Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 494–503

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

Fast Kernel Sparse Representation Classifier using Improved Smoothed-l_0 Norm

Dzati Athiar Ramli*, Tan Wan Chien

Intelligent Biometric Group (IBG), School of Electrical & Electronic Engineering, USM Engineering Campus, Universiti Sains Malaysia, 14300, Nibong Tebal, Penang, Malaysia

Abstract

The computation time for solving classification problem using sparse representation classifier remains a huge drawback as it is to be implemented in real time applications. The time consuming of sparse representation classifier is mainly due to the sparse signal recovery solver which is based on l_1 minimization or Basis Pursuit. Since then, a fast version of sparse signal recovery solver is introduced and it is based on smoothing the discontinuous properties of l_0 norm. In this work, a smoothed l_0 norm solver is implemented in sparse representation classifier algorithm. This smoothed l_0 norm solver is also modified and improved in such a way to increase its classification accuracy and to further reduce the computation time. The use of kernel version of sparse representation classifier to this modified solver is also implemented and described in this paper. Experiments based on human speech data are carried out in order to compare the improved version of sparse representation classifier with the state of the art classifiers. Experimental results prove that the computation time for classification using proposed algorithm is greatly reduced compared to the baseline performances.

Keywords: Kernel trick; Sparse representation classifier; Smoothed-$\boldsymbol{l_0}$ norm

* Dzati Athiar Ramli. Tel.: +06 04 5996028; fax: +06 04 5996909.
E-mail address: dzati@usm.my

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.148

1. Introduction

The study and development of sparse representation of signals for classification purpose has become high attention among researchers recently. Sparse representation of signal is an expression of the signal as a linear combination of atoms in an over-complete dictionary in which few of the entries are non-zero. Mathematically, a linear combination of sparse representation can be written as the following:

$$y_{m\times 1} = A_{m\times n}x_{n\times 1} \tag{1}$$

where y is the input signal which is in $R^{m\times 1}$ space, A is the dictionary which is in $R^{m\times n}$ $space$, and x is the sparse solution which is in $R^{n\times 1}$ space. In order to ensure that x is sparse, i.e. many of its entries are zero, optimization problem as the following is needed to be solved.

$$\text{minimise } \|x\|_0 \qquad subject\ to \quad y = Ax \tag{2}$$

Here, $\|x\|_0$ denotes the l_0 pseudo norm which is the number of non-zero entries in x. As stated in the literature[1,2], solving l_0 norm minimization is a nondeterministic polynomial (NP) hard problem as the dimension increases and it is too sensitive to noise. Therefore, many substitution approaches are developed by the researchers such as Basis Pursuit (BP) and Matching Pursuit (MP). The idea of BP is to replace the l_0 norm with l_1 norm and solving the minimization using linear programming. Whereas in MP, the algorithm involving searching one atom at a time which best describes the input signal for the first iteration, and the residual error for the following iteration until the threshold drops to a desired value. However, the time consumption of solving BP algorithm and poor estimation quality of MP are the weakness of this algorithm. To overcome this, a fast method called Smoothed-l_0 norm (SL0) is introduce in[3] which approximate the l_0 norm by a smooth function and applying gradient based methods for the minimization problem. This algorithm is very fast while maintaining the same accuracy and phase transition as the BP algorithm. This smoothed l_0 algorithm is a very fast algorithm which proved by the author that it is about two to three orders of magnitude faster than $l_1$magic which is a l_1 minimization solver.

In[4], the authors proposed the used of sparse representation with l_1 minimization and BP as classification algorithm for face recognition. The reported results have proved the robustness and discriminative properties of sparse representation classifier (SRC). In[5], the authors employed SRC as the classification method for speaker identification. Their experimental results based on TIMIT database show that SRC performed better in term of accuracy compared with GMM-SVM algorithms. Besides that, SRC was also implemented in[6] for heartbeats classification. The result shows that the SRC outperformed the SVM in classification of heartbeats from MIT-BIH ECG database. Karhunen-Loeve Transform (KLT) is another research direction in reducing computational complexity. This approach formalizes mathematical modelling by iteratively extracting suitable parameter collection[7-9].

On the other hand, a usual technique in machine learning called kernel tricks is also used by many researchers by kernelizing the non-linear feature space input into a linear high-dimensional kernel feature space to improve the classification accuracy. Such SRC with kernel tricks is called KSRC in literature[10-13]. This KSRC shows better performance compared to the SRC in many cases such as in face classification especially when the testing samples from different classes possess features that have same distribution direction. This is because the features in same distribution direction would be overlapped with each other after normalization in SRC. All of the SRC and KSRC implemented in the research discussed above are based on a sparse recovery approach known as relaxation method. Relaxation methods such as BP and Iterative Re-weighted Least Square minimization (IRLS) are very successful approaches to recover the sparse solution from the SRC and KSRC. Nonetheless, the computation time of SRC and KSRC based on relaxation methods is the main drawback of this algorithm for being implemented in real time application.

In this paper, a classification algorithm based on kernel trick and sparse representation using improved sparse signal recovery solver is developed. The variation on sparse signal recovery solvers in smoothed l_0 algorithm developed by[3] and a modified version in[14] is performed based on two objectives which is firstly to further reduce the computation time for solving high-dimensional kernel space sparse representation classification problems and secondly, to maintain or even surpass the classification ability of the SRC. The performances of the proposed algorithm are then compared experimentally with the baseline algorithms by varying the numbers of training samples and feature dimensions. The paper is organized as follows. Section 2 discusses the relatively new algorithm to find sparse solution, SL0 and its modified version, SL0-mod. The new variation based on the SL0 and SL0-mod known as SL02 is introduced in this section. In section 3, the SRC and KSRC classification algorithms incorporating SL02 are discussed. The result and discussion are illustrated in section 4. Experiments are carried out to compare the proposed classification method with other existing method. Finally, section 5 summarizes conclusions.

2. Smoothed l_0 norm and modified version

SL0 is a relatively new approach to solve the sparse representation problem which approximates the l_0 norm using the continuous zero-mean Gaussian function and uses gradient method such as steepest ascent to minimize the continuous function. The approximation of discontinuous function of l_0 norm into a continuous one is as follows:

$$f_\sigma(x) \triangleq \exp(-\frac{x^2}{2\sigma^2}) \tag{3}$$

The accuracy of this approximations is controlled by the parameter σ. When σ is large, the approximation is smooth, whereas when σ is very small or approaching zero, the Eq. 3 gives better approximation to l_0 norm as follows:

$$\sum_{i=1}^{N} \lim_{\sigma\to 0}\left(1 - f_\sigma(x_i)\right) = \|x\|_0 \tag{4}$$

$$F_\sigma(x) = \sum_{i=1}^{N} f_\sigma(x_i) \tag{5}$$

where N is the number of column of the dictionary,D. According to Eq. 4 and Eq. 5, in order to minimize the l_0 norm, the $F_\sigma(x)$ should be maximized. However, the small σ will result an approximation which is non-smooth and contains many local maximum. To avoid this local maxima, a decreasing value of σ gradually. For each value of σ, a fixed L steps of steepest ascent approach is used to maximize the $F_\sigma(x)$. In each steepest ascent iteration, the solution moves in the desired direction as follows:

$$\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}} + \mu_k \nabla F_\sigma(x) \tag{6}$$

where $\mu_k = \mu\sigma^2$ and μ_k is fixed and small in value. The solution x is then projected back onto the feasible set as follows:

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \boldsymbol{A}^{+}(\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{y}) \tag{7}$$

The SL0 algorithm is summarized in Algorithm 1.

In[14], the SL0-mod algorithm proposed based on three modification on the SL0 algorithm which are:
1) Modify the initial σ value in algorithm 1 by introducing a new relation as follows:

$$\sigma_{initial} = \frac{1}{2.75\left(\frac{n}{N}\right)} \max|\hat{\boldsymbol{x}}| \tag{8}$$

where n and N is the number of row and column of dictionary $\boldsymbol{A}$ respectively, and σ_k is increase from 0.5 to 0.7.
2) Instead of using constant L steps in steepest ascent approaches, the authors suggested the used of $L = 2$ initially and multiply it with $L_k = 2$ after each update of $\boldsymbol{\sigma}$. This modification is made because σ decreases, the accuracy of the Gaussian approximation increases and more steepest ascent steps is required for the solution to converge.
3) The replacement of constant step-size μ_k with a variable step-size as follows:

$$\mu(\sigma_j, j) = \begin{cases} \mu_{start}, & j < 4 \; or \; \sigma_j > 0.75\max|x| \\ 1.5, & otherwise \end{cases} \tag{9}$$

Where j is the current number of steepest ascent iteration, and

$$\mu_{start} = \begin{cases} 0.05, & \frac{n}{N} \leq 0.5 \\ 0.001, & otherwise \end{cases} \tag{10}$$

Besides the three modification, the authors also suggested that the inner loop of the steepest ascent algorithm is terminated when the solution is sufficiently close to convergence. This changes is accomplished to allow the increasing value of σ_k and L while maintaining the reconstruction time. The termination condition is as follows:

$$\left\|\hat{\boldsymbol{x}}_{\boldsymbol{j}} - \hat{\boldsymbol{x}}_{\boldsymbol{j-1}}\right\|_2 < \sigma_j \cdot \epsilon \tag{11}$$

Where $\epsilon = 0.01$. The SL0-mod algorithm is summarized in Algorithm 2.

```
Algorithm 1 – smoothed $l_0$ norm (SL0)
1. Initialization : $\sigma_k = 0.5$, $\sigma_{min} = 0.001$, $L = 3$ , $\mu = 2$, -
$\hat{\boldsymbol{x}} = \boldsymbol{A}^{+}\boldsymbol{y}$ and $\sigma = 2 \max|\hat{\boldsymbol{x}}|$
2. Main loop:
while $\sigma > \sigma_{min}$ then executes the following for loop.
        for $j = 1, \ldots\ldots, L$ do
                $\nabla F_\sigma(\boldsymbol{x}) = \hat{\boldsymbol{x}} \cdot \exp(-\hat{\boldsymbol{x}}^2/2\sigma^2)$
                $\hat{\boldsymbol{x}} \quad = \hat{\boldsymbol{x}} - \mu_k \cdot \nabla \boldsymbol{F}_{\boldsymbol{\sigma}}(\boldsymbol{x})$
% steepest ascent gradient step
                $\hat{\boldsymbol{x}} \quad = \hat{\boldsymbol{x}} - \boldsymbol{A}^{+}(\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{y})$
 % Project $\hat{x}$ back onto the feasible set
        end for
        $\sigma = \sigma \cdot \sigma_k$        % reduce sigma value by $\sigma_k$ times.
        end while
3. Output = $\hat{\boldsymbol{x}}$
```

```
Algorithm 2 – improved smoothed $l_0$ norm (SL0-mod)
1. Initialization : $\sigma_k = 0.7$, $\sigma_{min} = 0.001$, $L = 2$, $L_k = 2$ ,
$\mu = 2$, $\hat{\boldsymbol{x}} = \boldsymbol{A}^{+}\boldsymbol{y}$, $\epsilon = 0.01$, and $\sigma = \frac{1}{2.75 \cdot \frac{n}{N}} \cdot \max|\hat{\boldsymbol{x}}|$
% $n$ and $N$ is the number of row and column of dictionary $\boldsymbol{A}$
respectively.
2. Main loop:
while $\sigma > \sigma_{min}$ then executes the following for loop.
        for $j = 1, \ldots\ldots, L$ do
        $\hat{\boldsymbol{x}}_{\boldsymbol{prev}} = \hat{\boldsymbol{x}}$
        $\boldsymbol{\Delta}_{\boldsymbol{j}} \quad = \nabla F_\sigma(\boldsymbol{x}) = \hat{\boldsymbol{x}} \cdot \exp(-\hat{\boldsymbol{x}}^2/2\sigma^2)$
        $\hat{\boldsymbol{x}} \quad = \hat{\boldsymbol{x}} - \mu_k \cdot \boldsymbol{\Delta}_{\boldsymbol{j}}$
% steepest ascent gradient step
        $\hat{\boldsymbol{x}} \quad = \hat{\boldsymbol{x}} - \boldsymbol{A}^{+}(\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{y})$
% Project $\hat{x}$ back onto the feasible set
                if $\left\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{\boldsymbol{prev}}\right\|_2 < \sigma \cdot \epsilon$ then
                        break
                end if
        end for
        $\sigma = \sigma \cdot \sigma_k$ % reduce sigma value by $\sigma_k$ times.
        $L = L \cdot L_k$ % increase looping number by $L_k$times.
end while
3. Output = $\hat{\boldsymbol{x}}$
```

3 Improvement and Variations of SL0 and SL0-mod (SL02)

In this paper, a variation is done on the SL0-mod algorithm which a $\boldsymbol{\beta}$ variables is added to the calculation of the successive adjustment of the gradient direction and it is computed as follows:

$$\boldsymbol{\beta} = \frac{\nabla F_\sigma(\boldsymbol{x_j})^{\boldsymbol{T}} \nabla F_\sigma(\boldsymbol{x_j})}{\nabla F_\sigma(\boldsymbol{x_{j-1}})^{\boldsymbol{T}} \nabla F_\sigma(\boldsymbol{x_{j-1}})} \tag{12}$$

where $\nabla F_\sigma(\boldsymbol{x_j})$ and $\nabla F_\sigma(\boldsymbol{x_{j-1}})$ is the gradient in current and previous iteration of steepest ascent respectively. This idea is inspired from Conjugate gradient approaches which is a better alternative method of steepest descent. From the variation, a degree of robustness is built into the method by incorporating the gradient information from the previous step into the adjustment of gradient direction.

The $\boldsymbol{\beta}$ value will be multiple with the previous gradient value and the result will be added to current gradient value as follows:

$$\nabla F_{\sigma,\beta}(\boldsymbol{x_j}) = \nabla F_\sigma(\boldsymbol{x_j}) + \boldsymbol{\beta} \nabla F_\sigma(\boldsymbol{x_{j-1}}) \tag{13}$$

where $\nabla F_{\sigma,\beta}(\boldsymbol{x_j})$ indicates the resultant gradient step that taking consideration of both current and previous gradient into the optimization algorithm. This improvement encourage the sparse solution to converge accurately even if the number of loop L for steepest ascent algorithm is kept constant.

When the number of training samples, N or atoms in the dictionary increase or the training sample dimensions decrease, the σ value in the SL0-mod and SL02 will increase based on Eq. 8 and causing more loops is needed for the algorithm to find the sparse solution. This situation will increases the computation time of the algorithm. To overcome with this, the number of loops L which is kept constant as in Algorithm 3 instead of increasing its value by multiplied with a constant after each update of σ in SL0-mod.

```
Algorithm 3 – improved smoothed $l_0$ norm (SL02)
1. Initialization : $\sigma_k = 0.7$, $\sigma_{min} = 0.001$, $L = 2$, $\mu = 2$, -
$\hat{\boldsymbol{x}} = \boldsymbol{A}^{+}\boldsymbol{y}$, $\epsilon = 0.01$, and $\sigma = \frac{1}{2.75 \cdot \frac{n}{N}} \cdot \max|\hat{\boldsymbol{x}}|$
% $n$ and $N$ is the number of row and column of dictionary
$\boldsymbol{A}$ respectively.
2. Main loop:
while $\sigma > \sigma_{min}$ then executes the following for loop.
    for $j = 1, \ldots\ldots, L$ do
        $\hat{\boldsymbol{x}}_{\boldsymbol{prev}} = \hat{\boldsymbol{x}}$
        $\boldsymbol{\Delta_j} = \nabla F_\sigma(\boldsymbol{x}) = \hat{\boldsymbol{x}} \cdot \exp(-\hat{\boldsymbol{x}}^{\boldsymbol{2}}/2\sigma^2)$
        if $j > 1$ then
            $\boldsymbol{\beta} = \boldsymbol{\Delta}_{\boldsymbol{j}}^{\boldsymbol{T}}\boldsymbol{\Delta}_{\boldsymbol{j}}/\boldsymbol{\Delta}_{\boldsymbol{j-1}}^{\boldsymbol{T}}\boldsymbol{\Delta}_{\boldsymbol{j-1}}$
            $\boldsymbol{\Delta_j} = \boldsymbol{\Delta_j} + \boldsymbol{\beta}\boldsymbol{\Delta_{j-1}}$
% improved step
        end
        $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \mu_k \cdot \boldsymbol{\Delta_j}$
% steepest ascent gradient step
        $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \boldsymbol{A}^{+}(\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{y})$
% Project $\hat{x}$ back onto the feasible set

        if $\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{\boldsymbol{prev}}\|_2 < \sigma \cdot \epsilon$ then
            break
        end if
    end for
    $\sigma = \sigma \cdot \sigma_k$ % reduce sigma value by $\sigma_k$ times.
end while
3. Output = $\hat{\boldsymbol{x}}$
```

4 Classification

4.1 Sparse Representation Classifier

Sparse representation is originally applied for signal representation and reconstruction. Sparse representation of signal is an expression of the signal as a linear combination of atoms in an over-complete dictionary in which many of the coefficients are zero. The original goal of sparse representation is to represent and compress a signal using lower sampling rates than the Shannon-Nyquist rate[15]. Thus, the performance of the compress algorithm is based on the degree of sparsity of the representation to the original signal. Among all the atoms in an over-complete dictionary, the sparse representation selects the subset of the atoms which most compactly expresses the input signal and rejects all other less compact representation. Therefore, the sparsest representation of a signal is naturally discriminative and can be developed for signal classification purpose.

Sparse representation classifier is a nonparametric learning method which can directly predict or assign a class label to a test sample based on dictionary composed of training samples. This method is similar to Nearest Neighbor and Nearest Subspace classifier which do not have a training process in its classification process. Sparse representation for classification is first introduced in 2009 in face recognition problem[4]. They experimentally proved that sparse representation classifier (SRC) has better classification performance than the nearest neighbor and nearest subspace. In sparse representation classifier, the dictionary is constructed from training samples from various classes. Eq. 1 shows the input signal y is linearly represented by a dictionary, A and sparse representation, x. The j^{th} class training samples are arranged as column of a matrix A_j as shown in Eq. 14. The columns of dictionary is refered as atoms.

$$A_j = \left[a_{j,1}, \dots, a_{j,n_j}\right] \in R^{m \times n_j} \tag{14}$$

where $a_{j,i}$ denotes the training sample belonging to the j^{th} class, and n_j is the number of the training samples for j^{th} class. The dictionary, A is formed using the dictionary from each class as shown in Eq. 15

$$A = [A_1, A_2, \dots, A_c] \in R^{m \times n} \tag{15}$$

where $n = \sum_{j=1}^{c} n_j$. and c is the number of class.

Based on the Eq. 1, for SRC problem, the sparse representation x, is the vector of coefficients associated with the training sample in the dictionary matrix. The entries of x that is corresponding to the class which the test sample y belongs to is expected to be nonzero while the entries of x that corresponding to other classes is expected to be zero.

$$x = \left[0, \dots, 0, x_{j,1}, \dots, x_{j,n_j}, 0, \dots., 0\right]^T \tag{16}$$

where $x_{j,i} \in R$ is the coefficient corresponding to the training sample $A_{j,i}$. The sparse representation based classification method looks for the sparsest representation by solving the following l_0 minimization problem as shown in Eq. 2. This equation is known as NP hard problem and difficult to approximate. The developed theory from sparse representation and compressive sensing research reveals that the sparsest solution from Eq. 2 can be obtained by replacing the l_0 norm with the l_1 norm given that the solution, x is sparse enough[16-18]. Such approaches is called Relaxation method.

In[19], the authors used SL0 algorithm to recover the sparse solution in the SRC to classify the gender based on the ear image. The classification accuracy of 89.49% is achieved. In this paper, the SL02 algorithm discussed in section 2 will be implemented to complete the task of finding sparse solution for the classification based on SRC. From the sparse solution obtained, it can be noticed that most of the non-zero entries in the solution are associated with the atoms of a class which the test sample y belongs to. The minimum of the representation error or the residual error of each class is calculated by keeping the coefficients associated with that class and while setting the other entries to zero. This is done by introducing a characteristic function, ζ as follow,

$$r_c(y) = \|y - A\zeta_i x\|_2 \tag{17}$$

where $r_c(y)$ denotes the residual error. The vector ζ has value one at locations associated to the class i and zero for other entries. The class, d, of the test signal, y is computed as the one that produces smallest residual error.

$$d = \min_i r_i(y) \tag{18}$$

The algorithm summarizes the complete classification procedure of SRC is shown below

```
Algorithm 4 –Sparse Representation Classification method
Step1: The input for SRC are a matrix of dictionary constructed of training samples
        $A = [A_1, A_2, \dots, A_c] \in R^{m \times n}$
    for $c$ classes, a test sample $y \in R^m$ and an optional error tolerance $\epsilon > 0$.
Step 2: Normalize the atoms of $A$ to have unit $l_2$ norm.
Step 3: Compute the sparse solution, $x$ using SL02 algorithm described in Algorithm 3.
Step 4: compute the residuals $r_i(y) = \|y - A\zeta_i x\|_2 \; for \; i = 1, \dots \dots, c$.
Step 5: the class of the given test sample, y is determined by
        $identity(y) = \min_i r_i(y)$
```

4.2 Kernel sparse representation classifier

Kernel tricks is applied into a classifier to change the distribution of samples by mapping it into a high dimensional kernel feature space[20] in order to change the linear inseparable samples in the original feature space into linear separable in the high dimensional feature space. This means a test sample can be represented as the linear combination of training samples from same class more accurately by applying kernel trick into SRC. The classification performance of SRC will be improved as the nonzero entries of sparse representation, x of the test sample are more associated with training samples from same class as itself. In this work, radial basis function (RBF) kernel is employed in KSRC as following,

$$k(x, y) = e^{-t\|x-y\|^2} \tag{19}$$

where $t < 0$ is the parameter for RBF kernels. KSRC classification method is also able to overcome the disadvantages of SRC which cannot classify samples in the same direction which belong to different classes[8].

In KSRC, kernelized dictionary and testing sample is computed by the following equation,

$$A_{kernel} = \left[k(a_i, a_j)\right]_{n \times n} \tag{20}$$

$$y_{kernel} = \left[k(y, a_j)\right]_{n \times 1} \tag{21}$$

where $i, j = 1, \dots, n$, and n is the number of training samples. The kernelized dictionary dimension of the training samples is increased by kernel mapping as the number of atoms is smaller than feature dimension in original dictionary. The increase of dimension of dictionary and test samples after being kernelized means that more computation time is needed for SRC to find the sparse solution. In[11,12], dimensionality reduction is used to reduce dimension of the kernelized dictionary and test samples hence to moderate the complexity and computation time needed by SRC to recover the sparse solution. In this paper, the reduction of dimensionality of kernelized inputs is omitted as the proposed method SL02 which is able to compute sparse solution from high dimensional dictionary in a very short time. In KSRC, the classification task is executed by replacing A and y in SRC problem with A_{kernel} and y_{kernel}. The algorithm summarizes the complete classification procedure of KSRC is shown below.

```
Algorithm 5 –Kernel Sparse Representation Classification method
Step 1: The input for KSRC are a matrix of dictionary constructed of training samples
        $A = [A_1, A_2, \dots, A_c] \in R^{m \times n}$
    for c classes, a test sample $y \in R^m$ and an optional error tolerance $\epsilon > 0$.
Step 2: Kernelize of$A$ and $y$ to yield $A_{kernel}$ and $y_{kernel}$
Step 3: Normalize the atoms of $A_{kernel}$ and $y_{kernel}$ to have
```

unit l_2 norm.
Step 4: Compute the sparse solution, x using SL02 algorithm described in Algorithm 3.
Step 5: compute the residuals $r_i(y_{kernel}) = \left\| y_{kernel} - A_{kernel}\zeta_i x \right\|_2 \; for \; i = 1, \dots\dots, c.$
Step 6: the class of the given test sample, y is determined by

$$identity(y_{kernel}) = \min_i r_i(y_{kernel}).$$

5 Result and Discussion

Two experiments are carried out to evaluate the performance of the proposed algorithm by making comparison between the accuracy and computation time based on number of training data and feature dimension. The dataset used in these experiments are speech data from Audio-Visual Digit Database[21] which consists of speech ‘zero’ from 37 participants. The dataset consists of 60 samples of speech ‘zero’ from each of the participant. Both of the experiments are repeated for 20 times to take the mean accuracy and computation time. The computation time is based on the CPU time measured on a PC laptop with a Intel i7-3537U 2.50 GHz processor using MATLAB commands *tic* and *toc.* In the first experiment, MFCC method is used to extract the features from the speech ‘zero’ and the features dimension is set to 144 length. These features samples is divided into training samples as well as testing samples. Three different number of training samples are used in this experiment which are 10, 15 and 20 training samples from each participant while 40 testing samples from each participant are used.

In the second experiment, the variable is the features dimension extracted using MFCC method. Four different features dimension are used which are 64,144,256 and 400 while keeping the number of training samples and testing samples constant which is 20 and 40 from each participant respectively. The purpose of the experiment is to test the effect of feature dimension, n, on the computation time and accuracy of the SRC algorithms.

Based on Table 1, the SRC classifier improved with the increase number of training sample from each participant. This is because as the number of training sample increases, more information about each class is possessed by the algorithm to build dictionary. Besides that, the highest accuracy achieved is 99.2837% by KSRC using SL0, SL0-mod and SL02. However, they perform differently in computation time with KSRC using SL02 the fastest which is 25.59 second. This is due to the improvement and variation made to the SL0 algorithm describe in section 2.

The computation time of the algorithms show an increasing trend with the increase of training samples. This is because when the number of atom of the dictionary increases, the dimension of the sparse code is also increased and therefore more time is needed for the SRC to compute the sparse solution. The shortest computation time in the experiment is achieved by SRC using SL02 which is 5.028s, 5.921s and 8.032s in 10, 15 and 20 training samples respectively. The accuracy of this algorithm is also high and comparable to SRC using BP used by[4] and SRC using SL0 used by[19] which needs 36.63s and 10.11s for its computation respectively. Most of the KSRC is normally slower than its SRC counterpart because increase of dimensionality in the row of dictionary after kernel mapping. This is noticeable when comparing the SRC and KSRC which using SL0 and SL02. While there is an exception for SRC and KSRC using SL0-mod. The longer computation time of these SRC is due to update of L step on each update of σ in SL0-mod as shown in section 2.

In experiment 2, the accuracy of classification achieved by all of the classifiers is maximum when the feature dimension is equal to 144. From Table 2, the SRC and KSRC using SL02 achieved fastest time compared to other SRC and KSRC algorithm while maintaining the accuracy as in other SRC and KSRC.

Table 1 The accuracy and computation time of classifiers based on various number of training sample.

	Number of Training Sample					
	10		15		20	
Classifier	ACC	Time	ACC	Time	ACC	Time
SRC using BP	97.8141	30.13	98.4932	32.85	98.8108	36.63
SRC using SL0	96.9087	6.260	98.3682	7.822	98.9932	10.11
SRC using SL0-mod	97.0709	49.24	98.3682	63.12	98.9020	82.52
SRC using SL02	**97.5709**	**5.028**	**98.5304**	**5.921**	**98.9324**	**8.032**
KSRC using BP	95.6858	33.13	97.0641	43.21	97.6858	68.72
KSRC using SL0	98.3581	9.848	98.9831	23.03	99.2837	54.15
KSRC using SL0-mod	98.3581	6.697	98.9831	13.09	99.2837	28.83
KSRC using SL02	**98.3581**	**6.455**	**98.9831**	**11.08**	**99.2837**	**25.59**

Table 2. The accuracy and computation time of classifiers based on various feature dimension of speech sample.

	Feature Dimension							
	64		144		256		400	
Classifier	ACC	Time	ACC	Time	ACC	Time	ACC	Time
SRC using BP	98.3716	33.94	98.8108	36.63	98.6317	36.29	98.6250	47.26
SRC using SL0	98.3513	6.761	98.9932	10.11	98.5101	12.62	98.2263	24.38
SRC using SL0-mod	98.0912	67.26	98.9020	82.52	98.5067	77.66	98.5101	14.81
SRC using SL02	**97.7331**	**5.972**	**98.9324**	**8.032**	**98.6418**	**8.241**	**98.5067**	**14.28**
KSRC using BP	92.5236	69.64	97.6858	68.72	97.6081	62.81	97.7871	65.61
KSRC using SL0	98.9425	54.00	99.2837	54.15	98.7972	49.85	98.3783	54.42
KSRC using SL0-mod	98.9425	28.30	99.2837	28.83	98.7972	25.59	98.3783	31.59
KSRC using SL02	**98.9425**	**23.58**	**99.2837**	**25.59**	**98.7972**	**20.88**	**98.3783**	**24.32**

6 Conclusion

In this paper, a variant of KSRC and SRC algorithms is proposed and their performances are evaluated with speech recognition problem dataset. From the result obtained, the computation time of SRC and KSRC using SL02 is greatly reduced compared to baseline systems while the accuracy is maintained. This indicates that with the improvement and variation on the SRC and KSRC, the purposed algorithm has advantage to be implemented for real time classification application.

Acknwledgements

This work was supported by Universiti Sains Malaysia under Fundamental Research Grant Scheme (6071266).

References

1. Gribonval R., Lesage S. A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In: ESANN'06 proceedings-14th European Symposium on Artificial Neural Networks; 2006. p. 323-330.
2. Donoho DL, Elad M, Temlyakov VN. Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Transactions on Information Theory 2006;**52**(1):6-18.
3. Mohimani GH, Babaie-Zadeh M, Jutten C. Fast Sparse Representation based on Smoothed l_0 Norm. In: Independent Component Analysis and Signal Separation, Springer; 2007. p. 389-396.
4. Wright J, Yang AY, Ganesh A, Sastry SS, Yi M. Robust Face Recognition via Sparse Representation. Pattern Analysis and Machine Intelligence, IEEE Transactions 2009;**31**(2):210-227. doi:10.1109/TPAMI.2008.79.
5. Naseem I, Togneri R, Bennamoun M. Sparse representation for speaker identification. In: Pattern Recognition (ICPR), 20th International Conference 2010. p. 4460-4463.
6. Huang HF, Hu GS, Zhu L. Sparse representation-based heartbeat classification using independent component analysis. Journal of medical systems 2012;**36**(3):1235-1247.
7. Biagetti G, Crippa P, Orcioni S. Speaker Identification with Short Sequences of Speech Frames. In Proceedings of the 4th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2015); 2015. 2. p. 178-185.
8. Crippa P, Curzi A, Falaschetti L, Turchetti C. Multi-Class ECG Beat Classification Based on a Gaussian Mixture Model of Karhunen-Loève Transform. In: International Journal of Simulation Systems, Science & Technology (IJSSST) 2015;**16**(1):2.1-2.10.

9. Gianfelici F, Turchetti C, Crippa P. A Non-Probabilistic Recognizer of Stochastic Signals Based on KLT, Signal Processing 2009;**89**(4):422-437.
10. Gao S, Tsang IW-H., Chia L-T. Kernel sparse representation for image classification and face recognition. In: Computer Vision–ECCV: Springer; 2010. p. 1-14.
11. Zhang L, Zhou W-D, Chang P-C, Liu J, Yan Z, Wang T, Li F-Z. Kernel sparse representation-based classifier. IEEE Transactions on Signal Processing 2012;**60**(4):1684-1695.
12. Yin J, Liu Z, Jin Z, Yang W. Kernel sparse representation based classification. Neurocomputing 2012;**77**(1):120-128.
13. Jian M, Jung C. Class-Discriminative Kernel Sparse Representation-Based Classification Using Multi-Objective Optimization. IEEE Transactions on Signal Processing 2013;**61**(18):4416-4427.
14. Oxvig CS, Pedersen PS, Arildsen T, Larsen T. Surpassing the Theoretical l-Norm Phase Transition in Compressive Sensing by Tuning the Smoothed l0 Algorithm. In: IEEE International Conference on Acoustics, Speech, and Signal Processing; 2013. p. 6019-6023.
15. Candès EJ, Wakin MB. An introduction to compressive sampling. IEEE Signal Processing Magazine 2008;**25**(2):21-30.
16. Candès EJ, Romberg JK, Tao T. Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics 2006; **59**(8): 1207-1223. doi:10.1002/cpa.20124.
17. Donoho DL. For most large underdetermined systems of linear equations the minimal Communications on Pure and Applied Mathematics 2006; **59**(6): 797-829. doi:10.1002/cpa.20132.
18. Candes EJ, Tao T. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? IEEE Transactions on Information Theory 2006; **52**(12):5406-5425. doi:10.1109/TIT.2006.885507.
19. Rahman K. Gender classification using 2-D ear images and sparse representation. In: Mohamed, A-M. (ed.); 2013. p. 461-466.
20. Yu K, Ji L, Zhang X. Kernel nearest-neighbor algorithm. Neural Processing Letters 2002; **15**(2): 147-156.
21. Sanderson C, Paliwal KK. Noise compensation in a multi-modal verification system. In: IEEE International Conference Acoustics, Speech, and Signal Processing Proceeding (ICASSP '01); 2001. **151**. p. 157-160.