Hierarchical classification using discriminative dictionary learning

A project report submitted to Indian Institute of Technology Roorkee
in accordance with the requirements of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

GURWINDER SINGH
HANUMANT MITTAL
SAURABH GOYAL

Under the guidance of

DR. BIPLAB BANERJEE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247667 (INDIA)

Candidate's Declaration

We declare that the work presented in this project report with title "**Hierarchical classification using discriminative dictionary learning**" towards the achievement of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the **Dept. of Computer Science & Engineering**, **Indian Institute of Technology, Roorkee, India** is an authentic record of our own work done during the period **from August 2017 to April 2018** under the supervision of **Dr. Biplab Banerjee**, Assistant Professor, Dept. of CSE, IIT Roorkee.

The content of this report has not been submitted by us for the award of any other degree of this or any other institute.

DATE : .... SIGNED: ....

PLACE: ....

(Gurwinder Singh)

(14114026)

DATE : .... SIGNED: ....

PLACE: ....

(Hanumant Mittal)

(14114027)

DATE : .... SIGNED: ....

PLACE: ....

(Saurabh Goyal)

(14114051)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

DATE : ....

SIGNED: ....

(DR. BIPLAB BANERJEE)
ASSISTANT PROFESSOR
DEPT. OF CSE, IIT ROORKEE

ACKNOWLEDGEMENTS

First and foremost, we would like to express our sincere gratitude towards our guide **Dr. Biplab Banerjee**, Assistant Professor, Computer Science and Engineering, IIT Roorkee for his ideal guidance throughout the period. His advices, insightful discussions and constructive criticisms certainly enhanced our knowledge and improved our skills. His constant encouragement, support and motivation have always been key sources of strength for us to overcome all the difficult and struggling phases.

We would also like to thank **Department of Computer Science and Engineering, IIT Roorkee** for providing lab and other resources for this project, as well as for being a perfect catalyst to our growth and education over these past four years.

We also extend our gratitude to all our friends, for keeping us motivated and providing us with valuable insights through various interesting discussions.

Abstract

A label consistent K-SVD algorithm applied hierarchicaly to learn dictionary space discriminative in nature at each level for sparse coding is presented here. A binary tree is constructed with the all the classes combined representing the root node and leaf nodes representing the individual classes. Class Label is used alongwith associated label data with each item in dictionary, columns of our dictionary matrices, in order to enforce discriminatability in sparse code learnt during the entire process of dictionary learning. To elaborate, we use a new label-consistent constraint- “discriminative sparse-code error“ which is then combined with the classification error and reconstruction error in order to form a combined objective function. To obtain the optimal solution in an efficient manner we use the K-SVD algorithm. A single over-complete dictionary an optimally linear classifier is learnt jointly for each node of the class tree. Once the dictionaries and classifiers at each node is learnt we move to the classification of test data. An image is passed at the root node and is classified at every level till the leaf node is reached.

TABLE OF CONTENTS

List of Figures

Figure **Page**

List of Tables

Table **Page**

CHAPTER 1

INTRODUCTION

Sparse coding has successfully been applied to a various existing problems in image analysis and computer vision, including restoration of images [1],[2],image denoising [3] and image classification [4],[5],[6]. y, an input signal is approximated by the use of a linear sparse synthesis of items from a Dictionary,D which is over complete. The quality of D highly determines it's performance.The complete set of the training samples is employed as dictionary for sparse coding by [5]. Using this, remarkable performances have been achieved on face recognition. There are various many algorithms that claim to conveniently learning an over-complete dictionary enforcing some paradigm discriminative in nature [7], [8], [9].

Small-sized dictionary was developed [5], [10], [11], [12],[13], [14] in order to increase the scale to appropiate training sets large in size. [5] is an example where manually selected training sample data has been used to construct dictionary. A separate dictionary is learnt by [10] for each and every class, then classification is performed over the reconstruction error. However, it is very time extensive to construct dictionary during training and to run sparse code during testing, especially when the number of classes are huge. K-SVD, a dictionary learning algorithm introduced in [11] introduced a proficient way to learn an over-complete dictionary from a particular set of training signals. Filling the missing pixels Image compression are among the various applications. K-SVD does not consider the discrimination potential of the dictionary but rather focuses on the power of representation of the dictionary learned(optimal sparse representation for the training signals). The same effective sparse coding is shared by the method of optimal directions (MOD) [14] as K-SVD. A discriminitative dictionary is obtained by updation of items iteratively in the dictionary based on the output of a predictive linear classier in [12].The learning of dictionary and classifier processes are unified in Discriminative K-SVD algorithm proposed in [13].

We construct a binary class tree and then execute a supervised algorithm over the class tree to learn a compact and discriminitative sparse coding dictionary. The class tree is built in a way such that classes close to each other are grouped into similar child nodes. The root node represents all the classes combined and the lead node represents the individual classes.To represent any cluster of classes the mean vector of the classes are used. The dictionary space as well as the classifier is learned for each node of the class tree. It provides for a dictionary to be more accurate on the node level since it needs to classify the non adjacent ones. A "discriminative" error criterion for sparse coding is explicitly incorporated. An "optimal" performance criterion for classification is also incorporated into the objective- function and then optimized further utilising K-SVD algorithm. A "discriminative" sparse representations of signals is provided by the dictionary learned. A good accuracy is achieved on classification of object with just a very linear multi class classifier, in the contrast of alternative currently existing approaches of sparse coding [15], [16], [10], [4] which learn a single classifier for every pair of the categories since a both reconstructive and discriminative dictionary is learned, in contrariety to the constructive traditional ones [14], [11], [5], [1]. Due to the use of K-SVD, this approach is efficient and bounded by it. The discriminative dictionary is learned simultaneously with the linear classifier for each node of the class tree which is in clear contrast to approaches of dictionary learning like [15], [16] which solve subproblems iteratively so as to approximate a joint solution, or the approaches like [17], [18], [10], [19] which separately learn the dictionary and classifier.

1.1 Organization of the Report

Chapter 2 describes the problem statement

Chapters 4-7 describe our work done, and how we solved the three sub-problems mentioned in chapter 2.

Chapter 4 describes the dataset and the classification task.

Chapter 5,6,7 gives the details of the whole process.

Chapter 8 gives the results and discussions.

CHAPTER 2

PROBLEM DEFINITION

The main objective of the project is to use **hierarchical classification** along with **group sparsity** to improve accuracy of the image classification system. To create more discriminatative dictionaries, we attach the labels to dictionary elements along with input data during the estimation process, so that more efficient sparsecodes are obtained. This dictionary learning process is carried at each node in our tree alongwith the classifier learning process.

Our system performs these different tasks. It divides the dataset into clusters of classes which represents nodes of a tree. Every node gets its own classifier and dictionary so that it can be decided that which datapoint belongs to which one of its children. The leaf nodes in this system are the actual classes that are provided in the dataset. In this way, we have represented the whole system as a hierarchy of classes and clusters of classes in form of a binary tree. The root node represents the cluster of all classes. The leaf nodes represents individual classes.

2.1 A new application of sparse coding

Sparsecoding has been the goto approach for researchers in recent times and has been very useful in classification applications.Technically, Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data efficiently using dictionary. This helps in imposing many constraints on the system using objective functions. In our system, we use separate dictionaries and classifiers at each node in our tree using the sparse code optimizations.

2.1.1 Task 1 :Constructing the Hierarchical structure

We divide the dataset into clusters of classes , called nodes, which gives a tree like structure.The root node has all the classes while the leaf nodes have the actual classes.

2.1.2 Task 2 :Learning reconstructive dictionary and classifier

After we have obtained the hierarchical structure, we will use that to train level wise classifiers in such a way that at every node, we get which of the children nodes are next in path. Ultimately at leaf nodes it gives us one of the original classes. We will use KSVD to find the dictionary and classifier at each node using OMP for optimization. The complete procedure from intitialization of LC-KSVD, using OMP for KSVD1 and KSVD2 is computed for each node. This results in the D(Dictionary) and W (classifier) for every node. This completes the training process and the D,W are stored for each of these nodes in the earlier dictionary with key as the cluster of classes.

2.1.3 Task 3: Hierarchical Classifier

Every test image is passed to the root node classifier which passes the image to either of the child clusters. This process is carried at each node until the leaf node is reached. The leaf class represents the final prediction of this image. This process is carried for each image in the test set.

To summarize, the **hierachical sparse based learning** will aim at the following three major tasks:

1. Constructing the Hierarchical structure representing cluster of classes at every node with independent classes at root nodes
2. Computing the sparse over-complete dictionary as well as the binary classifier at each of these nodes using Label Consistent-KSVD.
3. Passing the test images from root to leaf of this tree with binary classification at every node and the last node representing the final prediction for this image.

The three tasks hold importance in the context of uses of sparse code for this different task. Sparse code has been used for signal recogniton but we used it for classification alongwith a hierarchical structure to further improve the accuracy. The dictionary at each node further improves the accuracy as the distinguishive features divide the cluster and combining features join the child classes to the parent. The use of Label Consistent- Singular Value Decomposition and Orthogonal Matching pursuit for optimised parameters and dictionary alongwith their performance is further dicussed in this report.

Chapter 3

Literature Review

Sparse coding techniques utilising supervised dictionary learning has been a hot topic for researchers in recent years. Some of these learn a class specific or type specific dictionaries as illustrated in [20], [10], [21],. There's a boosting procedure which is used in [20], [21], it provides extra ability for learning multiple dictionaries. Some [10] also have adopted the approach of learning one dictionary per category(class) , which does not involve sparse codes but based on the corresponding reconstruction errors.

Many popular works like [15], [16], [13], [10], [19], [12], [18] and [17] describes algorithms which attempts to include the distunguished terms in the objective functions.

There are several types of discriminative criterias which have been presented in various works like Fisher discrimination criterion [17], hinge loss, linear classification error [13], softmax discriminative cost function [19], [10], [16], [18].

3.1 Related Works

In the earlier works e.g. [19], [10], [20],[17], [22], it has been a popular thought to take dictionary learning and classifier training as a entirely isolated procedures. These works has the following flow ,start with learning a dictionary and then train a classifier using that dictionary (using different kinds of methods). The features used are typically sparse representations and then classification is done, e.g. using SVM, Random Forest etc. Some more advanced methods, [12], [15], [16], [13], use a unification of the processes by a combined discriminative and reconstructive formula of these two procedures.

The method we use for classifier and dictionary learning at each node, also falls under that category of approaches.In Some approaches [15], [16], several classification models and an over-complete dictionary are learned in parallel, but this usually does not scale good for a relatively huge number of the classes. In [12], in accordance with the output of a linear classifier, the dictionary is updated iteratively, though this approach suffers from the problem of local minimum because dictionary estimation and classifier learning are alternated because of the design of the process.

Some of these approaches [13], incorporates classification error within the main objective function, but in case of smaller dictionaries, discriminability is not ensured.

3.2 How our algorithm stands out

Compared to all these approaches, the approach used by us more effectively learnss a distinguishive dictionary at every node of each level thereby reflecting the accuracy. We applied Label Consistent KSVD with OMP to find the coefficients to effectively compute the dictionary and classifier at each node. The dictionary we learned has an overall good,more distinctive representation of the data for all classes.

CHAPTER 4

DATA COLLECTION

The Botswana hyperspectral dataset was used to validate the proposed framework.

4.1 The Botswana Hyperspectral Data :

The Botswana Hyperspectral dataset consists of a segment of heyperspectral-images obtained using 'EO-1' satellite's Hyperion sensor over an area located at the Okavango Delta, Botswana (with pixel capability 1476×256, on May 2001 (see Fig. 4.1). An image with spatial resolution at 30m and covering land-strip with size $7.7 \times 44.3 km^2$ with 145 from the 242 original spectral bands.10 from the 145 are chosen on the basis of their discriminative ability using an approach described in [23]. A total of 14 classes are identified for two separate spatially disjointed areas.

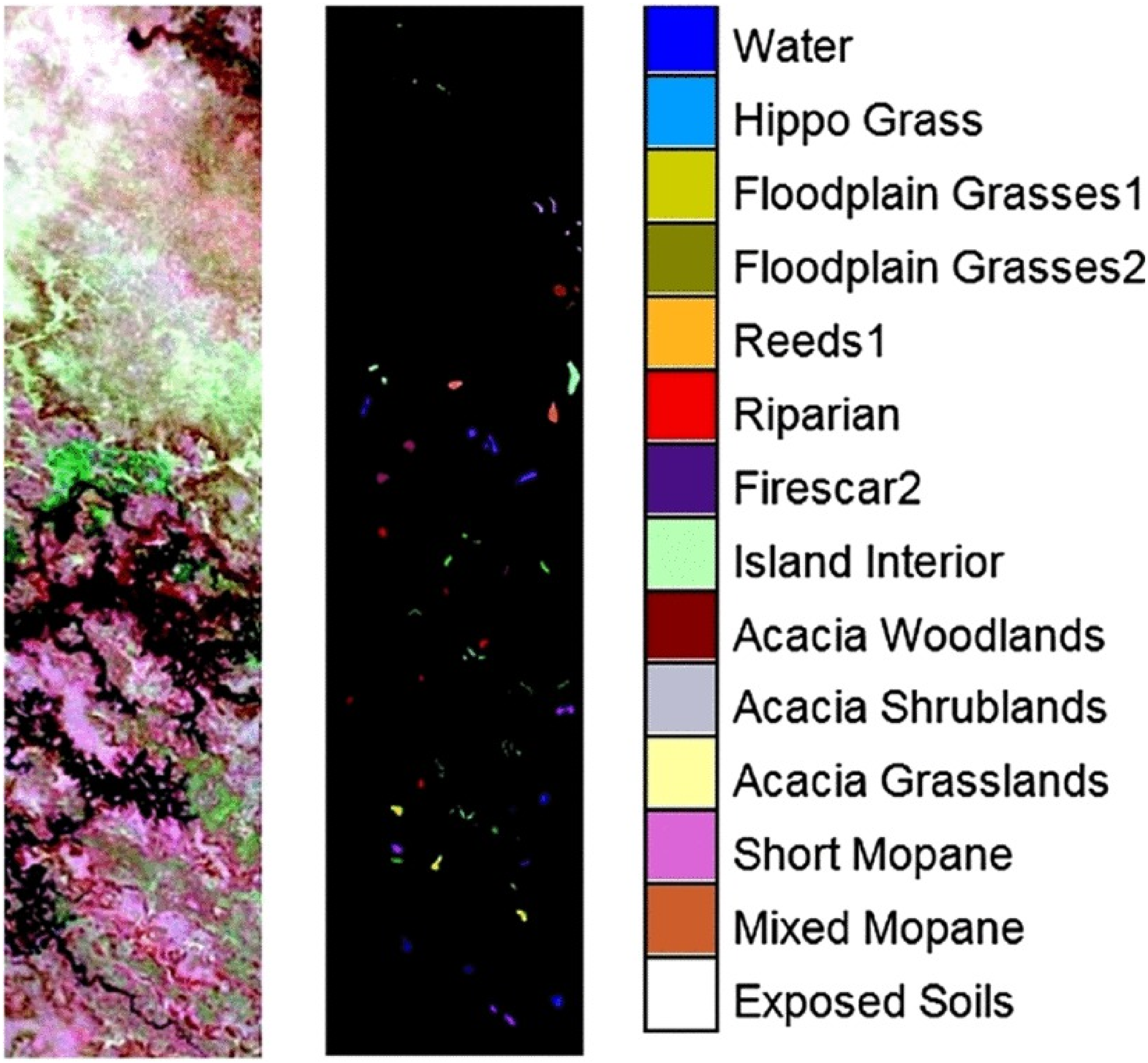

Figure 4.1: Selected train and test sites in Botswana Dataset.

CHAPTER 5

Construction of the Hierarchical Tree

Constructing the tree facilitating the classification in a hierarchical manner

5.1 Generating the mean vector of each class

- The mean vector of each class is generated which would be distinguishing feature of each class when part of any cluster.

5.2 Making the initial nodes of the tree

The mean vector of the classes are passed as data points at root node. K-means is applied at this node to classify these N data points into two cluster of classes. Each cluster represents a combination of the classes.

5.2.1 Recursively building the tree

The above procedure of using k-means to divide the classes into two classes is carried until the leaf node is represented by one single class. These N final nodes would represent the classification of all the test images.

5.2.2 Structure for storing a Node

To store any node we built a dictionary with key as the tuple representing cluster of the classes present at that node. The value is multiple with first element as the tuple of the left child of that node, and the second being the right child of that node. This data is further processed and the classifier and dictionary space at the node is further stored.

CHAPTER 6

Dictionary and Classifier learning

6.1 Level wise training of Classifier

After we have obtained the hierarchical structure, we use that to train level wise classifiers in such a way that at every node, we get which of the children nodes are next in path. Ultimately at leaf nodes it gives us one of the original classes.

6.2 Steps in learning

The following sections introduce the concepts used for the training of the classifier.

6.2.1 Orthogonal Matching Pursuit

OMP is used for the recovery of sparse signals of high(or low) dimensions using a tiny no/ of noisy linear measurements. OMP, an iterative greedy algorithm which chooses at every step the column, which is max correlated to the current residuals.

This is one of the simplest ways to recover. It is simple and greedy (with some chance to recover). Its like a discrete L1 version of the technique for SVD, and can be useful for many other hard optimization problems. We assume we know the measurement d*N matrix X, and the N measurements y.

First, we find the measurement column X_j (not the row x_i used to measure).

$$X_j = \arg\max_{X_{j'} \in X} |\langle y, X_{j'} \rangle|. \quad .. (1)$$

This represents the single index of S that explains the most about y. Next,we find the weight.

$$\gamma = \arg\min_{\gamma \in \mathbb{R}} \|y - X_j \gamma\| \quad .. (2)$$

that represents our guess of entry s_j in S. If S is always 0 or 1, then we may enforce that γ = 1.

Finally, we calculate the residual r = y - $X_j\gamma$. This is what remains to be explained by other elements of S.

Then we repeat for t rounds. We stop when the residual is small enough (nothing left to explain) or γ is small enough (the additional explanation is not that useful).

Algorithm 17.3.1 Orthogonal Matching Pursuit

Set $r = y$.
for $i = 1$ **to** t **do**
  Set $X_j = \arg\max_{X_{j'} \in X} |\langle r, X_{j'} \rangle|$.
  Set $\gamma_j = \arg\min_{\gamma} \|r - X_j \gamma\|$.
  Set $r = r - X_j \gamma_j$.
Return $\hat{S}$ where $\hat{s}_j = \gamma_j$ (or 0).

6.2.2 LC-KSVD

LC-KSVD is the label consistent variant of the K-SVD algorithm that is used to learn and iteratively reconstruct a discriminative dictionary which is further used for getting sparse code representation of given data.

K-SVD uses singular value decomposition technique for learning dictionary used for sparse representations. The standard k-means clustering is generalized to obtain the K-SVD algorithm. It works by interchangeably processing sparse coding part and dictionary learning part, updates the items in the dictionary to an improved fit according to the data. As of LC-KSVD, a discriminatative and reconstructive dictionary is learned using the class label info from the input signals. All dictionary items are selected in such a way that it can be associated with a particular class label because they are picked from that specific class only. After that some hard regularization terms are added in the equation, which denotes consistent label -regularization terms, and the joint-classification error.

6.3 Algorithm

For every node, the dictionary is learned in parallel to the classifier. For the first node, initiliazation for the Label Consistent- KSVD is done.

6.3.1 Initialization for KSVD

: The variables D_0,W_0 and A_0 are needed to be initialized for LC-KSVD. For D_0, K-SVD is run many times for each class and all the outputs are combined (basically, dictionary atoms are learned from every class invidually) for all K-SVDs. Every dictionary item (represented by d_k is given starting value on the basis of the label of the class it belongs to and this is kept constant throughout the process of estimation of the dictionary, however while learning, d_k is changed. Dictionary items are equivalently distributed within the classes such that number of items is in accordance with the dictionary size. To initialize A_0,as in [24], the ridge regression model is employed , which is as below:

A = $\underset{A}{\arg\min}$ + | | Q - AX | | ^{2} + λ_2 $||A||_2^2$ .. (3)

Note that the loss used is quadratic and l2 norm is used.

the following solution for A is obtained:

A = $(XX^t$ + λ_2I$)^{-1}$XQ^t .. (4)

In an anlogous way for W_0, using the ridge-regressions , this is obtained:

W = (XX^t λ_1I$)^{-1}$XH^t ..(5)

After we have obtained initial $D-0$, the K-SVD process is applied for computing the sparsecode X of the input training signals Y. After that, A_0 is initialized using X in the equation 4 and W_0 using equation 5.

For learning the dictionary , Orthogonal Matching Pursuit is used with α and β being the variables in the algorithm. A couple of distinct methods for LC-KSVD are employed.

6.3.2 LC-KSVD1

: The discriminatability of the obtained sparse codes of the input signals plays a major role in determing the effectiveness of the linear classification. Given a learned dictionary D, discriminatative sparsecodes X are obtained according to the following objective function:

$$<D, A, X> = \underset{D,A,X}{\arg\min} ||Y - DX||_2^2 + \alpha ||Q - AX||_2^2 \ s.t. \ \forall i \ , \ ||x_i||_0 \leq T \qquad ..(6)$$

In this equation, the variable α regulates the relative effect between label consistency maintainance and reconstructions, , and discriminatative sparsecodes of the data given for classifying is represented by Q = [q1...qN] $\epsilon \ R^{K\times N}$. Definition of saying that $q_i = [q_1^i ... q_K^i]^t = [0...1, 1, ...0]^t \ \epsilon \ R^K$ is 'discriminitative' sparsecode of data y_i , if at the points in dictionary where the label is common as that of data, non-zero value of q_i are obtained.For example, assuming D = $[d_1...d_6]$ and Y = $[y_1...y_6]$, where y_1, y_2, d_1 and d_2 are belongs to first class, y_3, y_4, d_3 and d_4 belongs to second class, and y_5, y_6, d_5 and d_6 belongs to third class, Q is given as:

$$Q \equiv \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

A is a 'linear-transformation matrix'.The sparsecodes are rendered more discriminatative in space R^K by a lineartransformation , as $g(x; A) = Ax$.

The discriminatative sparsecode error is represented as $||Q - AX||_2^2$, this forces X to estimate the discriminitatice sparsecodes Q. This enforces the points from same labels to have least distant sparsecodes conversion(which is basic idea of LC-KSVD), this improves the performance of the linear-classifiers used.

6.3.3 LC-KSVD2

: Now, we want to incorporate the classifier error expression in the objective-function used for the estimation of dictionary, to improve its effectiveness in using for a classifier. A simple linear-predictive classifier f(x; W) = W x is used. Dictionary is learned using an objective function having 'reconstructive' and 'discriminatative' features, it is as below:

$$<D,W,A,X> = \underset{D,W,A,X}{\arg\min} ||Y - DX||_2^2 + \alpha ||Q - AX||_2^2 + \beta ||H - WX||_2^2 \; s.t. \; \forall i, ||x_i||_0 \leq T \quad ..(7)$$

The classifier error term is $||H - WX||_2^2$. The classification variables are represented by W. The training labels of the data Y are represented as H = $[h_1...h_N] \in R^{m\times N}$.

The label of data y_i corresponding to non-zero position in rowvector $h_i = [0,0...1...0,0]^t \epsilon R^m$ is marked as the class of it. α, β are parameters that regulates the relative values of their respective terms.

Assume discriminatative sparsecodes X'=AX and $A\epsilon R^{K\times K}$ happens to be invertible matrix, then $D' = DA^{-1}, W' = WT^{-1}$. Then we can rewrite the expression in the eq. 7 as:

$$<D', W', X'> = \underset{D',W',X'}{\arg\min} ||Y - D'X'||_2^2 + \alpha ||Q - X'||_2^2 + \beta ||H - W'X'||_2^2 \; s.t. \; \forall i, ||x_i||_0 \leq T \quad ..(8)$$

The reconstructive-error is shown by the 1st term, discriminatative sparsecodes-error is shown by the 2nd term, and classifier-error by the 3rd term. The 2nd term $||Q - X'||_2^2$ enforces discriminatativeness between the sparsecodes of the classes while the 3rd term $||H - W'X'||$ imporoves the performance of the classification. If we learn dictionaries by the above method, they will be well-adjusted according to the inner-patterns and structures within the data and provides a more adequate translation for all data points with tight-sparsity enforced.Also,more discriminatative sparsecodes are generated regardless of the size of the dictionaries. The sparsecodes thus obtained can be used for calssification purposes,[5] yielding good results. The performance of linear-classification is significantly enchanced by the discriminatativeness of sparsecodes.

The complete procedure from intitialization of LC-KSVD, using OMP for KSVD1 and

KSVD2 is computed for each node. This results in the D(Dictionary) and W (classifier) for every node. This completes the training process and the D,W are stored for each of these nodes in the earlier dictionary with key as the cluster of classes.

Chapter 7

Classification

The classifier is trained independently for each node of the class tree.

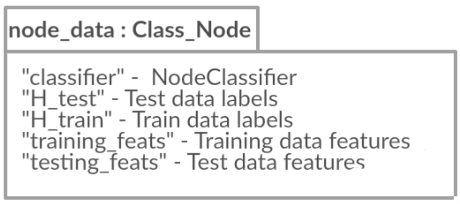

Figure 7.1: Attributes of each Class Node.

The test image Data Points are passed to the root node classifier which classifies them to either of the child nodes. This process is carried at each node until the leaf node is reached. The leaf class represents the final prediction of this image. This process is carried for each image in the test set. One such iteration is shown for a test datapoint belonging to the class Hippo Grass in Figure 7.2

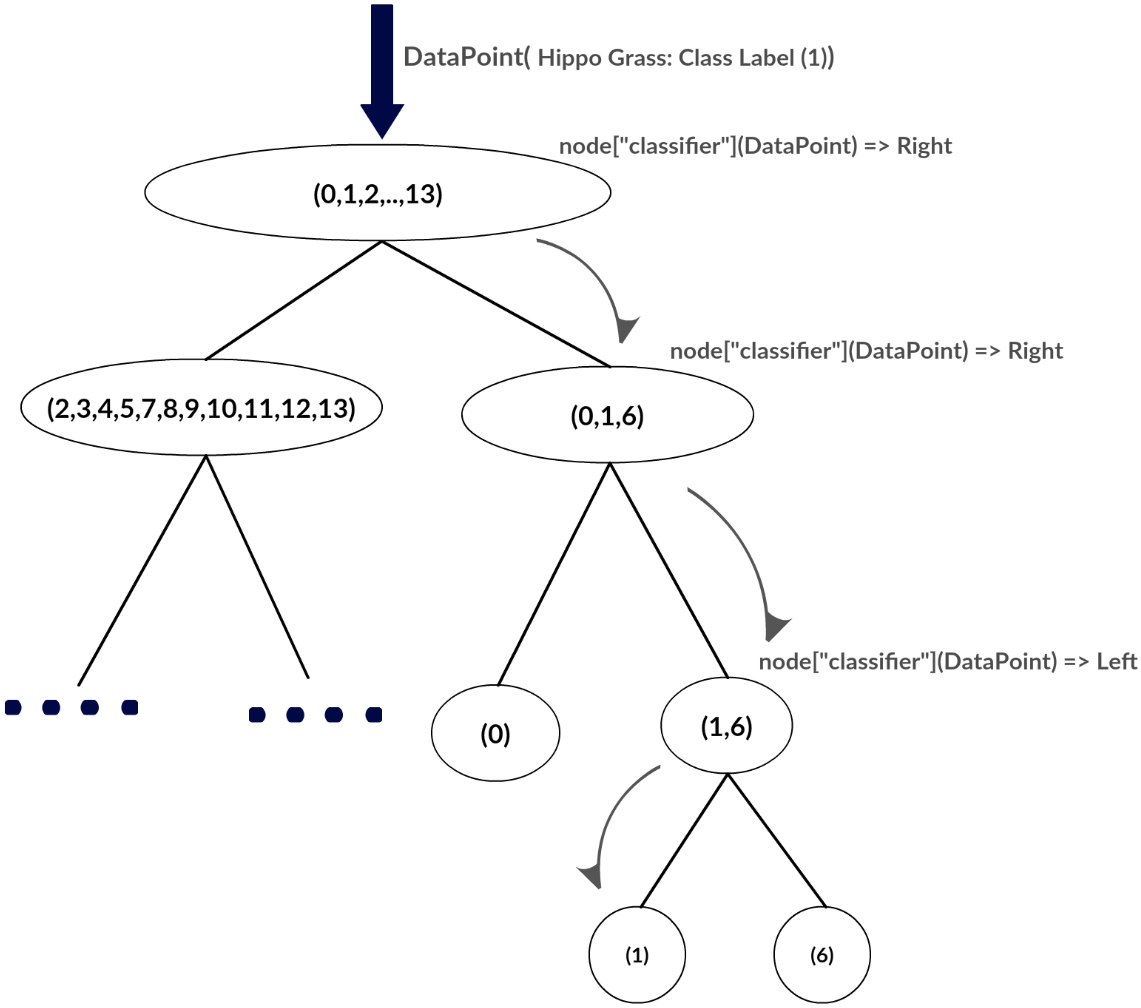

Figure 7.2: Datapoint is iteratively classified on each node to finally reach Leaf Node which represents the final prediction.

CHAPTER 8

Results and Discussion

In the previous chapters we have discussed how the the algorithm works and a description of the analyses it can provide good classification tree for real world data. In this chapter, we discuss the results obtained by our algorithm for hyperspectral multiclass data.

8.1 Performance on the Botswana Dataset

This is a demanding dataset consisting of multiple classes of hyperspectral data overlapping with each other. The class wise performance analysis for test datapoints in the Botswana dataset has been be presented in Table 8.1. The figure 8.1 represents the binary class tree constructed using the hyperspectral Botswana data. We can take in consideration the tree-based arrangement of all the classes in the binary tree wrt their spectral signatures, implying that the suggested hierarchical arrangement of the class tree sharply protects the physical attributes of the classes i.e nearly overlapped classes probably stay adjacent to each other upto the root of tree).

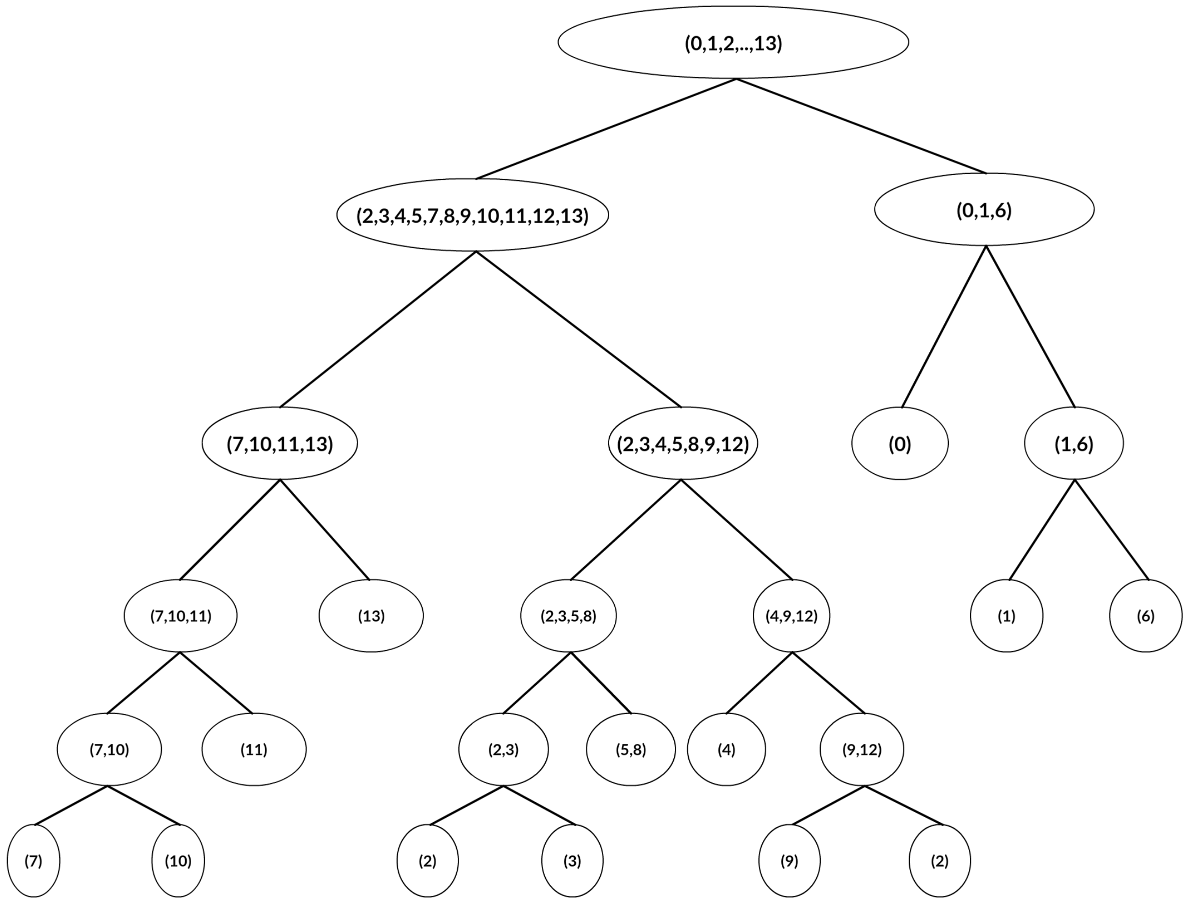

Figure 8.1: Binary Class Representation of Botswana Data.

Highly discriminative classes like Water(0) and Island Interior(7) are classified very accurately by the algoritm. It can further be observed from Table 8.1 that the algorithm used partly fail discriminating the finely-grained classes. To take an example, for some pairs of overlapping classe (Reeds1(4) and Hippo Grass(1)), one of the classes is well detected in the cost of the other. We obtain 31.8% classification accuracy for Reeds1(4) while the classification performance of 98.8% is reported for the overlapping class Hippo Grass(1). Similarly, the algorithm is unable to appropiately classify Mixed Mopane (performance of classification $\approx$ 53.1%) whereas its performance increases upto $\approx$79% for the Short Mopane class.

Table 8.1: Class Accuracy Table

Land Cover Classes	**Accuracy(Proposed Method)**
Water (0)	100.0
Hippo Grass(1)	98.765
Floodplain grasses1(2)	77.333
Floodplain grasses2(3)	78.021
Reeds1(4)	31.818
Riparian (5)	85.321
Firescar2 (6)	81.927
Island Interior (7)	100.0
Acacia Woodlands (8)	82.089
Acacia Shrublands (9)	56.179
Acacia Grasslands (10)	93.678
Short Mopane (11)	78.823
Mixed Mopane (12)	53.125
Exposed Soil (13)	89.583
Overall Performance	78.115

Table 8.2: Accuracy with different values of training parameters.

(SparsityThreshold, $\sqrt{\alpha}$, $\sqrt{\beta}$)	Accuracy(%)
('18', '2', '3')	75.95
('18', '8', '9')	64.13
('26', '2', '3')	63.58
('18', '8', '3')	78.11

8.2 Fine Tuning the training parameters

The hyper-parameters of the training in this case are α, β and SparsiyThreshold, (with respect to the equations described in chapter 6). During the implementation of the system, we tried to determine the optimal values of the hyper-parameters used in the training with respect to the dataset in hand. a few examples are shown in Table 8.2

8.3 Conclusion

We are able to obtain an overall good classification accuracy using our new design of classifier. This shows that the approach to divide classes into clusters of classes and learning individual dictionaries and classifiers for each node is able to produce significant results.

References

[1] B. F. H. M. P. J. S. G. Z. A. Mairal J., Leordeanu M., "Sparse representation for color image restoration," *IEEE Trans. Img. Proc.,*, 2008.

[2] C. S. L. Y. Ranzato M., Poultney C., "Efficient learning of sparse representations with an energy-based model," *NIPS*, 2006.

[3] E. M. Aharon M, "Image denosing via sparse and redundant representations over learned dictionaries," *IEEE Trans. Img. Proc.,*, vol. 54(12), pp. 3736–3745, 2006.

[4] g. Y. H. T. Yang J., Yu K., "Linear spatial pyramid matching using sparse coding for image classification," *CVPR*, 2009.

[5] G. A. S. S. M. Y. Wright J., Yang M., "Robust face recognition via sparse representation," *TPAMI*, vol. 31(2), pp. 210–227, 2009.

[6] B. J. Bradley D., "Differential sparse coding," *NIPS*, 2008.

[7] R. R. N. A. Lee H., Battle A., "Efficient sparse coding algorithms," *NIPS*, 2006.

[8] P. J. S. G. Mairal J., Bach F., "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research 11*, pp. 19–60, 2010.

[9] Y. K. L. F. h. T. G. Y. Wang J., Yang J., "Locality-constrained linear coding for image classification," *CVPR*, 2010.

[10] P. J. S. G. Z. A. Mairal J., Bach F., "Discriminative learned dictionaries for local image analysis," *CVPR*, 2008.

[11] B. A. Aharon M, Elad M, "K-svd: An algorithm for designing overcomplete dictionries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54(1), pp. 4311–4322, 2006.

[12] V. S. Pham D., "Joint learning and dictionary construction for pattern recognition," *CVPR*, 2008.

[13] L. B. Zhang Q., "Discriminative k-svd for dictionary learning in face recognition," *CVPR*, 2010.

[14] H. J. Engan K., Aase S., "Frame based signal compression using method of optimal directions (mod), 1999," *IEEE Intern. Symp. Circ. Syst*, 1999.

[15] H. T. Yang J., Yu K., "Supervised translation-invariant sparse coding," *CVPR*, 2010.

[16] P. J. S. G. Z. A. Mairal J., Bach F., "Supervised dictionary learning," *NIPS*, 2009.

[17] A. S. Huang K., "Sparse representation for signal classification," *NIPS*, 2007.

[18] L. Y. Boureau Y., Bach F., "Learning mid-level features for recognition," *CVPR*, 2010.

[19] B. F. H. M. P. J. S. G. Z. A. Mairal J., Leordeanu M., "Discriminative sparse image models for class-specific edage detection and image interpretation," *ECCV*, 2008.

[20] F. X. D. T. Zhang W., Surve A., "Learning non-redundant codebooks for classifying complex objects," *ICML*, 2009.

[21] S. R. J. F. Yang L., Jin R., "Unifying discriminative visual codebook genearation with classifier training for object category recognition," *CVPR*, 2008.

[22] S. G. Rodrihuez F., "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaryies," *IMA Preprint 2213*, 2007.

[23] C. P. L. Bruzzone, "A novel approach to the selection of spatially invariant features for the classification of hyperspectral images with improved generalization capability," *IEEE Trans. Geosci. Remote Sens.*, vol. 47,no. 9, pp. 3180–3191, Sep. 2009.

[24] O. D. Golub G., Hansen P., "Tikhonov regularization and total least squares.," *SIM J. Matrix Anal. Appl.*, vol. 21(1), pp. 185–194, 1999.