

Scalable Image Classification Using Compression

Akshata K. Naik¹, Dr. Dinesh Acharya U²

¹(Computer Science and Engineering, Manipal Institute of Technology, India)

²(Computer Science and Engineering, Manipal Institute of Technology, India)

Abstract: The increasing rate of data growth has led to finding techniques for faster processing of data. Big Data analytics has recently emerged as a promising field for examining huge volume of datasets containing different data types. It is a known fact that image processing and retrieval involves high computation especially with a large dataset. We present a scalable method for face recognition based on sparse coding and dictionary learning. Sparse representation has closer resemblance with a cortex like image representation and thus more closer to human perception. The proposed method parallelizes the computation of image similarity for faster recognition.

Keywords: Dictionary learning, Image classification, Parallel computation, Sparse representation.

I. Introduction

Data mining is a field of computer science, dealing with finding of hidden patterns from data. Recently, a line of research has developed towards parameter free data mining [1] [2]. The field exploits the Kolmogorov complexity which is a theoretical base for measuring randomness of data. The complexity however is not a computable quantity. It is thus closely measured by the size of data. The interesting aspect of this field is that, the methods employed are generic and application independent. One of the most widely used distance metric is the Normalized Compression Distance (NCD). It has been proved that the distance metric is successful in clustering and classification of one-dimensional data [1] [2]. The successful implementation of NCD in the field of multi-dimensional data like images is limited, as the compressors used for images are not a normal compressor [3].

Sparse representation is one of the effective methods for representation of signals and image processing [4] [5]. It has been widely used for various applications in image processing viz. image denoising, image compression and image classification [6] [7] [8]. Sparse coding is a class of unsupervised learning to find a set of over-complete basis $\mathbf{D} \in \mathbb{R}^{m \times n}$ referred to as the dictionary. A dictionary of dimension $m \times n$ is called an over-complete dictionary if $m < n$. The aim is to learn \mathbf{D} such that an input vector \mathbf{b} can be represented as a linear combination of these basis vectors. An over-complete dictionary produces a system of linear equations with infinite number of solutions [8]. The solution with few number of non-zero coefficients is more desirable. Such a solution is formulated as :

$$\text{Min } \sum || \mathbf{a}_x ||_0 \quad \text{s.t. } \forall i, || \mathbf{b}_x - \mathbf{D} \mathbf{a}_x ||_2 \leq \epsilon \quad (1)$$

The vector \mathbf{a}_x is the sparse representation of input vector \mathbf{b}_x and ϵ is the permissible error. An over-complete dictionary can either be pre-specified or learnt from training data. Learning a dictionary has an added advantage of adapting itself to fit a given set of signals. K-SVD algorithm, which is a generalization of the K-means clustering, is a popular dictionary learning algorithm.

Image similarity based on sparse reconstruction error uses the above mentioned dictionary learning technique [9]. A similarity measure viz. sparse Signal-to-noise ratio (SSNR) has been proposed by the authors. The measure involves the sparse coding which is computationally intensive. This paper proposes a parallel computation method for face recognition using SSNR. The proposed method distributes the task of measuring similarity between images, among multiple cores of the same as well as different computers in a cluster. The method has shown an almost linear speedup when there is a balance between the size of dataset and the number of nodes.

1.1. Dictionary Learning

Dictionary learning is based on a given set of examples. Given such a set $\mathbf{B} = \{ \mathbf{b}_{xi} \}$ such that, $1 \leq i \leq N$, a dictionary is learnt by solving the equation (1) for each of the example \mathbf{b}_{xi} . K-SVD is one of the successfully used dictionary learning algorithm. The algorithm iterates between a sparse coding stage and dictionary update stage. Dictionary for an image is learnt by extracting patches of dimension $\sqrt{m} \times \sqrt{m}$. The patches are then converted to vector $\mathbf{b}_{xi} \in \mathbb{R}^m$. Each of these patches act as a column vector for the matrix \mathbf{B} .

We then learn an over-complete dictionary $\mathbf{D} \in R^{m \times n}$ that has n atoms (basis vector) using the patches from B as input. The dictionary \mathbf{D} , should be able to closely approximate each patch as a linear combination of a small number of atoms [8].

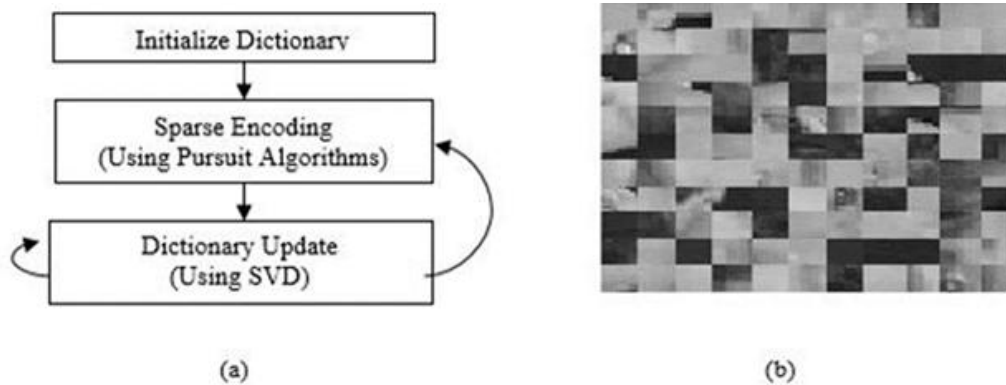


Fig. 1 (a) Block diagram of K-SVD algorithm; (b) Initialized patches for dictionary learning

K-SVD algorithm finds an approximation to equation (1) using two steps. The 2 steps are 1) sparse coding and 2) dictionary update. The first step employs any pursuit algorithm to compute a sparse vector \mathbf{a}_x , when the dictionary \mathbf{D} is fixed. We use a greedy algorithm called Orthogonal Matching Pursuit (OMP) due to its simplicity and faster implementation. The second step updates the dictionary columns sequentially along with the corresponding coefficients in \mathbf{a}_x . The detail implementation of K-SVD algorithm can be referred from the paper [10]. The initialized patches for dictionary learning and the block diagram for K-SVD is shown in Fig. 1(a) and 1(b) respectively.

1.2. Similarity Measure

The similarity measure SSNR [9] computes the similarity between two images X and Y. Computation of similarity between X and Y requires learning of dictionaries \mathbf{D}_x and \mathbf{D}_y for the images respectively. Each of the images are then reconstructed using both the dictionaries. The SSNR uses the Signal-to-noise ratio (SNR) between the original image and reconstructed image. SSNR function is given as:

$$SSNR(X, Y) = \frac{SNR(U_x, V_x) + SNR(U_y, V_y)}{SNR(U_x, W_x) + SNR(U_y, W_y)} \quad (2)$$

In equation (2) U_x and U_y are set of random patches belonging to images X and Y respectively. The term V_x and V_y are closest approximation of U_x and U_y using the dictionaries \mathbf{D}_y and \mathbf{D}_x respectively. Similarly W_x and W_y are approximation of U_x and U_y using the dictionaries \mathbf{D}_x and \mathbf{D}_y , respectively. SNR is the Signal-to-noise ratio. Higher value of numerator indicates that the images are similar. The denominator is used for normalization. The SSNR has the following properties [9]:

- a) *Non-negativity*: value of $SSNR(X, Y)$ lies between 0 and 1. $SSNR = 1$ when $X = Y$.
- b) *Symmetry*: $SSNR(X, Y) = SSNR(Y, X)$

It should be noted that the reconstruction of patches has sparsity (number of non-zero coefficients) constraint on the sparse vector. The sparsity is set to a value β . It is formulated as below:

$$\text{Min} \sum ||\mathbf{v}_{xi} - \mathbf{D}_y \mathbf{a}_x ||_2 \text{ s.t. } \forall i, ||\mathbf{a}_x ||_0 \leq \beta \quad (3)$$

In equation (3) \mathbf{v}_{xi} is a vector $\in V_x$ and \mathbf{a}_x is a sparse vector with β non-zero coefficients. Similarly for vector $\mathbf{w}_{xi} \in W_x$ and the sparse vector \mathbf{a}'_x , reconstruction is given by equation (4)

$$\text{Min} \sum ||\mathbf{w}_{xi} - \mathbf{D}_x \mathbf{a}'_x ||_2 \text{ s.t. } \forall i, ||\mathbf{a}'_x ||_0 \leq \beta \quad (4)$$

The rest of the paper is organized as follows. Section 2 gives a brief description of the background work. The section 3 describes the methodology employed. Section 4 presents the experimental results. The paper is concluded in section 5 with possible directions for improvisation of the current method.

II. Background work

Data mining paradigm based on compression allows parameter free or parameter light solutions to data mining tasks. It further helps in exploratory data mining rather than imposing presumptions on data [1]. Normalized Information Distance (NID) is based on the concepts of Kolmogorov’s complexity, which is a measure of randomness of strings based on their information content [11]. In algorithmic theory, the Kolmogorov complexity $K(x)$ of a string x , is defined as “the length of the shortest program capable of producing x on a universal computer”.

The quantity $K(x)$ is incomputable and therefore can be approximated by the compression algorithms. The Normalized Compression Distance (NCD) is based on this result and is given by:

$$NCD(x,y) = \frac{C(x,y) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}} \quad (5)$$

where, $C(x)$ is the length of the (lossless) compressed file of x , and $C(x, y)$ the length of compressed file obtained by concatenation of x and y . The concept is that, if x and y share a large amount of mutual information, then the compressor will reuse the repeating structures found in one, to compress the other. NCD metric has been successfully applied in clustering languages and music [2]. Researchers have also tried to apply the concept for classification of images. The successful application of NCD metric in the field of images is limited, as most of the image compressors are not a normal *compressor*. The definition of normal compressor can be found in the paper [3].

Another research founded on Kolmogorov principle has given a distance measure called CK-1 for classification of texture [12]. The researchers use the MPEG compressor to measure the similarity between two images by constructing a video consisting of both the images. A different research work has proposed similarity measure based on sparse representation [8]. The concept of sparseness of natural images is used to measure the degree of compression. Using the K-SVD algorithm, one dictionary per image is learnt. *Sparse Complexity* and *Relative Sparse Complexity* [8] is then used to measure compressibility of an image. This, in turn, is used to quantify similarity. Dictionary learning techniques have also been employed for image classification in other works [13] [14]. Another approach uses sparse reconstruction errors for measuring the similarity between two images [9]. The authors have proposed a similarity measure SSNR. Our work is based on the similarity measure described in the paper [9]. The accuracy for face recognition [9] is remarkable in comparison to state-of-the-art methods.



Fig. 2. Examples of few images from AT&T dataset used for face recognition.



Fig. 3. Sanity check: Image reconstruction using the dictionaries learnt.

III. Methodology

Our main objective is to parallelize the process of computing similarity between images using the similarity measure given in [9]. It is worth noting that the image reconstruction process using the Orthogonal Matching Pursuit algorithm involves matrix computation of higher dimensions. The size of matrices and number of computations involved thus make the parallel approach a suitable candidate for solving the problem faster. To test the efficiency of the parallel approach, task of face recognition is implemented on multiple nodes. The entire process of finding similar images from a given dataset involves the following two phases:

- Training Phase : To construct dictionary using K-SVD algorithm for the training set
- Testing Phase: To construct the dictionary for query image and compute similarity with all images in the training set.

Once the similarities are computed, k-NN with $k=3$ is used to classify the query image. The value of k is chosen based upon the previous research work in face recognition [9]. Considering the speed of the testing phase in real time system, the main focus is on the testing aspect of the problem. The training phase can also be parallelized in the similar manner, but it is out of the scope of this paper. We exploit the concept of data parallelism or single instruction multiple data (SIMD) computation. The calculation of similarity for each image involves four reconstruction operations (as mentioned in equation 2). Following are the four reconstructions:

- 1) Reconstruction of dataset image patches using dataset dictionary
- 2) Reconstruction of dataset image patches using query dictionary
- 3) Reconstruction of query image patches using dataset dictionary
- 4) Reconstruction of query image patches using query dictionary

Equations (3) and (4) are the basis for the reconstruction task. The paper proposes to implement the function of similarity measure (which includes the above 4 sub tasks) in parallel on a cluster of four nodes. Each of the node in turn uses its own multiple cores (in this case 4 cores) to implement the method. Thus the task of similarity computation of n images is divided among the 4 nodes on the cluster in a load balancing fashion.

IV. Experimental validation

In this section we measure the efficiency of our method in terms of speedup and scaleup. In order to measure the speedup, the data set size is kept constant and number of computers on the cluster are changed. Scaleup measures the scalability of the system and is calculated by increasing both the size of data set and the number of computers on the cluster proportionally. The training data has to be communicated to all the nodes in cluster. This communication process occurs only once and is considered as a part of system startup. Hence, the communication cost for training data has been neglected in computation of speedup and scaleup. At the runtime, only the query data needs to be sent to the nodes. It can be noted that, this runtime communication cost is almost same for the various number of nodes we considered. Therefore, the speedup and scaleup for varying number of nodes in the cluster is determined largely by the processing time. The experiments are performed using R 3.2.2 on a cluster of 4 computers, each having the following specification: 8 GB RAM and 3.0 GHz Intel core i5 processor. Additional R packages viz. parallel, EBImage [15] and pixmap are used for parallelization and image processing. The 'pixmap' package is used for reading bitmap images and converting it to grayscale images for color invariance. Similarly the 'EBImage' is used for processing jpeg images. The 'parallel' package is integrated as a part of R from R 2.14.0 onwards. This package is based on 'multicore' and 'snow' packages. In this paper, a set of worker processes are created that listens to the master node commands in the cluster via sockets. As a sanity check, the images are reconstructed using its own dictionary to validate the reconstruction process. Fig. 3 shows an example of the sanity check performed. The number of K-SVD iterations and structural similarity (SSIM) values are also as indicated in the figure.

Table 1. Processing time in minutes

Nodes per Cluster	Time for 70 images	Time for 140 images	Time for 210 images	Time for 280 images
1	2.13	4.22	6.23	8.37
2	1.05	2.06	3.05	4.06
3	0.71	1.35	2.02	2.67
4	0.56	1.02	1.51	2.01

The dataset used for validation is AT&T face dataset [16]. The dataset consists of 400 grayscale images. The images belong to 40 individuals with different facial expressions. Each individual category has 10 images each. Fig 2 shows few examples from the dataset. To learn a dictionary, a total of 3000 patches of dimensions 8×8 are extracted from each image. Dictionary of dimensions 64×128 is learnt using $\beta = 8$. A total number of 10 K-SVD iterations are used. The results are tabulated in table 1. The speedup for 140, 210 and

280 images are almost identical and hence they overlap in the graph. The graph of speedup and scaleup is given in fig. 4(a) and 4(b).

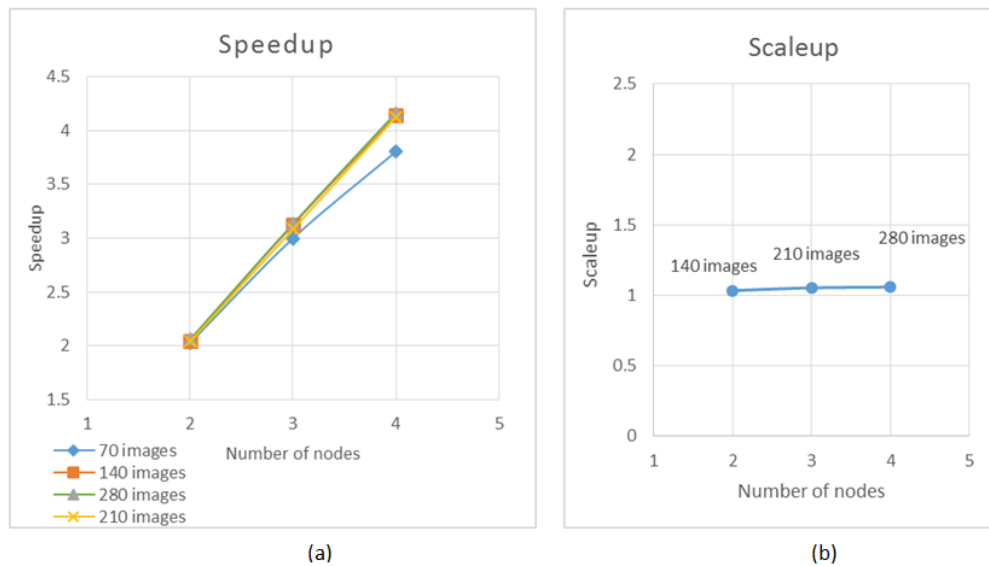


Fig. 4 (a) Speedup graph for the method; (b) Scaleup graph for the method

V. Conclusion

Image similarity using dictionary learning techniques has attracted significant attention from researchers lately. The dictionary and sparse coding are not yet fully optimized in terms of memory and speed. Our experimental results are based on data parallelism and does not involve any sophisticated parallelism approach for dictionary learning. The experimental results show that the speedup and scaleup is almost linear when the system initialization communication cost is not taken into account. A considerable amount of speedup is shown with the increase in the size of dataset. A perfect balance between the number of nodes and dataset size is the key to an almost linear speedup and scaleup, without which a communication overhead occurs. Further work on using other techniques for parallelization using map-reduce paradigm can be taken up. The communication overhead, which occurs during the system startup can also be improvised using a different memory architecture.

References

- [1]. M.Li, X. Chen,X.Li,B.Ma, P.Vintayi, The similarity metric, *IEEE Trans. Inf. Theory*, 2004, 50:3250-3264
- [2]. R.Calibrasi,P.Vintayi, Clustering by compression, *IEEE Trans.Inf.Theory*, 2005, 51:1523-1545
- [3]. Cebrian Manuel,Alofonseca Manuel, Ortega Alfonso, Common pitfalls using the normalized compression distance : what to watch out for in a compressor?, *Communications in Information & Systems*, 2005, 5:367-384
- [4]. Elad Michael, on the role of sparse and redundant representations in image processing, *IEEE proceedings*, 2010, 15: 3736-3745
- [5]. K.P. Soman, R. Ramanathan, Digital signal and image processing – The sparse way, *Isa publication*, 2012.
- [6]. Ori Bryt, Elad Michael, Compression of facial images using the K-SVD algorithm, *Elsevier*, 2008, 19:270-282
- [7]. Qiang Qiu, Patel Vishal,Chellapa Rama, Information-theoretic dictionary learning for image classification, *IEEE Transactions on pattern analysis and machine learning*, 2014, 36:2173-2184
- [8]. Guha Tanaya, Ward Rabab K., Image similarity using sparse representation and compression distance, *IEEE Transactions on multimedia*, 2014, 16: 980-987
- [9]. Guha Tanaya, Ward Rabab K., Aboulnasr Tyseer, Image similarity measurement from sparse reconstruction errors, *ICASSP*, 2013, 1937-1941
- [10]. Aharon Michal, Elad Michael,Bruckstein Alfred, K-SVD : An algorithm for designing overcomplete dictionaries for sparse representations, *IEEE Transactions on signal processing*, 2006, 54:4311-4322
- [11]. Vitányi Paul, M Balbach, F. J., Cilibrasi, R. L. & Li, M., Normalized information distance In *Information theory and statistical learning*, Springer, 2009, 45-82.
- [12]. Campana, Bilson JL, and Eamonn J. Keogh, A compression-based distance measure for texture, *Statistical Analysis and Data Mining*, 2010, 6: 381-398.
- [13]. Zhang Qiang, Li Baoxin, Discriminative K-SVD for dictionary learning in face recognition, *CVPR*, 2010, 2691-2698
- [14]. Yang Meng, Zhang Lei, Yang Jian, Zhang David, Robust sparse coding for face recognition, *CVPR*, 2011, 625-632
- [15]. Pau G, Fuchs F, Sklyar O, Boutros M and Huber W, EBImage-an R package for image processing with applications to cellular phenotypes, *Bioinformatics*, 2010 ,27:979-981
- [16]. AT&T, AT&Tarchive. [Online] Available at: http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.zip [Accessed October 2015].