 1 \]
\[ w \cdot x_{i_t} + b \leq -1 \quad \text{for} \quad l_{i_t} = -1 \]

for linear separable case, in which \( w \) is the normal vector of the hyperplane and \( b \) stands for inner product. The constraints (4) can be combined into:

\[ l_{i_t}(w \cdot x_{i_t} + b) \geq 1. \]

The requirements for separating hyperplane with the largest margin can formulate the problem into the following optimization model:

\[ \text{minimize} \quad ||w||^2 \]
\[ \text{subject to} \quad l_{i_t}(w \cdot x_{i_t} + b) \geq 1 \]

in which \( t = 1, 2, \cdots, T \). The dual form of (6) by introducing Lagrange multipliers

\[ \alpha_{i_t} \geq 0, \quad t = 1, 2, \cdots, T \]

is:

\[ \text{maximize} \quad \sum_{i_t} \alpha_{i_t} - \frac{1}{2} \sum_{i_t, j_t} \alpha_{i_t} \alpha_{j_t} l_{i_t} l_{j_t} x_{i_t} \cdot x_{j_t} \]
\[ \text{subject to} \quad \sum_{i_t} \alpha_{i_t} l_{i_t} = 0 \]
\[ \alpha_{i_t} \geq 0, \]

in which

\[ w = \sum_{i_t} \alpha_{i_t} l_{i_t} x_{i_t}. \]

Those \( x_{i_t} \) with \( \alpha_{i_t} > 0 \) are called support vectors. By substituting (9) into (3), the solution of separating hyperplane is:

\[ f(x) = \sum_{i_t} \alpha_{i_t} l_{i_t} x_{i_t} \cdot x + b. \] (10)
A function is a valid kernel if there exists a mapping \( \varphi \) satisfying (11). Mercer’s condition \([8]\) gives us the condition about what kind of functions are valid kernels. Actually, some common used kernels are as follows: polynomial kernels
\[
k(x_i, x_j) = (x_i \cdot x_j + 1)^d,
\]
radial basis kernels (RBF)
\[
k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2),
\]
and neural network type kernels
\[
k(x_i, x_j) = \tan h((x_i \cdot x_j + b)),
\]
in which the heavy-tailed RBF kernel is in the form of
\[
k(x_i, x_j) = \exp(-\gamma \|x_i^2 - x_j^2\|^2),
\]
and Gaussian RBF kernel is
\[
k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).
\]

### III. DIMENSIONALITY REDUCTION

Dimensionality reduction is a very effective tool in machine learning field.

In the rest of this paper, assuming the original dimensionality data are a set of \( M \) samples \( x_i \in \mathbb{R}^N, i = 1, 2, \ldots, M, \) the reduced dimensionality samples of \( x_i \) are \( y_i \in \mathbb{R}^K, i = 1, 2, \ldots, M, \) where \( K \ll N \). \( x_{ij} \) and \( y_{ij} \) are componentwise elements in \( x_i \) and \( y_i \), respectively.

#### A. Principal Component Analysis

PCA \([19]\) is the best-known linear dimensionality reduction method. PCA aims to find a subspace \( \Omega \) which can maximally retain the variance of the original dataset. The basis of \( \Omega \) is obtained by eigen-decomposition of covariance matrix. The procedure can be summarized into the following four steps.

1) Compute the covariance matrix of \( x_i \)
\[
C = \frac{1}{M} \sum_{i=1}^{M} \left(x_i - \bar{u}\right)\left(x_i - \bar{u}\right)^T
\]
where \( \bar{u} = \frac{1}{M} \sum_{i=1}^{M} x_i \) is the mean of the given samples, \( T \) means transpose.
2) Calculate eigenvalues \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \) and the corresponding eigenvectors \( v_1, v_2, \cdots, v_N \) of the covariance matrix \( C \).
3) The basis of \( \Omega \) is \( v_1, v_2, \cdots, v_K \).
4) Dimensionality reduction by
\[
y_{ij} = (x_i - \bar{u}) \cdot v_j.
\]
The value of \( K \) is determined by the criteria
\[
\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > \text{threshold}.
\]
Usually threshold = 0.95 or 0.90.

#### B. Kernel Principal Component Analysis

PCA works well for the high dimensionality data with linear variability, but always fails when nonlinear nature exists. KPCA \([20]\) is, on the other hand, designed to extract the nonlinear structure of the original data. It uses the kernel function \( k \) (same as SVM) to implicitly map the original data into a feature space \( F \), where \( \varphi \) is the mapping from original space to feature space. In \( F \), PCA algorithm can work well.

If \( k \) is valid kernel function, the matrix
\[
K = (k(x_i, x_j))_{i,j=1}^{M}
\]
must be positive semi-definite. The matrix \( K \) is the so-called kernel matrix.

Assuming the mean of feature space data \( \varphi(x_i), i = 1, 2, \cdots, M \) is zero, i.e.,
\[
\frac{1}{M} \sum_{i=1}^{M} \varphi(x_i) = 0.
\]
The covariance matrix in \( F \) is
\[
C_F = \frac{1}{M} \sum_{i=1}^{M} \varphi(x_i)\varphi(x_i)^T.
\]
In order to apply PCA in \( F \), the eigenvectors \( v_i^F \) of \( C_F \) are needed. As we know that the mapping \( \varphi \) is not explicitly known, thus the eigenvectors of \( C_F \) cannot be as easily derived as PCA. However, the eigenvectors \( v_i^F \) of \( C_F \) must lie in the span \([20]\) of \( \varphi(x_i), i = 1, 2, \cdots, M \), i.e.,
\[
v_i^F = \sum_{j=1}^{M} \alpha_{ij} \varphi(x_j).
\]
It has been proved that \( \alpha_{ij}, i = 1, 2, \cdots, M \) are eigenvectors of kernel matrix \( K \) \([20]\). In which \( \alpha_{ij} \) are componentwise elements of \( \alpha_i \).

Then the procedure of KPCA can be summarized into the following six steps:

1) Choose a kernel function \( k \).
2) Compute kernel matrix
\[
K_{ij} = k(x_i, x_j).
\]
3) The eigenvalues \( \lambda_1^K \geq \lambda_2^K \geq \cdots \geq \lambda_M^K \) and the corresponding eigenvectors \( \alpha_1, \alpha_2, \cdots, \alpha_M \) are obtained by diagonalizing \( K \).
4) Normalizing \( v_j^F \) by \([20]\)
\[
1 = \lambda_j^K(\alpha_j \cdot \alpha_j).
\]
5) The normalized eigenvectors \( v_j^F, j = 1, 2, \cdots, K \) constitute the basis of a subspace in \( F \).
6) The projection of a training point \( x_i \) on \( v_j^F, j = 1, 2, \cdots, K \) is computed by
\[
y_{ij} = (v_j^F, x_i) = \sum_{n=1}^{M} \alpha_{jn} k(x_n, x_i).
\]
The idea of kernel in KPCA is exactly the same with kernels in SVM. All of kernel functions in SVM can be employed in KPCA, too.
So far the mean of $\varphi(x_i), i = 1, 2, \cdots, M$ has been assumed to be zero. In fact, the zero mean data in the feature space are

$$\varphi(x_i) - \frac{1}{M} \sum_{i=1}^{M} \varphi(x_i).$$

The kernel matrix for this centering or zero mean data can be derived by \[20\]

$$\tilde{K} = K - 1_MK - K1_M + 1_MK1_M$$

in which $(1_M)_{ij} := 1/M$.

C. Maximum Variance Unfolding

MVU \[29\] approach will be applied in our experiments among all the manifold learning methods. Resorting to the help of optimization toolbox, MVU can learn the inner product matrix of $y_i$ automatically by maximizing their variance subject to the constraints that $y_i$ are centered and local distances of $y_i$ are equal to the local distances of $x_i$. Here the local distances represent the distances between $y_i$ ($x_i$) and its $k$ nearest neighbors, in which $k$ is a parameter.

The intuitive explanation of this approach is that when an object such as string is unfolded optimally, the Euclidean distances between its two ends must be maximized. Thus the optimization objective function can be written as

$$\text{maximize } \sum_{ij} \|y_i - y_j\|^2,$$

subject to the constraints,

$$\sum_{j} y_j = 0 \quad \sum_{j} y_j^2 = \|x_i - x_j\|^2, \text{ when } \eta_{ij} = 1$$

in which $\eta_{ij} = 1$ means $x_i$ and $x_j$ are $k$ nearest neighbors otherwise $\eta_{ij} = 0$.

Apply inner product matrix

$$I = (y_i \cdot y_j)_{i,j=1}^{M}$$

of $y_i$ to the above optimization can make the model simpler. The procedure of MVU can be summarized as follows:

1) Optimization step: because $I$ is an inner product matrix, it must be positive semi-definite. Thus the above optimization can be reformulated into the following form \[29\]

$$\text{maximize } \text{trace}(I)$$

subject to

$$I > 0$$

$$\sum_{ij} I_{ij} = 0$$

$$I_{ii} - 2I_{ij} + I_{jj} = D_{ij}, \text{ when } \eta_{ij} = 1$$

where $D_{ij} = \|x_i - x_j\|^2$, and $I > 0$ represents $I$ is positive semi-definite.

2) The eigenvalues $\lambda_1^y \geq \lambda_2^y \geq \cdots \geq \lambda_M^y$ and the corresponding eigenvectors $v_1^y, v_2^y, \cdots, v_M^y$ are obtained by diagonalizing $I$.

3) Dimensionality reduction by

$$y_i = \sqrt{\lambda_j^y} v_j^y$$

in which $v_j^y$ are componentwise elements of $v_j^y$.

Landmark-MVU (LMVU) \[39\] is a modified version of MVU which aims at solving larger scale problems than MVU. It works by using the inner product matrix $A$ of randomly chosen landmarks from $x_i$ to approximate the full matrix $I$, in which the size of $A$ is much smaller than $I$.

Assuming the number of landmarks is $m$ which are $a_1, a_2, \cdots, a_m$, respectively. Let $Q \ [39]$ denote a linear transformation between landmarks and original dimensional data $x_i \in \mathbb{R}^N, i = 1, 2, \cdots, M$, accordingly,

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \approx Q \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$$

in which

$$x_i = \sum_{j} Q_{ij} a_j, \quad i = 1, 2, \cdots, M.$$  

Assuming the reduced dimensionality landmarks of $a_1, a_2, \cdots, a_m$ are $\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_m$, and the reduced dimensionality samples of $x_1, x_2, \cdots, x_M$ are $y_1, y_2, \cdots, y_M$, then the linear transformation between $y_1, y_2, \cdots, y_M$ and $\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_m$ is $Q$ as well \[39\], consequently,

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} \approx Q \cdot \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{pmatrix}.$$  

Matrix $A$ is the inner-product matrix of $a_1, a_2, \cdots, a_m$,

$$A = (\tilde{y}_i \cdot \tilde{y}_j)_{i,j=1}^{m}, \text{ hence the relationship between } I \text{ and } A \text{ is }$$

$$I \approx QAQ^T.$$  

The optimization of \[33\] can be formulated into the following form:

$$\text{maximize } \text{trace } QAQ^T$$

subject to $A > 0$ \[40\],

$$\sum_{ij} (QAQ^T)_{ij} = 0$$

$$D_{ij} \leq D_{ij}, \text{ when } \eta_{ij} = 1$$

in which

$$D_{ij} = \|x_i - x_j\|^2,$$

$$D_{ij}^{y} = (QAQ^T)_{ij} - 2(QAQ^T)_{ij} + (QAQ^T)_{jj}, \text{ and } A > 0 \text{ represents } A \text{ is positive semi-definite. This optimization model differs from }$$

$$\text{in that equality constraints for nearby distances are relaxed to inequality constraints in order to guarantee the feasibility of the simplified optimization model.}$$

LMVU can increase the speed of programming but with the cost of decreasing accuracy. In this paper’s simulation, the LMVU will be applied.
IV. THE PROCEDURE OF USING SVM AND DIMENSIONALITY REDUCTION

As aforementioned, in radar and sensing signal processing, the control of DOF—or dimensionality—is the first step, called pre-processing. Both the linear and nonlinear methods will be investigated in this paper as pre-processing tools to extract the intrinsic dimensionality of the collected data.

First, SVM will be employed for classification in cognitive radio. The classification power of SVM is tested by the testing set. The procedure of SVM for classification is summarized as follows:

1) The collected dataset $\mathbf{x}_i, i = 1, 2, \cdots, M$ will be divided into training sets and testing sets. The training samples and testing samples are $\mathbf{x}_i$ with indexes $i_t, t = 1, 2, \cdots, T$ and $i_s, s = 1, 2, \cdots, S$, respectively.

2) The labels $l_{i_t}$ and $l_{i_s}$ for $\mathbf{x}_{i_t}$ and $\mathbf{x}_{i_s}$ are extracted.

3) Choose a kernel function from (13) to (17) and the corresponding parameter’s values for the chosen kernel are designated.

4) The separating hyperplane in higher dimensional feature space $\mathbf{F}$ which is in the form of (12) is trained by the training set $\mathbf{x}_{i_t}$.

5) The classification performance of the trained hyperplane will be tested by the testing set $\mathbf{x}_{i_s}$.

The above process will be repeated to gain averaged test errors.

Apart from applying SVM to classification, dimensionality reduction will be implemented before SVM to get rid of the redundant information of the collected data. The procedure of the proposed algorithm that is SVM combined with dimensionality reduction can be summarized as follows:

1) The collected dataset $\mathbf{x}_i, i = 1, 2, \cdots, M$ will be divided into training sets and testing sets. The training samples and testing samples are $\mathbf{x}_i$ with indexes $i_t, t = 1, 2, \cdots, T$ and $i_s, s = 1, 2, \cdots, S$, respectively.

2) The labels $l_{i_t}$ and $l_{i_s}$ for $\mathbf{x}_{i_t}$ and $\mathbf{x}_{i_s}$ are extracted.

3) Obtain reduced dimension data $\mathbf{y}_{i_t}$ and $\mathbf{y}_{i_s}$ by using dimensionality reduction methods.

4) $\mathbf{y}_{i_t}$ and $\mathbf{y}_{i_s}$ are taken as the new training set and testing set.

5) The labels $l_{i_t}$ and $l_{i_s}$ for $\mathbf{y}_{i_t}$ and $\mathbf{y}_{i_s}$ are kept unchanged with $\mathbf{x}_{i_t}$ and $\mathbf{x}_{i_s}$.

6) Choose a kernel function from (13) to (17), and the corresponding parameter’s values for the chosen kernel are designated.

7) The separating hyperplane in higher dimensional feature space $\mathbf{F}$ which is in the form of (12) is trained by the new training set $\mathbf{y}_{i_t}$.

8) The classification performance of the trained hyperplane will be tested by the new testing set $\mathbf{y}_{i_s}$.

The above process will be repeated to gain averaged test errors. The flow chart of SVM combined with dimensionality reduction for classification is shown in Fig. 1.

Dimensionality reduction methods of PCA, KPCA and MVU are systematically studied in this paper which can meet the needs of all kinds of data owning different structures. Dimensionality reduction with PCA to derive $\mathbf{y}_{i_t}$ and $\mathbf{y}_{i_s}$ is implemented as follows.

1) The training set $\mathbf{x}_{i_t}$ is input to PCA procedure from step 1) to step 3) in section III-A to obtain the eigenvectors of $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K$.

2) Dimensionality reduction by

$$
\mathbf{y}_{i_t}^T = (\mathbf{x}_{i_t} - \mathbf{u}) \cdot (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K),
$$

(43)

$$
\mathbf{y}_{i_s}^T = (\mathbf{x}_{i_s} - \mathbf{u}) \cdot (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K).
$$

(44)

Dimensionality reduction with KPCA to derive $\mathbf{y}_{i_t}$ and $\mathbf{y}_{i_s}$ is implemented as follows.

1) The training set $\mathbf{x}_{i_t}$ is input to KPCA procedure from step 1) to step 5) in section III-B to obtain eigenvectors $\mathbf{v}_j^F, j = 1, 2, \cdots, K$, implicitly.

2) Dimensionality reduction by

$$
\mathbf{y}_{i_t}^T = \left( (\mathbf{v}_1^F, \cdots, \mathbf{v}_M^F) \cdot \mathbf{x}_{i_t} \right)
= \sum_{n=1}^{T} \alpha_{nk}^F(\mathbf{x}_n, \mathbf{x}_{i_t}),
$$

(45)

$$
\mathbf{y}_{i_s}^T = \left( (\mathbf{v}_1^F, \cdots, \mathbf{v}_M^F) \cdot \mathbf{x}_{i_s} \right)
= \sum_{n=1}^{T} \alpha_{nk}^F(\mathbf{x}_n, \mathbf{x}_{i_s}).
$$

(46)

Dimensionality reduction with LMVU to derive $\mathbf{y}_{i_t}$ and $\mathbf{y}_{i_s}$ is implemented as follows.

1) Both the training set and testing set $\mathbf{x}_{i_t}$ and $\mathbf{x}_{i_s}$ are input to MVU procedure from step 1) to step 2) in section III-C to obtain eigenvalues $\lambda_1^M \geq \lambda_2^M \geq \cdots \geq \lambda_M^M$ and the corresponding eigenvectors $\mathbf{v}_1^M, \mathbf{v}_2^M, \cdots, \mathbf{v}_M^M$.

2) Dimensionality reduction by

$$
\mathbf{y}_{i_t} = \left( \sqrt{\lambda_1^M} \mathbf{v}_{i_t,1}^M, \cdots, \sqrt{\lambda_M^M} \mathbf{v}_{i_t,K}^M \right),
$$

(47)

$$
\mathbf{y}_{i_s} = \left( \sqrt{\lambda_1^M} \mathbf{v}_{i_s,1}^M, \cdots, \sqrt{\lambda_M^M} \mathbf{v}_{i_s,K}^M \right).
$$

(48)

To use LMVU, it is necessary to substitute (33) by (40) in the MVU procedure.
V. WI-FI SIGNAL MEASUREMENT

Wi-Fi time-domain signals have been measured and recorded using an advanced DPO whose model is Tektronix DPO72004 [40]. The DPO supports a maximum bandwidth of 20 GHz and a maximum sampling rate of 50 GS/s. It is capable to record up to 250 M samples per channel. In the measurement, a laptop accesses the Internet through a wireless Wi-Fi router, as shown in Fig. 2. An antenna with a frequency range of 800 MHz to 2500 MHz is placed near the laptop and connected to the DPO. The sampling rate of the DPO is set to 6.25 GS/s. Recorded time-domain Wi-Fi signals are shown in Fig. 3. The duration of the recorded Wi-Fi signals is 40 ms.

The recorded 40-ms Wi-Fi signals are divided into 8000 slots, with each slot lasting 5 µs. The time-domain Wi-Fi signals within the first 1 µs of every slot are then transformed into frequency domain using fast Fourier transform (FFT). The spectral states of the measured Wi-Fi signal at each time slot present two possibilities. One is that current spectrum is occupied (state is busy $l_i = 1$) at this time slot or current spectrum is unoccupied (state is idle $l_i = 0$). In this paper, the frequency band of 2.411 - 2.433 GHz is considered. The resolution in frequency domain is 1 MHz. Thus, for each slot, 23 points in frequency domain can be obtained. The total obtained data are shown in Fig. 4.

VI. EXPERIMENTAL VALIDATION

The spectral states of all time slots for measured Wi-Fi signal can be divided into two classes (busy $l_i = 1$ or idle $l_i = 0$). The powerful classification technique in machine learning, SVM, can be employed to classify the data at each time slot. The processed Wi-Fi data is shown in Fig. 4. The data used below are from the 1101th time slot to the 8000th time slot.

In the next experiment, amplitude values in frequency domain of $i^{th}$ usable time plot are taken as $x_i$. The experiment data is taken by

$$\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_{tot}
\end{bmatrix} = \begin{bmatrix}
    x_{1,12-m}, \ldots, x_{1,12}, \ldots, x_{1,12+n} \\
    x_{2,12-m}, \ldots, x_{2,12}, \ldots, x_{2,12+n} \\
    \vdots \\
    x_{tot,12-m}, \ldots, x_{tot,12}, \ldots, x_{tot,12+n}
\end{bmatrix}$$ (49)

where $x_{ij}$ represents amplitude value on the $j^{th}$ frequency point of the $i^{th}$ time slot. The dimension of $x_i$ is $N = n + m + 1$, in which $0 \leq n - m \leq 1$.

First, the true state $l_i$ of $x_i$ is obtained. The false alarm rate, miss detection rate and total error rate of experiments are shown in Fig. 5, Fig. 6 and Fig. 7 respectively, in which total error rates are the total number of miss detection and false alarm samples divided by the total number of testing set. The results shown are the corresponding averaged values of 50 experiments. In each experiment, the number of training set is 200 and the number of testing set is 1800.

In these results, the steps of the “SVM” method for the $j^{th}$ experiment are:

1) Training set and testing set are chosen denoted by $x_{i,t}$ and $x_{i,t}$.
2) $x_{i,t}$ and $x_{i,t}$ are taken by (49) with dimensionality $N = 1, 2, 3, \ldots, 13$, in which $0 \leq n - m \leq 1$. 
3) For each dimension $N$, $x_{i_1}$ and $x_{i_2}$ are fed to SVM algorithm.

The above process repeats with $j = 1, 2, \ldots, 50$, then the corresponding averaged values of these 50 experiments are derived for each dimension.

However, for the methods of “PCA with SVM”, “KPCA with SVM” and “LMVU with SVM”, their steps for the $j$th experiment are:

1) Training set and testing set are chosen denoted by $x_{i_1}$ and $x_{i_2}$.
2) $x_{i_1}$ and $x_{i_2}$ are taken by (49) with dimensionality $N = 13$, in which $n = m = 6$.
3) $y_{i_1}$ and $y_{i_2}$ are reduced dimensional samples of $x_{i_1}$ and $x_{i_2}$ with PCA, KPCA and LMVU methods, respectively.
4) The dimensions of $y_{i_1}$ and $y_{i_2}$ vary by $K = 1, 2, 3, \ldots, 13$ manually.
5) For each dimension $K$, $y_{i_1}$ and $y_{i_2}$ are fed to SVM algorithm.

The above process repeats with $j = 1, 2, \ldots, 50$, then the corresponding averaged values of these 50 experiments are derived for each dimension.

In fact, for LMVU approach, both the placements and the number of landmarks can influence its performance. The choice of landmarks for each experiment is as follows. For every experiment, the number of landmarks $m$ is equal to 20. At the beginning of the LMVU process, ten groups of randomly chosen positions in $x_i$ (including both training and testing sets) are obtained with 20 positions in each group, and then these positions are fixed. For each experiment, results with landmark positions assigned by each group are obtained. Given ten groups of landmarks, the group of which that can get minimal total error rate is taken as the landmarks for this experiment.

In this whole experiment, Gaussian RBF kernel with $2\sigma^2 = 5.5^2$ is used for KPCA. The parameter $k = 3$, in which $k$ is the number of nearest neighbors of $y_i$ ($x_{i_1}$) (including both training and testing sets) for LMVU. The optimization toolbox SeDuMi 1.1R3 [41] is applied to solve the optimization step in LMVU. The SVM toolbox SVM-KM [42] is used to train and test SVM processes. The kernels selected for SVM are heavy-tailed RBF kernels with parameters $\gamma = 1, a = 1, b = 1$. These parameters keep unchanged for the whole experiment.

Take data set of the first experiment as an example. The distributions of corresponding normalized (the summation of total eigenvalues equals one) eigenvalues of PCA, KPCA and LMVU methods are shown in Fig. 8. It can be seen that the largest eigenvalues for the three methods are all dominant ($84\%, 73\%, 97\%$ for PCA, KPCA and LMVU, respectively). Consequently, reduced one-dimensional data can even extract most of the useful information from the original data.

As can be seen from Fig. 5, Fig. 6 and Fig. 7, with only use of SVM, the classification errors are very low. It means SVM successfully classifies the states of the Wi-Fi signal. The results also verify the fact that more and more dimensions of the data are included (more information are included), the error rates should become smaller and smaller globally. On the other hand, the results with $y_i$ fed to SVM can gain a higher accuracy at lower dimensionality than with $x_i$ directly fed to SVM. By dimensionality reduction, most of the useful information including in $x_i$ can be extracted to the first $K = 1$ dimension of $y_i$. Therefore, even if we increase the dimensions of the reduced data, the error rates do not embody obvious improvement. The error rates with only one feature of the proposed algorithm can match the error rates of 13 features of the original data.

**VII. CONCLUSION**

One fundamental open problem is to determine how and where machine learning algorithms are useful in a cognitive radio network. The network dimensionality has received attention in information theory literature. One naturally wonders
how network (signal) dimensionality affects the performance of system operation. Here we study, as an illustrative example, spectral states classification under this context. Both linear (PCA) and nonlinear methods (KPCA and LMVU) are studied, by combining them with SVM.

Experimental results show that data with only one feature fed to SVM, the false alarm rate of method with dimensionality reduction is at worst 0.03068% comparing with 0.06581% of method without dimensionality reduction, and the miss detection rate is 0.9883% comparing with 4.255%. The error rates with only one feature of the methods with dimensionality reduction can nearly match the error rates of 13 features of the original data.

The results of only applying SVM verify the fact that more and more dimensions of the data are included, the error rates should become smaller and smaller globally since more information of the original data are considered. However, SVM combined with dimensionality reduction does not embody such property. This is because that the reduced dimension data with only one dimension already extracts most of the information of the original Wi-Fi signal. Therefore, even if we increase the dimensions of the reduced data, the error rates do not embody obvious improvement.

In this paper, SNR of the measured Wi-Fi signal is high which makes the error rates of the classification all very low. In fact, dimensionality reduction can not only get rid of the redundant information but also have the effect of denoising. When the collected data contains more noise, SVM combined with dimensionality reduction should embody more advantages. Besides, the original dimension of the spectral domain Wi-Fi signal is not high which also makes the advantage of dimensionality reduction not obvious. In the future, dimensionality reduction used as pre-processing tool will be applied to more scenarios in cognitive radio such as low SNR (spectrum sensing) and very high original dimensions (cognitive radio network) contexts.

In the MVU approach, LMVU method is used in this paper which decreases the accuracy. On the other hand, the optimization toolbox ScDuMi 1.1R3 is used to solve the optimization step which slows down the computation. The dedicated optimization algorithm for MVU should be proposed in the future.

In the future, more machine learning algorithms will be, systematically, explored and applied to the cognitive radio network under different scenarios.

ACKNOWLEDGMENT

This work is funded by National Science Foundation through two grants (ECCS-0901420 and ECCS-0821658), and Office of Naval Research through two grants (N00010-10-1-0810 and N00014-11-1-0006).

REFERENCES

[1] C. Clancy, J. Hecker, E. Stuntebeck, and T. O’Shea, “Applications of machine learning to cognitive radio networks,” IEEE Wireless Communications, vol. 14, no. 4, pp. 47–52, 2007.
[2] C. Bishop et al., Pattern recognition and machine learning. Springer New York, 2006.
[3] V. Vapnik, The nature of statistical learning theory. Springer Verlag, 2000.
[4] V. Vapnik, Statistical learning theory. Wiley New York, 1998.
[5] V. Vapnik, S. Golowich, and A. Smola, “Support vector method for function approximation, regression estimation, and signal processing,” Advances in neural information processing systems, pp. 281–287, 1997.
[6] C. Burges, “A tutorial on support vector machines for pattern recognition,” Data mining and knowledge discovery, vol. 2, no. 2, pp. 121–167, 1998.
[7] A. Smola and B. Schölkopf, “A tutorial on support vector regression,” Statistics and Computing, vol. 14, no. 3, pp. 199–222, 2004.
[8] N. Cristianini and J. Shawe-Taylor, An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge Univ Pr, 2000.
[9] C. Cortes and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.
[10] B. Schölkopf, C. Burges, and V. Vapnik, “Extracting support data for a given task,” in Proceedings, First International Conference on Knowledge Discovery & Data Mining. AAAI Press, Menlo Park, CA, 1995.
[11] B. Schölkopf, C. Burges, and V. Vapnik, “Incorporating invariances in support vector learning machines,” Artificial Neural NetworksICANN 96, pp. 47–52, 1996.
[12] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter, “Comparison of view-based object recognition algorithms using realistic 3D models;” Artificial Neural NetworksICANN 96, pp. 251–256, 1996.
[13] G. Xu and Y. Lu, “Channel and Modulation Selection Based on Support Vector Machines for Cognitive Radio,” in International Conference on Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006, pp. 1–4, 2006.
[14] H. Hu, Y. Wang, and J. Song, “Signal classification based on spectral correlation analysis and SVM in cognitive radio,” in Proceedings of the 22nd International Conference on Advanced Information Networking and Applications, pp. 883–887, IEEE Computer Society, 2008.
[15] M. Ramon, T. Atwood, S. Barbin, and C. Christodoulou, “Signal Classification with an SVM-FFT Approach for Feature Extraction in Cognitive Radio,”
[16] X. He, Z. Zeng, and C. Guo, “Signal Classification Based on Cyclostationary Spectral Analysis and HMM/SVM in Cognitive Radio,” in 2009 International Conference on Measuring Technology and Mechatronics Automation, pp. 309–312, IEEE, 2009.
[17] T. Atwood, M. Martinez-Ramon, and C. Christodoulou, “Robust Support Vector Machine Spectrum Estimation in Cognitive Radio,” in Proceedings of the 2009 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, 2009.
[18] J. Lee and M. Verleysen, Nonlinear dimensionality reduction. Springer Verlag, 2007.
[19] I. Jolliffe, Principal component analysis. Springer verlag, 2002.
[20] B. Schölkopf, A. Smola, and K. Muller, “Nonlinear component analysis as a kernel eigenvalue problem,” Neural computation, vol. 10, no. 5, pp. 1299–1319, 1998.
[21] B. Schölkopf and A. Smola, Learning with kernels. Citeseer, 2002.
[22] S. Hou and R. Qiu, “Spectrum sensing for cognitive radio using kernel-based learning,” Arxiv preprint arXiv:1105.2978, 2011.
[23] H. Seung and D. Lee, “The manifold ways of perception,” Science, vol. 290, no. 5500, pp. 2268–2269, 2000.
[24] H. Murase and S. Nayar, “Visual learning and recognition of 3-D objects from appearance,” International Journal of Computer Vision, vol. 14, no. 1, pp. 5–24, 1995.
[25] J. Tenenbaum, V. Silva, and J. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, no. 5500, p. 2319, 2000.
[26] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, p. 2323, 2000.
[27] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” Neural computation, vol. 15, no. 6, pp. 1373–1396, 2003.
[28] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimension reduction via local tangent space alignment,” SIAM Journal of Scientific Computing, vol. 26, no. 1, pp. 313–338, 2004.
[29] K. Weinberger and L. Saul, “Unsupervised learning of image manifolds by semidefinite programming,” International Journal of Computer Vision, vol. 70, no. 1, pp. 77–99, 2006.
[30] D. Donoho and C. Grimes, “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data;” Proceedings of the National Academy of Sciences, vol. 100, no. 10, p. 5591, 2003.
[31] M. Brand, “Charting a manifold,” Advances in Neural Information Processing Systems., pp. 985–992, 2003.

[32] R. Coifman and S. Lafon, “Diffusion maps,” Applied and Computational Harmonic Analysis, vol. 21, no. 1, pp. 5–30, 2006.

[33] T. Lin, H. Zha, and S. Lee, “Riemannian manifold learning for nonlinear dimensionality reduction,” Lecture Notes in Computer Science, vol. 3951, p. 44, 2006.

[34] R. C. Qiu, Z. Chen, N. Guo, Y. Song, P. Zhang, H. Li, and L. Lai, “Towards a real-time cognitive radio network testbed: architecture, hardware platform, and application to smart grid,” in Proceedings of the fifth IEEE Workshop on Networking Technologies for Software-Defined Radio and White Space; June 2010.

[35] H. Li, L. Lai, and R. Qiu, “Communication Capacity Requirement for Reliable and Secure Meter Reading in Smart Grid,” in First IEEE International Conference on Smart Grid Communications, pp. 191–196, October 2010.

[36] H. Li, R. Mao, L. Lai, and R. Qiu, “Compressed Meter Reading for Delay-sensitive and Secure Load Report in Smart Grid,” in First IEEE International Conference on Smart Grid Communications, pp. 114–119, October 2010.

[37] R. C. Qiu, M. C. Wicks, L. Li, Z. Hu, S. Hou, P. Chen, , and J. P. Browning, “Wireless tomography, part I: a novel approach to remote sensing,” in Proceedings of IEEE 5th International Waveform Diversity & Design Conference, 2010.

[38] R. C. Qiu, Z. Hu, M. C. Wicks, S. Hou, L. Li., and J. L. Garry, “Wireless tomography, part II: a system engineering approach,” in Proceedings of IEEE 5th International Waveform Diversity & Design Conference, 2010.

[39] K. Weinberger, B. Packer, and L. Saul, “Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization,” in Proceedings of the tenth international workshop on artificial intelligence and statistics, pp. 381–388, Citeseer, 2005.

[40] Z. Chen and R. C. Qiu, “Prediction of channel state for cognitive radio using higher-order hidden Markov model,” in Proceedings of the IEEE SoutheastCon, pp. 276 – 282, March 2010.

[41] J. Sturm, “the Advanced Optimization Laboratory at McMaster University, Canada. SeDuMi version 1.1 R3,” 2006.

[42] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, “Svm and kernel methods matlab toolbox.” Perception Systmes et Information, INSA de Rouen, Rouen, France, 2005.