Hyperspectral Data Classification via Sparse Representation in Homotopy

Qazi Sami ul Haq,Lixin Shi,Linmi Tao,Shiqiang Yang
Key Laboratory of Pervasive Computing, Ministry of Education
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

***Abstract*— Sparse representation has significant success in many fields such as signal compression and reconstruction but to the best of our knowledge, no sparse-based classification solution has been proposed in the field of remote sensing. One of the reasons is that the general optimizers are extremely slow, time consuming and needs intensive processing for l_1-*minimization* sparse representation. In this paper, we propose a fast sparse representation using l_1-*minimization* based on *homotopy* for the classification of hyperspectral data. This method is based on the observation that a test sample can be represented by train samples from a pool of large number of train samples i.e the sparse representation. Hence the sparse representation for each test sample is achieved by the linear combination of the train samples. This proposed method has the advantages that learning on the training samples is not required, both model selection and parameter estimation are not needed, and low computational load, by which a bagging algorithm is introduced to increase the classification accuracy via voting. A real hyperspectral dataset (AVIRIS 1992 Indiana's Indian Pines image) is used to measure the performance of the proposed algorithm. We compared the accuracy results with state-of-the-art SVM and general purpose linear programming solvers. We also presented a time comparison between our approach and general LP solvers. The comparisons prove the effectiveness of the proposed approach.**

***Index Terms*—Remote sensing, hyperspectral data,sparse representation,homotopy,classification.**

I. Introduction

Hyperspectral imaging is excellent source of information for the classification of materials. Hyperspectral sensors contain hundreds of spectral channel where each channel covers a small portion of electromagnetic spectrum. This is in sharp contrast to multispectral sensors which have limited number of spectral channels with each channel covering the large portion of the electromagnetic spectrum. Such large number of channels in hyperspectral sensors provide excellent information which helps in discriminating large number of materials as different materials have different response in different channels, so more channels increases the ability to distinguish one materia from another.

In literature, many algorithms have been proposed for classification of hyperspectral data like unsupervised classification algorithms [13] which have the advantage of no requirement of train samples but the clusters which are produced by the algorithms can hardly be mapped to the actual classes. There are supervised classification algorithms like maximum likelihood classifier, which does not work for few train samples and ANN [5] but these require quite a few train samples and needs low dimensional data to give good results. SVM [11] is another supervised method for hyperspectral data classification that is known as state-of-the-art. SVM not only gives excellent results but work with high dimensional data as well.

Sparse representation is an effective model which has a lot of real life application e.g in image compression and reconstruction as when right basis are used to examine it then many of the coefficients may be unnecessary and hence the image size can be reduced without sacrificing the image quality. Well-known transforms such as discrete wavelet transform represents a signal uniquely but on the other hand the redundant dictionaries also knows as redundant systems do not represent a signal uniquely. Finding the sparsest or almost sparsest in a redundant system is called sparse representation In the recent years many sparse based solutions have been used successfully in areas like signal reconstruction [2, 3, 7], face recognition [12] and others but to the best of our knowledge, this is the first time that sparse classification is being proposed for the classification of hyperspectral data. One reason for absence of sparse based solution in remote sensing may be that while the general purpose LP solvers can compute the l_1-*minimization* for sparse representation but the algorithms are extremely slow and not suitable for the application on remote sensed data.

In this paper, we propose a fast sparse representation using l_1-*minimization* based on *homotopy* for the classification of hyperspectral data. Homotopy was first proposed by Osborne et al [10]. We have used it here for the classification purposes and we will prove that the *homotopy* algorithm runs much faster than the general LP solvers while the accuracy is still comparable to the general purpose LP solvers. The *homotopy* algorithms offers the advantage that if the underlying solution has *k* non-zeros, the homotopy algorithm achieves that solution in *k* iterative steps [6] i.e the k-step property. Our approach is based on the observation that a test sample can be represented by few features from a pool of large number of features i.e the sparse representation. Hence the sparse representation for each test sample is achieved by the linear combination of few train samples by using l_1-*minimization* based on *homotopy*.

Our extensive experiments on real hyperspectral dataset show that the proposed approach offers more classification accuracy than state-of-art methods i.e SVM and semi-supervised graph based approaches. Our algorithm has very low computational load, and a bagging algorithm is introduced to improve the classification accuracy. The results proved the efficiency and enhanced accuracy of the bagging algorithm.

978-1-4244- 7618-3 /10/$26.00 ©2010 IEEE

In addition we also performed time comparisons of homotopy based approach with general LP solvers which clearly show that homotopy based approach is much faster.

The rest of paper is organized as follows. Section II describes the sparse representation and homotopy based proposed approach. Section III presents the experimental results and comparisons. Finally the section IV concludes the paper.

II. Proposed Approach

The proposed homotopy-based approach, calculates the sparse representation of each test sample using the train samples and the classification of the test samples is based on *Sparse Assumption: Sparse representation will be the same for the test samples belonging to same class and the sparse representation will be different for test samples belonging to different classes.*

A. Sparse representation

Suppose we have dictionary of training samples (a_i, c_i), a_i are the *d*-dimensional column vectors which represent hyperspectral pixels. The *d* dimensions corresponds to hyperspectral bands and c_i are the corresponding class labels of the hyperspectral train pixels. All the training samples from all classes are concatenated in in the matrix A

$$A = [a_1, a_2, \cdots, a_n] \tag{1}$$

and we assume that a test sample y may be represented by linear combination of training samples from the same class:

$$y = \alpha_1 a_1 + \alpha_2 a_2 + \cdots + \alpha_n a_n \tag{2}$$

which can be written as

$$y = A\alpha \tag{3}$$

To find the sparse representation of any test sample y, a coefficient vector α needs to be calculated

$$\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]^T \text{ subject to } y = A\alpha \tag{4}$$

Ideally the coefficient vector should only contain those non zero entries which are associated with the class of test sample y. To calculate this sparsest coefficient vector we need to solve the optimization problem:

$$\alpha_0 = \min \| \alpha \|_0 \text{ subject to } y = A\alpha \tag{5}$$

Where y is an observed test sample, A is known matrix containing train samples from all classes,α is an unknown vector and $\| \alpha \|_0$ represents the number of non-zero components in the coefficient vector α. This is not a convex optimization problem and generally its solution is NP-hard [6]. So the problem can be solved using l_1-norm, which is a convex optimization problem:

$$\alpha_1 = \min \| \alpha \|_1 \text{ subject to } y = A\alpha \tag{6}$$

In real life, its almost always the case that the observed signals are corrupted by noise, so we add the noise term η in the eq. 3 as:

$$y = A\alpha + \eta \qquad \text{such that} \quad \| \eta \|_2 < \epsilon \tag{7}$$

so the sparse solution α can be found out using the l^1-minimization as:

$$\alpha_1 = \text{ argmin } \| \alpha \|_1 \text{ subject to } \| A\alpha - y \|_2 \leq \epsilon \tag{8}$$

Whereas l_1-norm can be described as

$$\| \alpha \| = \Sigma_i \mid \alpha_i \mid$$

B. Homotopy

Although solving sparse representation has been reduced to a ℓ_1-norm optimization problem, solving it efficiently is still a challenging problem for researchers. Naive methods includes solving ℓ_1-norm as linear programming, resulting in an iterative basis pursuit algorithm[4]. However, this optimization doesn't satisfy the demanding speed of many applications. Following this thread, other approximation solving methods have been proposed, such as the orthogonal matching pursuit method [9] and the alternating direction algorithm[14]. In this context, we will use a fast and robust method, namely Homotopy, specifically designed for ℓ_1-norm optimization problem which is supposed to have sparse solutions[8]. The high efficiency of this algorithm relies on the sparsity of the solution.

Theoretically based on the sparseland assumption, only the entries corresponding to the class of $\boldsymbol{y}$ can be nonzero. That indicates that Homotopy is expected to be efficient here in our application, and it is verified by our experiment results. The outline of Homotopy is iteratively calculating the solution α of equation (3) until it is under a predefined precision based on the initial value $\alpha = \mathbf{0}$.

It is proved that the case of solving equation (6) is equivalent to solving

$$\min_{\alpha} f_\lambda(\alpha) = \min_{\alpha} \left\{ ||y - A\alpha||_2^2 + \lambda ||\alpha||_1 \right\} \tag{9}$$

Specifically we have when $\lambda \to 0$, the solution of $\max_x f_\lambda(\alpha)$ converges to the solution of (6). [8] Therefore, we have reduced the problem of (6) to that of solving (9).

To find the minima, it's necessarily $\partial_\alpha f_\lambda(\alpha_\lambda) = 0$. We can easily calculate the partial difference of $f_\lambda(\alpha)$:

$$\partial_\alpha f_\lambda(\alpha_\lambda) = -A^T(y - A\alpha_\lambda) + \lambda\mu(\alpha_\lambda) \tag{10}$$

where $u(\alpha) = \partial ||\alpha||_1 \in \mathbb{R}^n$, the i-th element of it is defined as:

$$u_i(\alpha) = \begin{cases} \text{sgn}(\alpha_i) & \alpha_i \neq 0 \\ \epsilon \in [-1, 1] & \alpha_i = 0 \end{cases}$$

In order to simplify equation (10), we can define indicator set $T = \{i : \alpha_\lambda(i) \neq 0\}$, i.e., the nonzero entries, and residual

correlations $c = A^T(y - A\alpha_\lambda)$. In this way, equation (10) can be rewritten as

$$c(I) = \lambda \cdot \text{sgn}(\alpha_\lambda(I)), |c(I^c)| \leq \lambda \tag{11}$$

At the l-th stage, homotopy computes the update direction d_l by solving

$$A_I^T A_I d_l(I) = \text{sgn}(c_l(I)) \tag{12}$$

and the elements which is not in I of d are set to zeros. Equation 11 serves as a main indicating condition of updating I when updating α. It's not hard to verify that two scenarios will lead to a violation of equation 11, the occur when:

$$r_l^+ = \min_{i \in I^c} \{\frac{\lambda - c_l(i)}{1 - a_i^T v_l}, \frac{\lambda + c_l(i)}{1 + a^T v_l}\} \tag{13}$$

where $v_l = A_I d_l(I)$, or

$$r_l^- = \min_{i \in I} \{-\frac{x_l(i)}{d_l(i)}\} \tag{14}$$

For the purpose of convenience, define

$$r_l = \min\{r_l^+, r_l^-\} \tag{15}$$

Now we can update I by appending index i^+ that makes 13 hold when $r_l^+ \leq r_l^-$; otherwise we append i^- to I. At the same time, we can update α, using equation 16:

$$\alpha_l = \alpha_{l-1} + r_l d_l \tag{16}$$

The last one we have to update is λ, i.e., $\lambda_l = \lambda_{l-1} - r_l$.

The termination condition of Homotopy can be set in two ways:

- Using a predefined small value λ_0, the algorithm terminates when $\lambda_l \leq \lambda_0$
- The algorithm terminates by monitoring the difference of f between two iterations.

The outline of the algorithm is shown in algorithm 1.

Algorithm 1 Homotopy Algorithm

1: **initialize:** $\alpha \leftarrow 0, I \leftarrow \emptyset$
2: {we can choose other termination condition:}
3: **while** $\lambda > \lambda_0$ **do**
4: Compute $r_l^+, r_l^-, i^+, i^-, r_l$ by equation (13), (14), (15)
5: Compute d_l by equation (12)
6: Update I by appending i^+ or i^-
7: Update $\lambda \leftarrow \lambda - r_l$
8: Update $\alpha \leftarrow \alpha + r_l d_l$
9: **end while**

The homotopy algorithm has a complexity of $O(t^3 + n)$ where t is the number of nonzero elements in α. Hence we can see that it largely deduces the highest order item from n^3 to t^3; if the number of nonzero elements is close to be a constant, then homotopy will be near a linear speed. There are also recognizable benefits when the solution is not so sparse, as in our case, we can see a great time improvement in our experiment.

The improvement of time efficiency is not at the cost of largely drop in accuracy. The Homotopy algorithm is theoretically correct and precise by proving the equivalence of equation (9), and the practical precision is decided by the termination condition. We can manually set it to a level that it has a similar precision to basis pursuit. Experiments show that the classification accuracy is at least, if not higher than, similar to that of original sparse solution.

Now we get a new test sample y and we compute its sparse representation by algorithm 1 using the training samples of all classes which are stacked in the matrix *A*. Let $\hat{\alpha}$ is the sparse representation computed and ideally all the non zeros entries in the coefficient vector $\hat{\alpha}$ are associated with training samples in *A* that belong to single class and we can assign the test sample y to that class. But in practice, it rarely happens like that [12] as the noise makes the non zero entries in the coefficient vector $\hat{\alpha}$ associated with multiple classes. So there is need to define some objective function to cop with that. So we define a function $h(y)$:

$$h_i(y) = \| y - A\hat{\alpha}_i \|_2 \quad i = 1, \cdots, n \tag{17}$$

Where we get $\hat{\alpha}$ by getting only those values in coefficient vector $\hat{\alpha}$ which are associated with class i and then we generate the test vector $\hat{y}$ by $\hat{y} = A\hat{\alpha}$ and then we subtract it from the actual test vector y. So the class for which this objective function is minimized ,we assign the test vector to that class. Experiments show that this approach is much better than other approaches like e.g assigning the test vector to the class which has maximum entry in the coefficient vector $\hat{\alpha}$.

Algorithm 2 Sparse representation based classification

Input: training samples,labels (x_i, c_i) $i = 1, \cdots, n$ and test vector y.
1: Normalize all the training samples x_i and test sample y.
2: Solve the l_1-minimization problem using homotopy algorithm 1.
3: Compute $h_i(y), i = 1, \cdots, n$.
Output: $\text{argmin}_i \; h_i(y)$.

C. Sparse Bagging

In real-life the sparse representation for classification does not work perfectly due to the problems such as noise in the data, the lack of training samples, the incomplete searching of homotopy, etc. To tackle these problems, a bagging algorithm is proposed which is based on voting of individual sparse classifiers. In *Bagging Classification Algorithm*(algorithm 3), the individual sparse classifiers are used for the classification of each test sample. In this algorithm we keep a pool $\mathcal{P}$ of labeled samples, initially $\mathcal{P}$ is the training set. At the same time, we maintain the undetermined test set $\mathcal{T}$ which is initialized to be the test set. The process contains three global iterations. In each of global iterations there are seven subiterations i.e one for each individual sparse classifier. Each

individual classifier gets a random input from $\mathcal{P}$. In the first global iteration, seven individual sparse classifiers predict the output of each test sample $y_i \in \mathcal{T}$ and the results for each test sample is stored as $c_{i1}, c_{i2}, \cdots, c_{i7}$. After the completion of first global iteration, voting is performed for the seven sparse classifiers and a threshold θ is defined to determine the majority vote. If there is a class label c satisfying $\{j : c_{ij} = c\}$ has at least θ elements, it means there is *consensus* and we assign label the class of y_i as c, remove y_i from set $\mathcal{T}$; otherwise we keep the sample in $\mathcal{T}$, hence in this way the non-agreed test samples are given more weight.

Algorithm 3 Bagging classification

Input: set of training samples pool $\mathcal{P}$ and test vectors $\mathcal{T}$.

initialize: Normalize all vectors in $\mathcal{P} \cup \mathcal{T}$
while $\mathcal{T} \neq$ **and** number of iterations ≤ 3 **do**
  for $y \in \mathcal{T}$ **do**
    for each sparse classifier $1 \leq i \leq 7$ **do**
      randomly choose training set from $\mathcal{P}$
      classify y to c_i
    end for
    if $vote(c_1, \cdots, c_7) \geq \theta$ **then**
      $Class(y) \leftarrow major(c_1, \cdots, c_7)$
      $\mathcal{T} \leftarrow \mathcal{T} - \{y\}$
    end if
  end for
  $y^{(c)} = \text{argmin}_{y \notin \mathcal{T}} h_i(y)$
  $\mathcal{P} \leftarrow \mathcal{P} \bigcup_c y^{(c)}$
end while

Output: Class(y)

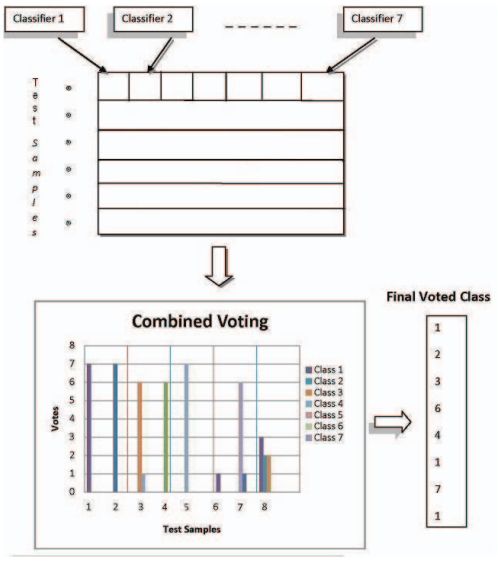

Fig. 1. Workings of Bagging algorithm

III. Experiments and Discussion

In this section, we performed extensive experiments to evaluate the effectiveness of the proposed approach. A well known hyperspectral dataset i.e AVIRIS 1992 Indian Pines image is used for the experiments and comparisons. It has 145*145 pixels and 220 spectral channels. Noisy channels(104-108,150-163,220) due to water absorption were not used, in the end 200 channels are used in total. The total classes available are 16 but due to insufficient training samples, 7 classes are not used. All the training samples are taken randomly and all the results are average of 10 runs. We compared our results with the state-of-the-art methods i.e. support vector machines and semi-supervised graph-based hyperspectral image classification [1] as well as with general LP solvers. All the compared results, i.e SVM and semi supervised, are taken from [1]. We take two scenarios here for evaluating the performance.

A. AVIRIS subimage

In the first scenario, we took a subset part from the whole AVIRIS image i.e columns [27-94] and rows [31-116] having size of 68*86 containing four classes. Table I,II shows the results of classification of proposed approach using homotopy, in comparison with support vector machines,semi-supervised methods and general LP solvers. The results prove the effectiveness of our approach. We can see from the results that in every case, the proposed approach performs better than the support vector machine and semi-supervised method as well as is comparable to LP solvers. In fig 2, we performed the time comparison between homotopy and general LP solvers. It is evident that homotopy based approach is by far the least time consuming.

training samples	SVM	Semi-Supervised	General Sparse	Homotopy	Bagging
15	74.24%	75.75%	84.82%	82.83%	84.93%
20	75.35%	76.93%	85.88%	85.61%	85.93%
25	78.32%	79.85%	86.83%	86.06%	86.68%
30	78.90%	80.68%	87.96%	86.25%	88.02%
100	84.50%	84.83%	90.13%	90.55%	90.73%

TABLE I
THE RESULTS OF SEMI-SUPERVISED LEARNING, SVM AND SPARSE IN SUBIMAGE

B. AVIRIS Full Image

In the second case, we take the whole AVIRIS image i.e 145*145 image which is used with 9 classes. From Table II the results of classification of the proposed technique in comparison with support vector machines,semi-supervised methods and general LP solvers can be observed. In all cases the performance of proposed approach is better than support vector machine,semi-supervised methods and comparable to general LP solvers. In fig 2, we performed the time comparison between homotopy and general LP solvers for full image. It can be seen that homotopy based approach is by far the least time consuming.

C. Discussion

Due to large number of channels in hyperspectral sensors ,i.e high dimensional data, different problems are faced including hughes phenomenon which means that to get good

training samples	Semi-Supervised	SVM	General Sparse	Homotopy	Bagging
15	51.05%	68.45%	72.11%	70.62%	73.24%
20	54.94%	69.30%	74.93%	73.51%	76.76%
25	54.46%	69.61%	75.25%	75.28%	77.12%
30	53.47%	71.22%	75.76%	76.35%	76.57%
100	57.26%	76.21%	81.07%	83.71%	83.48%

TABLE II
THE RESULTS OF SEMI-SUPERVISED LEARNING, SVM AND SPARSE IN FULLIMAGE

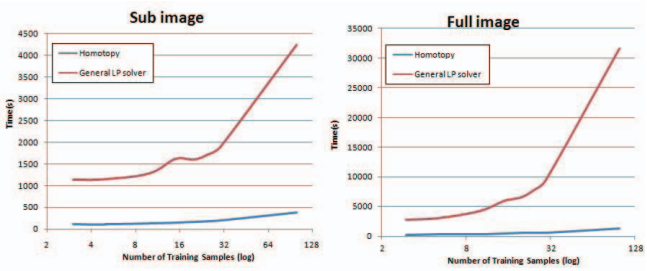

Fig. 2. Time Comparison of proposed approach For AVIRIS full and subimage

classification accuracy we need more train samples which are rarely feasible in remote sensing and especially in case of hyperspectral imaging. Due to these reasons, the hyperspectral data classification in high dimensional space is an uphill task. Dimension reduction is usually adopted to decrease the high dimensions of the data. There are two approaches in dimension reduction; feature selection and feature extraction. To reduce the dimensions of hyperspectral data, many dimension reduction techniques have been proposed but all of these techniques are useful in specific cases and no general technique is available. To find an effective dimension reduction technique for a hyperspectral data requires not only time but some prior information as well. In our approach, this core issue of dimension reduction has also been addressed because our approach does not need reduced dimensions, so it also saves time and effort.

Traditional classification methods, which requires training and testing phases separately and which use the training phase for the model creation that to be used in testing, our sparse approach doesn't need any training and testing phases separately as our approach calculates the sparse representation of test samples using the training samples directly. Also the proposed technique does not require any model selection unlike SVM, which do need optimal parameters using model selection process.

The performance of sparse based classification depends on how well the sparse representation of test samples is calculated using train samples or how much the *Sparse Assumption* is fulfilled. In case of [12], a well selected and organized face database is used as a training set, which perfectly meets *Sparse Assumption*, but in hyperspectral sensing area, the training samples are often limited and not selectable which means *Sparse Assumption* can hardly be hold perfectly. To deal with this problem, as well as the problems mentioned in *Section II C*, we proposed a bagging algorithm to ensemble the sparse classifiers by iteratively voting individual weak classifiers for increasing the classification accuracy. These individual weak classifiers take random inputs from a pool of training samples. The final decision for the classification of test sample is based on the majority votes among the individual classifiers.

IV. CONCLUSION

In this paper, we proposed fast sparse representation based on homotopy for the classification of hyperspectral data. Our approach has the benefits of very fast execution in comparison to general purpose LP solvers. Unlike other classification approaches, it has no requirements of dimension reduction, training models and parameters selection. We also proposed a bagging algorithm to further increase the accuracy. We tested the proposed approach on AVIRIS hyperspectral image. The experiments and comparisons with the state-of-the-art methods shows both the accuracy and the efficiency of proposed approach. It is also evident from the time comparisons that our approach is much faster than general purpose LP solvers.

ACKNOWLEDGMENT

This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 60873266 and 90820304.

REFERENCES

[1] G. Camps-Valls, T. Bandos, and D. Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):2044–3054, 2007.
[2] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
[3] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.
[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.
[5] D. L. Civco. Artificial neural networks for landcover classification and mapping. *International Journal of Geophysical Information Systems*, 7(2):173–186, 1993.
[6] D. Donoho and Y. Tsaig. Technical report 2006-18, department of statistic, 2006.
[7] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
[8] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. Technical Report Stanford CA, 94305, 2006.
[9] D. L. Donoho, Y. Tsaig, I. Drori, and J. luc Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. Technical Report 2006-02, 2006.
[10] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA J. Numerical Analysis*, 20:389–403, 2000.
[11] B. Scholkopf and A. Smola. *Learning With Kernels Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press Series, Cambridge, MA, 2002.
[12] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.
[13] H. Wu, Kuang, Gangyao, and W. Yu. Unsupervised classification method for hyperspectral image combining pca and gaussian mixture model. *Journal of Computer and System Sciences*, 5286:729–734, 2003.
[14] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. (TR09-37), 2009.