# Online Dictionary Learning using Biogeography-based Optimization for Sparse Representation

Morteza Kolali Khormuji
Young Researchers and Elite Club, Islamic Azad University, Bushehr Branch, Bushehr, Iran

Mehdi Sadeghzadeh
Islamic Azad University, Computer Department, Mahshahr Branch, Mahshahr, Iran

## ABSTRACT

Computational visual attention modeling is a topic of increasing importance in machine understanding of images. The model with the $\ell$-0 norm as constraint is an NP hard problem. How to find the global optimal solution is a difficult point of this area. For Biogeography-based optimization (BBO) is good at solving NP hard problem, a dictionary learning method based on it is proposed in this paper. Biogeography-based optimization (BBO) algorithm is a new category of optimization technique based on biogeography concept. This population-based algorithm uses the idea of the migration strategy of animals or other species for solving optimization problems. The samples are first classified randomly for generate original population and residual of approximate the sample class with a rank-1 matrix as habitat suitability index (HSI) is calculated. Then, select better individuals using league matches. After that new individuals are generated from migration operators and mutation and the residual of the representation is used as data samples for training the dictionary for the next layer. The experimental results show the algorithm are effective.

### General Terms:

Evolutionary Computing, Pattern Recognition, Digital Image Processing

### Keywords:

Biogeography-based optimization, Sparse Representation, Dictionary Learning

## 1. INTRODUCTION

Recently, there has been growing interests in developing systems to automatically analyze video and image data. Although many algorithms try to solve tracking problems, from statistics view to human vision system view, it has not been solved thoroughly yet. The main problem in tracking is that the object may undergo occlusion, deterioration or even variation in its shape, thus, the need for a robust algorithm is obvious.

Sparse representation [1, 2] has been successfully applied in many pattern recognition applications as a part-based data representation method, including face recognition [3], speech recognition [4], handwritten digit recognition [5], image clustering [5], etc. Some of these algorithms just work in stationary discriminative background and some others can adapt themselves to changes of background or object. In recent years, a new approach has been introduced to solve the tracking problem as a classification problem [6]. The use of sparse coding and dictionary learning in computer vision is inspired by works in the neuroscience community [7] and researchers in machine vision tried to adapt it such as works presented in [8, 9, 10]. These works vary from image representation to classification by using dictionary learning. Although there is a controversy that if sparse coding is really relevant to classification or not, but the out coming results of this method on classification are promising [11].

So far there have been two general approaches for signal representation: the first is orthogonal and bi-orthogonal dictionaries due to their mathematical simplicity[12], such as curvelet, contourlet, wedgelet and bandlet. Most of these transforms have their fast transform algorithms, so they are widely used in image segmentation, denoising and inpainting. But the applications of these transforms depend too much on geometrical characteristics. For the natural images which always have complex geometrical characteristics, this type of dictionaries obviously lack flexibility.

The other one is learning dictionaries from samples, such as K-SVD[13]. Although from mathematical theory there are only one global optimal dictionary and its represent coefficients[14], how to find them is still a difficult problem.

For its excellent performance of solving many NP hard problems in the past years, Biogeography-based optimization (BBO) is used for learning dictionaries for giving samples. Although we can not prove the dictionary is the only global optimal one from mathematical theory, the experimental results show that it performs well.

## 2. PROBLEM SPECIFICATION DICTIONARY-LEARNING ALGORITHMS

Sparse representations have become a very active research topic in recent years. Many new algorithms have been developed that take advantage of sparse representations to achieve state-of-the-art results in a wide range of image processing applications including inpainting, denoising, and compression.

A sparse representation scheme consists of $(i)$ a generally overcomplete basis (called the $dictionary$) and $(ii)$ an algorithm that selects basis vectors (called the $atoms$) and weighting coefficients to produce a linear approximation of an input signal. The representation is termed sparse because only a small number of atoms / coefficients will be selected by the representation algorithm. The basic idea is to learn the dictionary adaptive to the target image so as to achieve better sparsity than the fixed ones. Most existing dictionary learning methods consider an over-complete dictionary and formulate the learning process as a minimization problem.

We now turn to discuss the learning methodology for constructing $\mathbf{A}$. Assume that a training database $\{y_i\}_{i=1}^{M}$ is given, and thought to have been generated by some fixed but unknown model $\{M_{\mathbf{A},k_0,\aleph}\}$. Can this training database allow us to identify the generating model, and specifically the dictionary $\mathbf{A}$?

## 2.1 The K-SVD Algorithm

A different update rule for the dictionary can be proposed, in which the atoms (i.e., columns) in A are handled sequentially. This leads to the $K-SVD$ algorithm, as developed by Aharon et al. Keeping all the columns fixed apart from the $j_0$-th one, $a_{j_0}$, this column can be updated along with the coefficients that multiply it in $\mathbf{X}$. We isolate the dependency on $a_{j_0}$ by rewriting as 1

$$A_k = \arg\min_{\mathbf{A}} \|Y - AX_k\|_F^2 = YX_k^T(X_kX_k^T)^{-1} = YX_k^+ \quad (1)$$

The optimal $a_{j0}$ and $x_{j0}^T$ minimizing Equation 2 are the rank-1 approximation of $E_{j0}$, and can be obtained via an SVD, but this typically would yield a dense vector $x_{j0}^T$, implying that we increase the number of non-zeros in the representations in X. In order to minimize this term while keeping the cardinalities of all the representations fixed, a subset of the columns of $E_{j0}$ should be taken those that correspond to the signals from the example-set that are using the $j_0$-th atom, namely those columns where the entries in the row $X_{j0}^T$ are non-zero. This way, we allow only the existing non-zero coefficients in $X_{j0}^T$ to vary, and the cardinalities are preserved.

$$\|Y - AX_k\|_F^2 = \|Y - \sum_{j=1}^{m} a_j x_j^T\|_F^2 = \|(Y - \sum_{j\neq j0} a_j x_j^T) - a_{j0}x_{j0}^T\|_F^2 \quad (2)$$

$$E_{j0} = Y - \sum_{j\neq j0} a_j x_j^T \quad (3)$$

The optimal $a_{j0}$ and $x_j0^T$ minimizing Equation 2 are the rank-1 approximation of $E_{j0}$, and can be obtained via an SVD, but this typically would yield a dense vector $x_j0^T$, implying that we increase the number of non-zeros in the representations in $\mathbf{X}$. Therefore, we define a restriction operator, $P_{j0}$, that multiplies $E_{j0}$ from the right to remove the non-relevant columns. The matrix $P_{j0}$ has M rows (the number of overall examples), and $M_{j0}$ columns (the number of examples using the $j0$-th atom). We define $(x_{j0}^R)^T = x_{j0}^T P_{j0}$ as the restriction on the row $x_{j0}^T$ , choosing the non-zero entries only.

## 2.2 Biogeography-Based Optimization algorithm

The BBO algorithm was first proposed by Simon in 2008 . The basic idea of this algorithm was inspired by biogeography, which refers to the study of biological organisms in terms of geographical distribution (over time and space). The case studies might include different islands, lands, or even continents over decades, centuries, or millennia. In this field of study, different ecosystems (habitats or territories) are investigated to find the relationships between different species (habitants) in terms of $immigration$, $emigration$, and mutation. The evolution of ecosystems to reach a stable situation while considering different kinds of species (such as predator and prey), and the effects of migration and mutation was the main inspiration for the $BBO$ algorithm.

In the science of biogeography, a habitat is an ecological area that is inhabited by a particular plant or animal species and which is geographically isolated from other habitats. Each habitat is classified by $Habitat\ Suitability\ Index\ (HSI)$. Areas or habitats which are well suited as residences for biological species are said to have a high $HSI$ while habitats that are not good have low $HSI$. The value of $HSI$ depends upon many features of habitat like rainfall, temperature, diversity of vegetation, land area, safety and security. If each of the features is assigned a value, $HSI$ is a function of these values. Each of these features that characterize habitability is known as $Suitability\ Index\ Variables\ (SIV)$. $SIVs$ can be considered the independent variables of the habitats, and $HSI$ can be considered the dependent variable.

Habitats with high $HSI$ have large population, high emigration rate $\mu$, simply by virtue of large number of species that migrate to other habitats. The immigration rate $\lambda$ is low for these habitats as these are already saturated with species. On the other hand, habitats with low $HSI$ have high immigration rate $\lambda$, low emigration rate $\mu$ because of sparse population. The value of $HSI$ of low $HSI$ habitat may increase with the influx of species from other habitats as suitability of a habitat is function of its biological diversity. But if $HSI$ does not increase and remains low, species in that habitat go extinct and this leads to additional immigration. For sake of simplicity, it is safe to assume a linear relationship between a habitat $HSI$ and its immigration and emigration rate and also that the rates are same for all the habitats. The immigration and emigration rate depends upon the number of species in the habitats.

The values of emigration rate $\mu_a$ 4 and immigration rate $\lambda_a$ 5 are given as:

$$\mu_a = \frac{E \times n}{N} \quad (4)$$

$$\lambda_a = 1 \times \frac{1-n}{N} \quad (5)$$

The other component of $BBO$, mutation, improves the exploration of $BBO$ and keeps habitats as diverse as possible. This component is defined as follows:

$$m_k = M \times (1 - \frac{P_k}{P_max}) \quad (6)$$

where $M$ is an initial value for mutation defined by the user, $p_n$ is the mutation probability of the nth habitat, and $p_max = argmax_{(p_k)}, k = 1, 2, ..., K$. The general steps of the $BBO$ algorithm are illustrated in the flowchart of Fig. 2. This figure shows that the $BBO$ algorithm starts with a random set of habitats. After calculating the $HSI$ of each habitat, the emigration, immigration, and mutation rates are updated. The non-elite habitants are migrated and mutated according these rates. A pre-defined number of the best habitats are saved as elites for the next generation. Finally, the $BBO$ algorithm is terminated by the satisfaction of a termination criterion. Note that elitism prevents the best solutions from being corrupted by immigration. To do this, we retain some of the best solutions (habitats) at each iteration. So, the best solutions can be recovered if their $HSI$ is ruined by mutation.

# 3. DICTIONARY TRAINING METHODS

## 3.1 Hierarchical dictionary learning algorithm

Let Z=[$z_1$,$z_2$,$z_3$,...,$z_n$] $\in$ $R^{m\times n}$ be data samples, with $z_1$,$z_2$,$z_3$,...,$z_n$ as its column vectors. If the data samples are images, we should change them into one-dimensional vectors. Assume $a$ is an image, which is $4 \times 4$ and $\acute{a}$ is the transformed vector, for the continuity of the signals, the following transform is employed.

$$A_{4,4} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix} \tag{7}$$

$\acute{a} = [a_{1,1}, a_{2,1}, a_{3,1}, a_{4,1}, a_{4,2}, a_{3,2}, a_{2,2}, a_{1,2}$
$, a_{1,3}, a_{2,3}, a_{3,3}, a_{4,3}, a_{4,4}, a_{3,4}, a_{2,4}, a_{1,4}]$

Then hierarchical dictionary training method is used. First, $D^1 = [d_1^1, d_2^1, ..., d_{z1}^1] \in R^{m\times z1}$ is generated for $\min_{\mathbf{D^1}} \|Z - D^1X^1\|^2$ with $X^1 = [x_1^1, x_2^1, ..., x_n^1]$ as the representation coefficients. $sp(.)$ denotes $\ell_0$-norm.

$$sp(x_1^1) = sp(x_2^1) = ... = sp(x_n^1) = 1 \tag{8}$$

So we need to cluster the data samples $z_1$,$z_2$,$z_3$,...,$z_n$ for finding the data sample clusters who share the same atoms. This method is similar to K-means. But when clustering uses $k - SVD$ and $Biogeography - Based\ Optimization$, the process can be prevented to fall into local minimum points. After training, the atoms of the first layer and residuals $R = Y - D^{z1}X^{z1}$ are calculated. Then the residuals can be used as new data samples as $\acute{y} = R$ for training the atoms $D^2 = [d_1^2, d_2^2, ..., d_{z2}^2]$ for the next layer satisfying $\min_{\mathbf{D^2}} \|\acute{Y} - D^2X^2\|_2$ With the same method, $m$ layer atoms can be trained until the satisfied SNR is achieved. The final dictionary $D$ is constructed by $D = [D^1, D^2, ..., D^r]$.

## 3.2 Dictionary learning with K-SVD and BBO

In Equation 9 , the underlying dictionary $\mathbf{A}$ is assumed known, being the redundant DCT. We are supposed to minimize this function with respect to both the sparse representations $q_k$, and the overall output image $\mathbf{z}$. As done before, we adopt a block-coordinate minimization algorithm that starts with an initialization $z = M^Ty$, and then seeks the optimal $\hat{q}_k$.

$$\{\{\hat{q}_k\}_{k=1}^{M}, \hat{y}\} = \arg\min_{z,q_k} \lambda\|Mz - y\|_2^2 + \tag{9}$$

$$\sum_k \mu_k\|q_k\|_0 + \sum_k \|Aq_k - R_kz\|_2^2$$

notice that we have turned the quadratic penalty in Equation 9 into a constraint, thus removing the need to choose the parameter $\mu_k$. Also, the energy of the patch-error $Ap - q_k$ is evaluated using only the existing pixels in this patch, as indicated by the multiplication by $M_k = R_kM^TMR_k^T$ a local mask that corresponds to the $k$-th patch. $n_k = 1^TM_k1$. Thus, this stage works as a sliding window sparse coding stage, operated on each block of size $\sqrt{n} \times \sqrt{n}$ at a time, high performing.

Given all $\hat{q}_k$,we can now fix those and turn to update $z$. Returning to Equation 9, we need to solve

$$\hat{y} = \arg\min_z \lambda\|Mz - y\|_2^2 + \sum_k \|A\hat{q}_k - R_kz\|_2^2 \tag{10}$$

The K-SVD counterpart is better behaved. We target the update of the columns of A one at a time. we thus minimize:

$$Err(A) = \sum_{k\in\Omega_j} \|M_k(Aq_k - p_k)\|_2^2 \tag{11}$$

$$= \sum_{k\in\Omega_j} \|M_k(Aq_k - a_jq_k(j) - p_k) + M_ka_jq_k(j)\|_2^2$$

We denote $y_k^j = p_k - Aq_k + a_jq_k(j)$. This is the residual in the representation of the $k$-th patch, where all the atoms apart from the $j$-th are used. Thus, our task would be:

$$\min_{a_j\cdot q_k(j)} \sum_{k\in\Omega_j} \|M_k(y_k^j - a_jq_k(j))\|_2^2 \tag{12}$$

While the above could be posed as a rank-1 approximation problem, a simpler approach is a short iterative (2-3 iterations) process where we update the one unknowns alternating. The update of $q_k(j)$ is obtained by:

$$\hat{q}_k(j) = [a_j^TM_ka_j]^{-1}a_j^TM_ky_k^j \tag{13}$$

We now, The rank-k approximate matrix $A_k$ of $A$, $K < r = rank(A)$. $A_k = \sum_{i=1}^{k} M_iq_ia_i$ , $k < r$. B is arbitrary matrix, $\|.\|_F$ is Frobenius norm of matrix.

$$\min_{rank(B)=k} \|A - B\|_F^2 = \|A - A_k\|_F^2 = M_{k+1}^2 + M_{k+2}^2 + ... + M_r^2 \tag{14}$$

It can be concluded from Equation 14 that $A_1 = M_1q_1a_1$ is the best rank-1 approximation of $A$.

*3.2.1 Generating the original populations.* Code the data samples $Z = [z_1, z_2, ..., z_n]$, such as $z_1$ is No.1,...,$z_n$ is No.n. Every data sample can choose its class randomly. For example, the $Habitat_j$ is the $j$th individual of the original population. It can be described as following matrix $Habitat_j$

$$Habitat_j = \begin{pmatrix} 1 & 7 & 8 & 9 \\ 6 & 2 & 3 & 10 \\ 4 & 5 & 0 & 0 \end{pmatrix}$$

The matrix $Habitat_j$ means that the data samples is classified into three classes. The first class includes data samples of No.1, 7, 8 and 9, the second class includes No.6, 2, 3 and 10, and the third class includes No.4 and 5. In order to improve the diversity of the population, each individual needn't be the same class number, but the class number should be in some range. The individual matrix should insure each number appearing only once. In that way, the original population is generated as $\{Habitat_j\}_{j=1}^{num}$.

*3.2.2 Calculate the Habitat Suitability Index (HSI).* Choose each sample class to form a matrix and k-SVD. As the above example $Habitat_j$, the data sample can be separated into:
$Z_1 = [z_1, z_7, z_8, z_9]$
$Z_2 = [z_6, z_2, z_3, z_10]$ and $Z_3 = [z_4, z_5]$

and $Z_1, Z_2, Z_3$ are employed with k-SVD $Z_1 = Q^{z_1}S^{z_1}A^{z_1}$, $Z_2 = Q^{z_2}S^{z_2}A^{z_2}$, $Z_3 = Q^{z_3}S^{z_3}A^{z_3}$. $Q_1^{Z_1}$ is the first column of $Q^{Z_1}$, $M_1^{Z_1}$ is $S^{Z_1}(1,1)$, and $a_1^{Z_1}$ is the first column of $A^{Z_1}$, then we get $\widetilde{Z^1} = q_1^{Z_1}M_1^{Z_1}a_1^{Z_1}$.

$\widetilde{Z^1}$ is the best rank-1 approximation of $Z^1$, $\widetilde{Z^2}$, can be obtained by the same method. By defining $HSI_{zi,t}$ as the Habitat Suitability Index of solution $zi, t$ and $hsi_z i(s), t$ as the contribution of $z_{i(s),t}$ to $z_{i,t}$, where $s \in [1, 2, ..., v]$, we obtain:

$$HSI_{zi,t} = H\{hsi_{z_{i(1),t}}, , hsi_{z_{i(2),t}}, ..., hsi_{z_{i(v),t}}\} \tag{15}$$

where $\mathbf{H}$ is the function to calculate $HSI_{zi,t}$.
Define $hsi_{z_{j(s),t}} \geqslant hsi_{z_{i(s),t}}$ to denote that $z_{j(s),t}$ contributes more than or equally with $z_{i(s),t}$ to $HSI_{zi,t}$, where equality holds when $z_{j(s),t}$ contributes equally with $z_{i(s),t}$ to $HSI_{zi,t}$.

After that we can get the HSI Error, $HSI{-}err^{Z_1} = \|Z^1 - \widetilde{Z}^1\|_F^2$, $HSI - err^{Z_2}$ and $HSI - err^{Z_3}$. $HAB.dB_j$ is the HSI of the approximation, and it can be used as habitat suitability index of the individual.

$$HAB.dB_j = \tag{16}$$

$$10 \log_{10}(\frac{\|Z\|_F^2}{(HSI - err^{Z_1} + HSI - err^{Z_2} + HSI - err^{Z_3})})$$

With this method, every individual can achieve its Habitat Suitability Index (HSI) $\{HAB.dB_j\}_{j=1}^{num}$.

*3.2.3 Selection , migration and mutation.* For the convenience of computation, we choose a fixed number of individuals after each iteration. Experiments show that league match is the best selection method.
We employ per defined shares. For example, in the $n$th iteration there are two Habitat individuals, $Habitat_i^n$ and $Habitat_k^n$.

$$Habitat_i^n = \begin{pmatrix} 1 & 7 & 8 & 9 \\ 6 & 2 & 3 & 10 \\ 4 & 5 & 0 & 0 \end{pmatrix}$$

$$Habitat_k^n = \begin{pmatrix} 1 & 2 & 6 & 8 & 0 & 0 & 0 & 0 \\ 9 & 7 & 3 & 4 & 5 & 10 & 0 & 0 \end{pmatrix}$$

After Migration operators (immigration and emigration), they can be changed into $Habitat_i^{n+1}$ and $Habitat_k^{n+1}$.

$$Habitat_i^{n+1} = \begin{pmatrix} 1 & 7 & 8 & 9 & 0 & 0 & 0 & 0 \\ 6 & 2 & 3 & 10 & 0 & 0 & 0 & 0 \\ 9 & 7 & 3 & 4 & 5 & 10 & 0 & 0 \end{pmatrix}$$

$$Habitat_k^{n+1} = \begin{pmatrix} 1 & 2 & 6 & 8 & 0 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

For each data sample must be included in one class, we should drop the repeated number and add the missed number. After that , we get

$$Habitat_i^{n+1} = \begin{pmatrix} 1 & 7 & 8 & 9 & 0 & 0 & 0 & 0 \\ 6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 & 5 & 10 & 0 & 0 \end{pmatrix}$$

$$Habitat_k^{n+1} = \begin{pmatrix} 1 & 2 & 6 & 8 & 0 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Then, drop unnecessary zeros.

$$Habitat_i^{n+1} = \begin{pmatrix} 1 & 7 & 8 & 9 \\ 6 & 2 & 0 & 0 \\ 3 & 4 & 5 & 10 \end{pmatrix}$$

$$Habitat_k^{n+1} = \begin{pmatrix} 1 & 2 & 6 & 8 & 7 & 9 \\ 4 & 5 & 3 & 10 & 0 & 0 \end{pmatrix}$$

In BBO, mutation is a probabilistic operator which is used for modifying one or more randomly selected $SIV$ of a solution based on its priori probability of existence $P_i$. In BBO, just like GA, this operator is used for increasing diversity among the population. In this algorithm, the mutation probability $m_i$ is calculated according to the solution probability, as in Equation17. Therefore, mutation probability and solution probability are proportioned inversely.

$$m_i = m_m ax(1 - \frac{p_i}{p_m ax}) \tag{17}$$

During mutation, We can change the position of the element position in the individual matrix. For example:

$$Habitat_i = \begin{pmatrix} 1 & 7 & 8 & 9 \\ 6 & 2 & 0 & 0 \\ 3 & 4 & 5 & 10 \end{pmatrix}$$

is mutated into

$$Habitat_i = \begin{pmatrix} 1 & 7 & 8 & 0 \\ 6 & 2 & 10 & 9 \\ 3 & 4 & 5 & 0 \end{pmatrix}$$

# 4. PROPOSED APPROACH

According to [15], the probability for non-immigration on Island $i$ is

$$Pr(non - immigration) = (1 - \lambda_i)$$

If there is no immigration on Island $i$, we obtain

$$Pr(f_{yi(s),t^+} = f_{yi(s),t^-}) = 1$$

$$Pr(f_{yi(s),t^+} > f_{yi(s),t^-}) = 0$$

Hence, if there is no immigration on Island $i$,

$$Pr_{non-immigration}(f_{yi(s),t^+} \geqslant f_{yi(s),t^-}) = (1 - \lambda_i)$$

On the other hand, the probability that immigration occurs on Island $i$ is $Pr(immigration) = \lambda_i$.

**Theorem1**.. For $y_{i,t}$ by defining $E_{obtain}[Pr(f_{yi(s),t^+} > f_{yi(s),t^-})]$ as the expectation of probability of obtaining a new feature that is not worse than the old one. $E_{obtain}$ can be as large as possible if the following conditions are met,
a. is as large as possible;

$$\frac{\sum_{i=1}^{n} \sum_{j \in J_i(s)} \mu_j}{n \sum_{j=1}^{n} \mu_j}$$

b. $\lambda_i$ is as small as possible.
where n is the largest possible species count that the habitat can support.
Note: According to Theorem 4, enlarging $\frac{\sum_{i=1}^{n} \sum_{j \in J_i(s)} \mu_j}{n \sum_{j=1}^{n} \mu_j}$ and diminishing $\lambda_i$ can help solutions in $BBO$ obtain a new feature which is not worse than the old one.

According to pattern theory, suppose there is a pattern $H$ in a matrix[16].

$$Hab = \begin{pmatrix} a_1 & a_2 & \dots & a_m & * \\ * & * & \dots & * & * \\ * & * & \dots & * & * \end{pmatrix}$$

Table 1. Experiment Result Migration Operators

| Dictionary column number | population | immigration rate | emigration rate | HAB(dB) |
|---|---|---|---|---|
| [10,7] | 700 | 0.3 | 0.2 | 405.67 |
| [8,8] | 700 | 0.4 | 0.2 | 410.69 |

For convenience, our analysis is based on the following assumptions:

(1) Each individual matrix has the same row number.
(2) During the addition of missed elements, the number of pattern H can't be changed.
(3) Because the mutation rate is very low, it can be ignored when we analysis the change of pattern number.

Suppose each individual in the population has the same row number which means that data samples are separated into same class numbers. $m = m(H,t)$ is the number of pattern $H$ in the $t$th generation. The individuals which contain $H$ are $\{w_1, w_2, \ldots, w_m\}$ with average habitat suitability index (HSI) $HSI(H,t)$.

$$HSI(H,t) = \sum_{j=1}^{m} HSI\frac{w_j}{m} \tag{18}$$

The whole population of $t$-th generation is $P_t = \{w_1, w_2, \ldots, w_n\}$, so the individuals are selected with the probability of $p_i = \frac{HSI(w_i)}{\sum_{i=1}^{m} HSI(w_i)}$. The number of survived pattern $H$ after selection is $m_{Hab}(H,t+1)$.

$$m_{Hab}(H,t{+}1) = n.\frac{\sum_{j=1}^{m} HSI(w_j)}{\sum_{i=1}^{n} HSI(w_i)} = m(H,t).n.\frac{HSI(H,t)}{\sum_{i=1}^{n} HSI(w_i)} \tag{19}$$

When the average habitat suitability index of populations is HSI.

$$H\bar{S}I = \sum_{i=1}^{n} \frac{HSI(w_i)}{n} \tag{20}$$

After selection the number of survived pattern $H$ is $m_{Hab}(H,t+1)$.

$$m_{Hab}(H,t+1) = \frac{m(H,t).HSI(H,t)}{H\bar{S}I} \tag{21}$$

Suppose the immigration rate is $P_{immigration}$ and emigration rate is $p_{emigration}$ so $P_{immigration} \cdot p_{emigration} \cdot n$ will participate in the population after selection. Suppose the survived number after selection is $k$. The probability of the two individuals who participate in Migration operators (immigration and emigration) which all contains $H$ is $\left(\frac{m_{Hab}(H,t+1)}{k}\right)^2$, and as the premising the process of Migration will not damage the pattern $H$. The probability of the two individuals who anticipate Migration which one contains $H$ and the other does not is $\frac{m_{Hab}(H,t+1)}{k}\left(1 - \frac{m_{Hab}(H,t+1)}{k}\right)$, the probability of drop same element which damage $H$ is $P_d$.

Suppose the probability of the position of each element in matrix being at the lower half is 0.5 and the algorithm drops the repeat elements with the same probability, so the only condition that $H$ is not damaged is all of the elements is in the same half part of individual matrix and the drop same elements process does not choose the part which contains $H$ . In that case the probability of $H$ not being damaged is $\left(\frac{1}{2}\right)^{2m-1}$.

$$P_d = 1 - \left(\frac{1}{2}\right)^{2m-1} \tag{22}$$

After selection, the number of elements which contain $H$ is $m_{Migration}(H,t+1)$

$$m_{Migration}(H,t+1) = \left(2\left(\frac{m_{Hab}(H,t+1)}{k}\right)^2 \right. \tag{23}$$

$$\left. + \frac{m_{Hab}(H,t+1)}{k}\left(1 - \frac{m_{Hab}(H,t+1)}{k}\right)P_d\right).P_{Migration}.n$$

After migration and ignore mutation the whole number of elements which contain $H$ is $m(H+1,t)(m_{migration}(H,t+1)$ is the number of elements which contain $H$ obtained by migration operators (immigration and emigration).

## 5. EXPERIMENTAL RESULTS

In the experiment, ten images are randomly selected from the image dataset as the training set for the BBO optimization. The other images in the image dataset are used as the test set. For dictionary learning and BBO, we have chosen a reasonable set of value and have not made any effort in finding the best parameter settings.

(1) population size: NP = 700;
(2) habitat modification probability = 1;
(3) mutation probability: $m_max$ = 0.005;
(4) maximum Number of HSI Function Evaluations: ($MAX_{HSI}$)=150,000.

Moreover, in our experiments, each function is optimized over 50 independent runs. We also use the same set of initial random populations to evaluate different algorithms. All the algorithms are implemented in MATLAB. To improve the efficient of the experiment, data samples are generated only with $\{0,1\}$. The length of each signal is 4. We exhausted the 16 vectors.

$$Z = \begin{pmatrix} 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 1 & \ldots & 1 \\ 0 & 1 & 0 & \ldots & 1 \end{pmatrix}$$

Suppose we can train a dictionary $D$ and the represent coefficient is $X = [x_1, x_2, \ldots, x_{16}]$.

$$\min_{D,X} \|Y - DX\|_2^2$$

$$\|x_i\|_0 < T$$
$$i = 1, 2, ..., 16$$

We use a vector to denote the column numbers for learning, for example, [5,6,2] means that the first layer of dictionary is 5, the second layer dictionary is 6 and the third layer dictionary is 2. Experimental results are shown in table 1.

## 6. CONCLUSIONS

In this paper, we proposed a dictionary learning algorithm using Biogeography-Based Optimization (BBO) for sparse representation. The fundamental principle of this algorithm is to use K-SVD and Biogeography-Based Optimization (BBO) to find the global optimum point. The Biogeography-Based Optimization (BBO) is applied here to optimize the feature weight. Experiments show that the improved dictionary can be find with this method. But dictionaries for higher dimensional data samples have not been trained because of the high complexity.

## 7. REFERENCES

[1] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: NIPS, NIPS, 2007, pp. 801-808.

[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, Journal of Machine Learning Research 11 (2010) 19-60.

[3] J. Sun, Q. Zhuo, C. Ma, W. Wang, Sparse image coding with clustering property and its application to face recognition, Pattern Recognition 34 (2001) 1883- 1884.

[4] G. Sivaram, S. Nemala, M. Elhilali, T. Tran, H. Hermansky, Sparse coding for speech recognition, in: Proceedings 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2010, Signal Process. Soc., 2010, pp. 4346-4349.

[5] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Transactions on Image Processing 20 (5) (2011) 1327-1336.

[6] S. Avidan, ”Ensemble tracking,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 2, pp. 261-271, feb. 2007.

[7] B. Olshausen and D. Field, ”Sparse coding with an overcomplete basis set: A strategy employed by vi” Vision research, vol. 37, no. 23, pp. 3311-3325, 1997.

[8] I. Ramirez, P. Sprechmann, and G. Sapiro, ”Classification and clustering via dictionary learning with structured incoherence and shared features,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, june 2010, pp. 3501 -3508.

[9] X. Mei and H. Ling, ”Robust visual tracking and vehicle classification via sparse representation,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 11, pp. 2259-2272, nov. 2011.

[10] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, ”Robust face recognition via sparse representation,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 2, pp. 210-227, feb. 2009.

[11] R. Rigamonti, M. Brown, and Y. Lepetit, ”Are sparse representations really relevant for image classification” in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, june 2011, pp. 1545-1552.

[12] R. Rubinstein, A.M. Bruckstein, and M. Elad, ”Dictionaries for Sparse Representation Modeling,” 2010. 98(6): p. 1045-1057.

[13] M. Aharon, M. Elad, and A. Bruckstein, ”k-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” Signal P rocessing, IEEE Transactions on, vol. 54, no. 11, pp. 4311-4322, nov. 2006.

[14] M. Aharon, M. Elad, and A. Bruckstein, ”On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them,” Linear Algebra and its Applications, vol. 416(1): pp.48-67,2006.

[15] D. Simon, Biogeography-based optimization, IEEE Transactions on Evolutionary Computation 12 (6) (2008) 702713.

[16] Minqiang Li, Jisong Kou, Basis theory of genetic algorithm and its application, 2002.