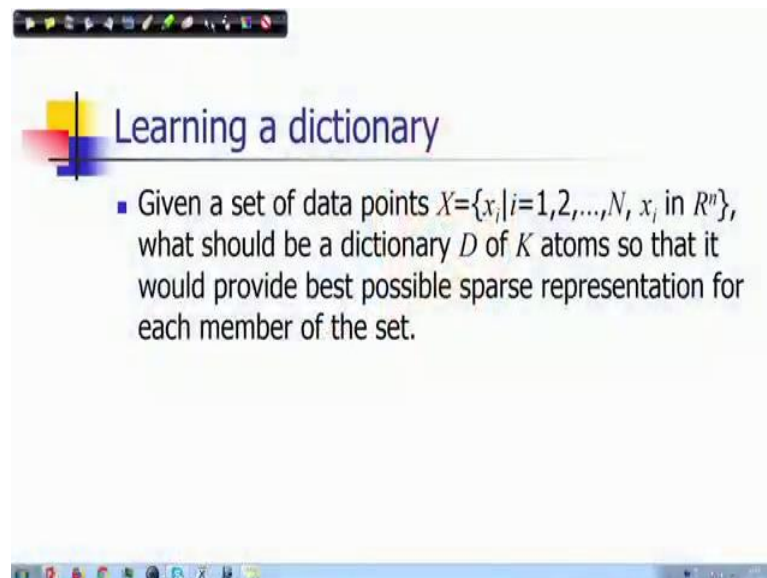


**Computer Vision**  
**Prof. Jayanta Mukhopadhyay**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 54**  
**Dimension Reduction and Sparse Representation Part – IV**

We are discussing about Sparse Representation of signal or input data vector. And in the last lecture we have discussed given a dictionary how we can obtain sparse representation of a signal using different pursuit algorithms particularly we have discussed about orthogonal matching pursuit algorithms.

(Refer Slide Time: 00:31)



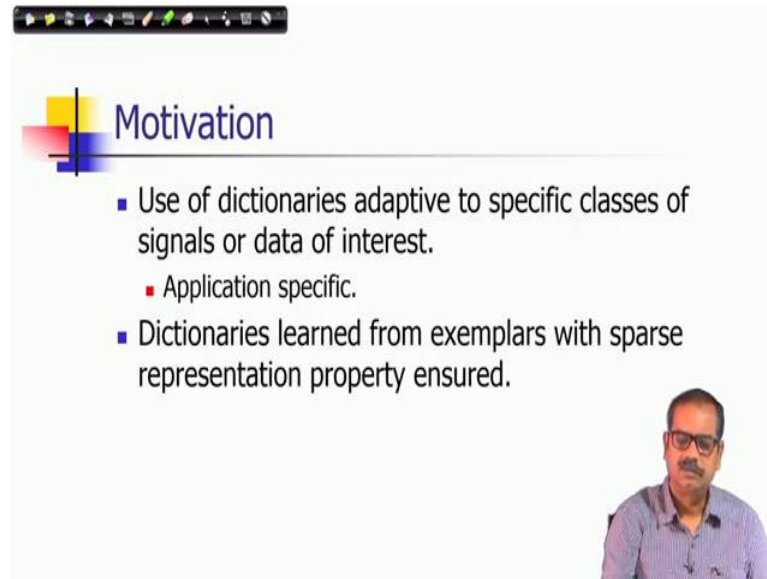
**Learning a dictionary**

- Given a set of data points  $X = \{x_i | i=1, 2, \dots, N, x_i \text{ in } R^n\}$ , what should be a dictionary  $D$  of  $K$  atoms so that it would provide best possible sparse representation for each member of the set.

Today I will discuss about another problem of sparse representation, here we would like to learn the dictionary itself. So, in the previous case we have considered the dictionary is given and given the dictionary of say  $K$  atoms we wanted to get a sparse representation given any input signal of dimension  $n$ .

Now, in this case the problem is that we would like to learn a dictionary specific to a set of data. So, given a set of data points here we have considered the symbol  $X$  as a set and  $x_i$  which is representing  $n$  dimensional data vector. Suppose you have  $N$  number of such data vectors in that set, then what should be a dictionary  $D$  of  $K$  atoms so that it would provide best possible sparse representation for each member of the set.

(Refer Slide Time: 01:41)



The slide is titled "Motivation" in a blue serif font. To the left of the title is a small graphic consisting of overlapping yellow, red, and blue squares with a black crosshair. Below the title, there is a list of two main bullet points, each preceded by a blue square. The first bullet point is "Use of dictionaries adaptive to specific classes of signals or data of interest." and it has a sub-bullet point preceded by a red square: "Application specific." The second main bullet point is "Dictionaries learned from exemplars with sparse representation property ensured." In the bottom right corner of the slide, there is a small video inset showing a man with glasses and a mustache, wearing a blue shirt, looking towards the camera.

- Use of dictionaries adaptive to specific classes of signals or data of interest.
  - Application specific.
- Dictionaries learned from exemplars with sparse representation property ensured.

Motivation of learning dictionary could be several that could be several such motivations. In particular you would like to learn dictionary adaptive to specific classes of signals or data of interest. So, that would make a dictionary suitable for certain application and would give you relatively better performance compared to any standard fix dictionary. And, this dictionary learns from exemplars and it also ensures that sparse representation properties are insured using this dictionaries.

So, sometimes it may happen that if we use a fixed dictionary given any input data representation sparse representation may not be possible with that set of atoms in the dictionary. But, if we can tune the dictionary for a specific classes of data set then most likely we will be getting a better sparse representation of data. So, this is also another motivation.

(Refer Slide Time: 02:58)

**Problem statement**

$$n \times N \rightarrow X = [x_1 \ x_2 \ \dots \ x_N] \quad x_i \in \mathbb{R}^n$$

$$n \times K \rightarrow D = [d_1 \ d_2 \ \dots \ d_K] \quad d_i \in \mathbb{R}^n$$

$$K \times N \rightarrow Y = [y_1 \ y_2 \ \dots \ y_N] \quad y_i \in \mathbb{R}^K$$

- To obtain a sparse  $Y$  in  $\mathbb{R}^K$  such that
  - $X = DY$ , or  $X \sim DY$

Handwritten diagram illustrating the linear combination of dictionary atoms  $d_i$  to represent data points  $x_j$ :

$$\begin{bmatrix} \vec{x}_1 & \vec{x}_2 & \dots & \vec{x}_N \end{bmatrix} = \begin{bmatrix} d_1 & d_2 & \dots & d_K \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix}$$

$$\vec{x}_1 = \sum_{i=1}^K a_{1i} d_i \quad \vec{x}_2 = \sum_{i=1}^K a_{2i} d_i$$

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse

So, just precisely let us define the problem statement here for this dictionary learning, we consider a data as an  $n$  dimensional vector and you consider a set  $X$  which consists of  $N$  number of data points and let us represent it in a matrix form. So, we can consider a matrix  $X$  by each column represent the data and each column is of  $n$  dimension.

And, then we are considering there are dictionaries of  $K$  atoms and again each atom is of  $n$  dimension whereas, it has to be the same as the dimension of the data we would like to have linear combination of these atoms to represent the data. And, that linear combination should be sparse and the coefficient of linear combination is represented by vectors  $y_i$ 's.

So,  $Y$  is also another matrix where each column of  $Y$  represents the sparse representation of data  $X$ . So, the dimension of  $Y$  is  $K$  dimensional vector because  $K$  dimensional representation of data point in this case and so, the column vector of the matrix  $Y$  of dimension  $K$ . So, our problem is that we would like to obtain a sparse  $Y$  in the  $K$  dimensional real space such that  $X$  equals  $DY$ .

So, this expresses itself the linear combination of dictionary atoms for each data point and the coefficient of linear combinations they are coming from the columns of  $Y$  or it could be approximate. So, let us consider the let us make a dimensional check of this particular fact; that means, we are trying to factorize the input matrix data matrix  $X$  into two factors. One part should give me a dictionary of  $K$  atoms, the other part should give me a dictionary of  $N$  sparse representation for  $N$  data points;  $n$  here is  $N$ .

So, we consider the dimension of X this is say  $n \times N$ , then dimension of D this is  $n \times K$  as we can see that  $n$  is the size of the column vectors. So, that is why that many rows should be there that is  $n$  and  $K$  is the number of atoms in dictionary D. So, that is why its number of columns should be  $K$  columns. Similarly for  $X_n$  should be the size of the row because  $n$  represents the dimension of the data point and  $N$  is the number of data points in X.

So, number of columns in X is  $N$ . Similarly for Y it should be  $K \times N$ . So,  $K$  once again is the dimension of sparse representation of data; that means, a few of those elements in that column vector Y should be nonzero rest of them should be 0 that is sparsity that is a interpretation of sparsity very few of them should be nonzero.

But the dimension could be very large, it could be quite large from  $n$  and that is a dimension of data original dimension of data and then  $N$  is the number of sparse represented data vectors. So, you can see the multiplication  $DY$  means  $n \times K$  matrix multiplied by  $K \times N$  matrix you get the dimensional matching here because you get  $n \times N$  data matrix.

And let us also understand that how this representation is to be interpreted. So, we are considering that the data vectors  $[x_1, x_2, \dots, x_N]$  and say the dictionaries are  $[d_1, d_2, \dots, d_K]$  and then each  $y_1$  let me write  $y_1$  as see  $a_{11}, a_{12}$  to  $a_{1K}$  and like this. So, if I consider only the first row, if I consider if I consider this dictionary and these are the vectors these are the atoms and you know these are the this is the coefficients.

So,  $x_1$  vector and this is  $y_1$  vector, but which is represented by this  $K$  dimensional vector. So,

$$\vec{x}_1 = \sum_{i=1}^K a_{1i} \vec{d}_i = 1$$

So, in this way for all  $n$  vectors you have one column at  $y$ ; one column of Y. And, this is how these vectors are represented and this is how these linear combinations of each column vector of  $x$  is represented by these corresponding matrix multiplication of D and Y.

(Refer Slide Time: 10:10)

**Problem statement**

$$n \times N \longrightarrow X = [x_1 \ x_2 \ \dots \ x_N] \quad x_i \in \mathbb{R}^n$$

$$n \times K \longrightarrow D = [d_1 \ d_2 \ \dots \ d_K] \quad d_i \in \mathbb{R}^n$$

$$K \times N \longrightarrow Y = [y_1 \ y_2 \ \dots \ y_N] \quad y_i \in \mathbb{R}^K$$

- To obtain a sparse  $Y$  in  $\mathbb{R}^K$  such that
  - $X = DY$ , or  $X \sim DY$

Handwritten note:  $\vec{x}_1 \Rightarrow \sum_{i=1}^K a_i \vec{d}_i$

Various Sparsity constraints:

$\min_y \|y\|_0 \text{ subject to } x = Dy$

$\min_y \|y\|_0 \text{ subject to } \|x - Dy\|_2 \leq \epsilon$

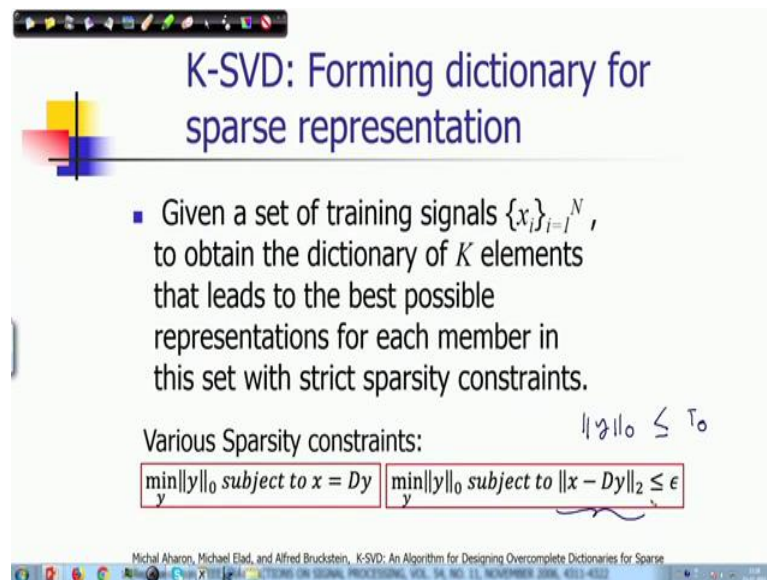
Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4322.

So, you can apply various sparsity constraints as I mentioned that the  $K$  dimensional representation of input vector which are given in the matrix  $Y$  now these would be sparse which means there should be only a few nonzero elements. So, when I have written a representation say  $x_1$  this vector when I am representing  $y$  say some of  $a_{1i} \vec{d}_1 = 1$  to  $k$  only few of this coefficient should be nonzero and rest of them should be 0.

So, you can apply various sparsity constraints there. So, one of the some of these examples of these constraints can be shown here say the 0th norm of  $y$ ; 0th norm means number of nonzero element number of nonzero element of  $y$  and that should be minimum. You should have that representation that your representation of dictionary atoms and also these sparse representation would be such that the number of zero nonzero elements of the each sparse represented vector that number should be kept as minimum.

So, if I consider the whole set  $y$  at the set  $y$  you can count how many number of nonzero elements are there in that matrix that can be also consider as we measured and we would like to minimize it. Or we can consider that the approximation; that means, it is for exact reconstruction  $X = DY$  we can consider this or it may be approximate reconstruction that it could be you may not get exact reconstruction of all the input vectors that could be a tolerance of Epsilon in the reconstruction in the sense of  $L_2$  now and there also it would like to minimize the 0th now of the  $Y$  vectors.

(Refer Slide Time: 12:30)



### K-SVD: Forming dictionary for sparse representation

- Given a set of training signals  $\{x_i\}_{i=1}^N$ , to obtain the dictionary of  $K$  elements that leads to the best possible representations for each member in this set with strict sparsity constraints.

Various Sparsity constraints:

$\min_y \|y\|_0 \text{ subject to } x = Dy$

$\min_y \|y\|_0 \text{ subject to } \|x - Dy\|_2 \leq \epsilon$

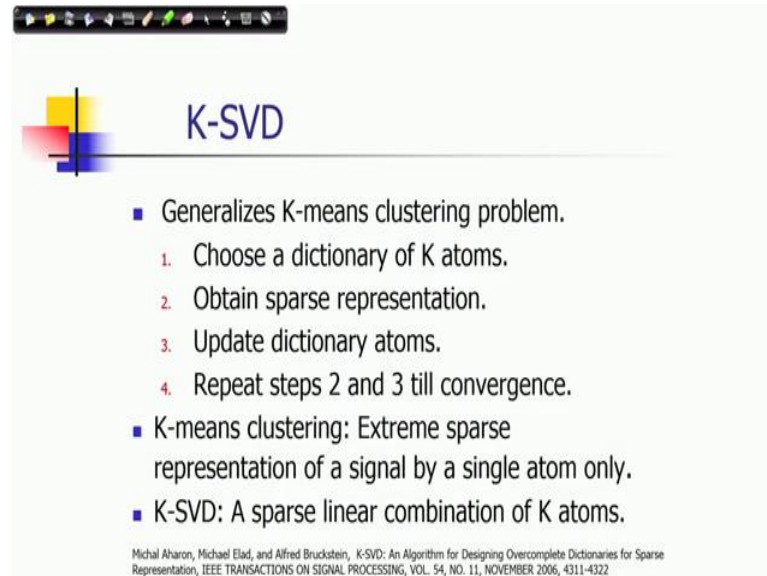
$\|y\|_0 \leq T_0$

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse

We will discuss if particular type of dictionary forming algorithm which is called K- SVD or K singular value decomposition and in this case it is a same problem statement as you can see it is the given a set of training signals to obtain the dictionary of K elements that leads to the best possible representation for each member in this set with strict sparsity constraints.

Now, the sparsity constraints here we have mention with this I would like to add in this case instead of considering always minimization we can make a bound that the 0th norm it should be less than or equals sum constant T0 which is a small number. And you would like to minimize the L 2 norm in this case that we will took by keeping this constraints we would like to minimize the L 2 norm of the reconstruction.

(Refer Slide Time: 13:44)

A presentation slide titled "K-SVD" with a logo on the left consisting of overlapping yellow, red, and blue squares. The slide contains a bulleted list of points. At the bottom, there is a small text citation.

**K-SVD**

- Generalizes K-means clustering problem.
  1. Choose a dictionary of K atoms.
  2. Obtain sparse representation.
  3. Update dictionary atoms.
  4. Repeat steps 2 and 3 till convergence.
- K-means clustering: Extreme sparse representation of a signal by a single atom only.
- K-SVD: A sparse linear combination of K atoms.

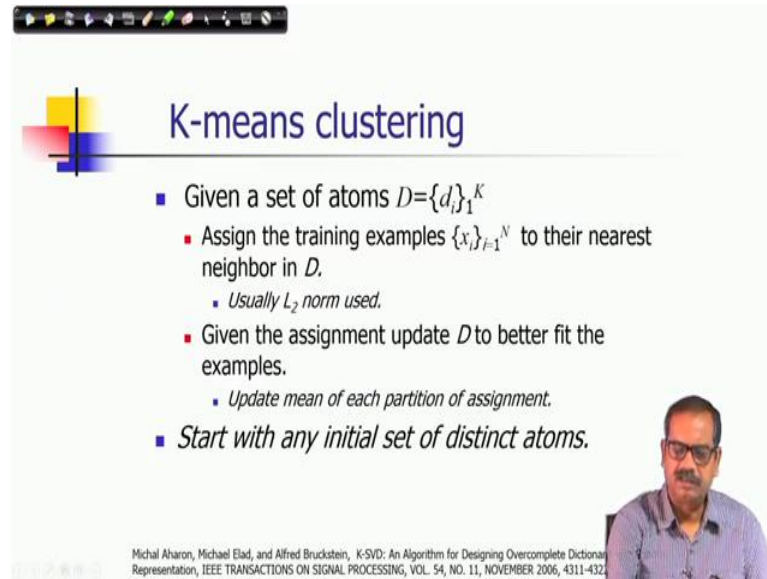
Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4322

So, the principal of this algorithm K similar value decomposition or K SVD that is how this algorithm is popular to popularly known, it generalizes the K means clustering algorithm or K means clustering problem rather. And, we know that in K means clustering what we get, we get representatives of K groups of data vectors and each representative acts like an atom in this context. And then you get an extreme sparse representation if I consider the any member of that group is represented by that atom only.

So, we will see that in that in terms of Y matrix how this vector looks when we consider its an extreme sparse representation. So, in the case of and in K SVD instead of only a single atom representation of K means clustering we consider a sparse linear combination of K atoms. So, what it does it chooses a dictionary of K atoms and then it obtains a sparse representation and then it updates dictionary atoms to get a better representation out of this and repeat steps two and three till convergence.

So, once it obtains a sparse representation using this atoms then it checks whether any sparse representation is possible by updating some of destruction atoms and also with better approximation of the input signals. And, if it is possible then it updates and it repeats its processes till there is no such improvement is possible.

(Refer Slide Time: 15:41)



**K-means clustering**

- Given a set of atoms  $D = \{d_j\}_{j=1}^K$ 
  - Assign the training examples  $\{x_i\}_{i=1}^N$  to their nearest neighbor in  $D$ .
    - Usually  $L_2$  norm used.
  - Given the assignment update  $D$  to better fit the examples.
    - Update mean of each partition of assignment.
  - Start with any initial set of distinct atoms.

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionary Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4322

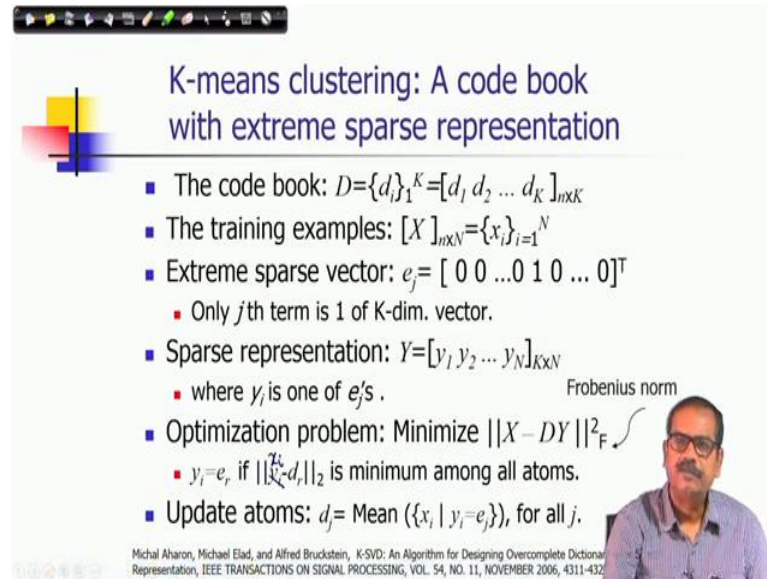
So, let us revisit this K means clustering algorithm and width in the context of this particular discussion. Say you have a set of atoms which from a dictionary you remember that we have to initialize the centers of clusters K centers of clusters now those initial centers are considered here as atoms of dictionary.

And then what we do that we assign the training examples to their nearest neighbor in the dictionary D. So, that is what we did in K means and then given an assignment then we update D to better fit the examples. So, far K means clustering we have used L 2 norm to check to compute the distance between any sample vector to the dictionary to the atoms or to the centre of clusters.

And then whichever is nearest we assign that vector input vector to that cluster and once the assignment is over then you have once again K partitions, then again re compute the centers. So, that is what the update is updation process see update mean of each partition of assignment. We started with any initial set of distinct atoms and then after convergence you consider those are the learn dictionary atoms.



(Refer Slide Time: 17:13)



**K-means clustering: A code book with extreme sparse representation**

- The code book:  $D = \{d_i\}_1^K = [d_1 \ d_2 \ \dots \ d_K]_{m \times K}$
- The training examples:  $[X]_{m \times N} = \{x_i\}_{i=1}^N$
- Extreme sparse vector:  $e_j = [0 \ 0 \ \dots 0 \ 1 \ 0 \ \dots 0]^T$ 
  - Only  $j$ th term is 1 of  $K$ -dim. vector.
- Sparse representation:  $Y = [y_1 \ y_2 \ \dots \ y_N]_{K \times N}$ 
  - where  $y_i$  is one of  $e_j$ 's.
- Optimization problem: Minimize  $\|X - DY\|_F^2$  (Frobenius norm)
  - $y_i = e_r$  if  $\|x_i - d_r\|_2$  is minimum among all atoms.
- Update atoms:  $d_j = \text{Mean}(\{x_i \mid y_i = e_j\})$ , for all  $j$ .

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionary Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4322

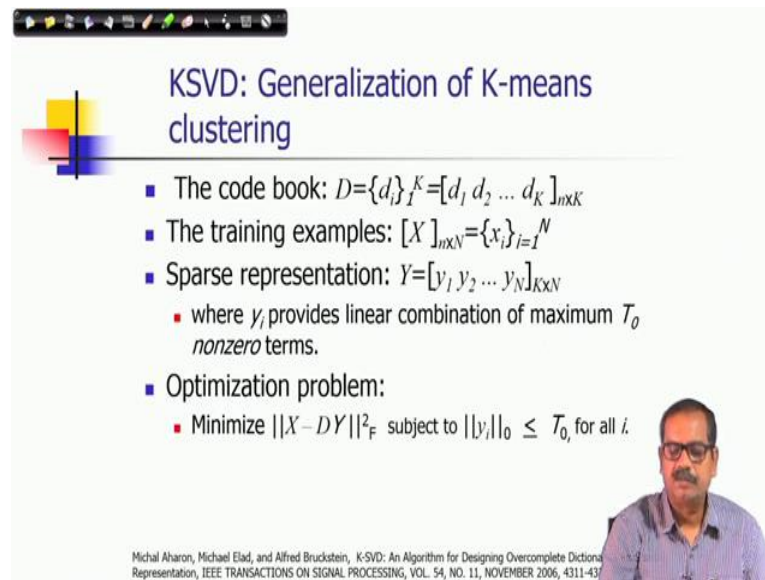
So, your codebook here is the  $K$  cluster centers those are the dictionary atoms that is your code book and you have this training examples and this is what is the representation exchange sparse represent sparse vector you can say  $j$  ith vector representation would be in this form all the elements would be 0 except 1. So, the 1 denotes that atom which is assigned to that input vector or this input vector belongs to that group where this atom is the centre of that cluster.

So, that is why it is called extreme sparse because it has only one nonzero element and rest of them at 0. So, in your sparse representation you have only this type of vectors; that means, extreme sparse vectors it should be one of them and your optimization problem as we have considered for  $K$  means. It is once again as you can see your minimizing the  $L_2$  norm square of the  $L_2$  and your update policies that if  $i$ th input vector is closest to the dictionary atom  $r$  th dictionary atom then assign the  $r$  th sparse  $r$  th you know, then the presentation should be here which is exchange sparse vector which means an nonzero elements 1 would be only at the  $r$  th location.

So, once you have computed that location then for all the vectors once the assignment is over, then you update each partition which means that you are considering mean of those input vectors which have the same assignment. For example, with the same assignment of  $e_j$  that is the sparse vector representation for the  $j$  th group and you compute its mean that would be your new dictionary updated dictionary atom.

So, this operation you are doing iteratively and at the when it converges then we are you stop and that is the dictionary you learned and already the sparse representations are also derived as extreme sparse vectors for each input vector. So, you have mentioned this is problem provenience norm that you should consider.

(Refer Slide Time: 19:46)



**KSVD: Generalization of K-means clustering**

- The code book:  $D = \{d_i\}_i^K = [d_1 \ d_2 \ \dots \ d_K]_{n \times K}$
- The training examples:  $[X]_{n \times N} = \{x_i\}_{i=1}^N$
- Sparse representation:  $Y = [y_1 \ y_2 \ \dots \ y_N]_{K \times N}$ 
  - where  $y_i$  provides linear combination of maximum  $T_0$  nonzero terms.
- Optimization problem:
  - Minimize  $\|X - DY\|_F^2$  subject to  $\|y_i\|_0 \leq T_0$ , for all  $i$ .

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionary Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4321

So, that was K means clustering and let us considered now that how this concept is generalized for K singular value decomposition approach or K SVD algorithm. In this case also we start with any initial codebook; we will discuss how this code book is derived here and suppose there is an initial course code book.

And then and also you have the training examples as given in this form a N number of data points and it can be represented as a data matrix X and you would like to get sparse representation once again a matrix of K X N each column represents the sparse representation of K data vector. And why I provide linear combination of maximum T naught nonzero term that I mentioned while discussing this sparsity constraints that is involved in KSVD algorithm.

So, the optimization problem here is that you would like to minimize the provenience norm  $X - DY$  its a generation of L2 norm square and subject to that number of nonzero elements for each sparse representation is less or equal to T 0.

(Refer Slide Time: 21:14)

Minimize  $\|X - DY\|_F^2$  subject to  $\|y_i\|_0 \leq T_0$  for all  $i$ .

### Rewriting optimization function

■  $y_T^j$ :  $j$ th row of  $Y$ .

$$\|X - DY\|_F^2 = \|X - \sum_{j=1}^K d_j y_T^j\|_F^2$$

$$DY = \sum_{j=1}^K d_j y_T^j$$

$$\begin{bmatrix} X \end{bmatrix}_{n \times N} \approx \sum_{j=1}^K \begin{bmatrix} d_j \end{bmatrix}_{1 \times K} \begin{bmatrix} y_T^j \end{bmatrix}_{K \times N} \Rightarrow \begin{bmatrix} \end{bmatrix}_{n \times N}$$

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse

So, this computation is carried out by a nice manipulation of the optimization function that we will discuss here. If I consider my optimization function if you note here the optimization function is given this; that means, Frobenius norm of  $X - DY$  and how this can be written in a different form that is that we would like to discuss now consider the  $j$ th row of  $Y$ . So, if you remember the representation of  $y$  each column is a sparse representation. So, I have represented by the coefficients  $a_{11}$ ,  $a_{12}$  to see  $a_{1k}$ .

So, this is the  $y_1$  of the sparse represented vector and this is representing  $x_1$ . So,  $y_1$  is represented representing  $x_1$ . Similarly  $y_2$  can be represented as linear combination say  $a_{21}$   $a_{22}$ , so this is my say notation  $a_{2k}$ . So, this is  $y_2$  and in this way you are representing say any particular small  $k$   $y$  as see  $a_{k1}$   $a_{k2}$  to  $a_{kK}$  is  $k$  another is  $K$ ;  $K$  is the dimension and finally, there are  $n$  number of representations because there are  $n$  number of data points.

Now, you consider any particular row say  $j$ th row which means this should be  $a_{1j}$   $a_{2j}$  a small  $k$   $j$  a capital  $N$   $j$ . So, this  $j$  is represented as the  $j$ th row vector. So, now, this expression  $X - DY$  this can be conveniently written in this form you can see that what is doing it is the sum of the rank 1 matrixes in the other way.

So, you have if I write it in this way that  $d_1$   $d_2$  to  $d_n$  and this is the  $y$  matrix. So, so we can write it as say  $d_1$  into this first row; that means,  $y_1^T$  in our notation. So,  $d_1$  this into this. So, the dimension of  $d_1$  is  $n \times 1$  and  $y_1^T$  its dimension is  $1 \times N$  because there are

N elements. So, there are N elements in a rho there are N columns. So, and only 1. So, this would give you a matrix of n cross N and x is a matrix which is of n cross N.

So, if I consider sum of all these matrixes that would actually cover the x. So, it is a same. So, this DY; DY is the same as this 1 it is not same it is trying to represent that we are trying to get that D and Y which will be which will be the same as a x if I multiply them, but  $DY = \sum_{j=1}^K d_j y_j^T$ .

So, this is the trick. So, you are considering every atom  $d_j$  and find out what is its contribution to the reconstruction of DY the whole multiplayer multiplied factor. So, the advantage here is that if you do that, then what you can consider you can separate out the contribution of a single atom say  $d_k$ .

(Refer Slide Time: 26:05)

Minimize  $\|X - DY\|_F^2$  subject to  $\|y_i\|_0 \leq T_0$  for all  $i$ .

### Rewriting optimization function

- $y_T^j$ :  $j$  th row of  $Y$ .

$$\|X - DY\|_F^2 = \|X - \sum_{j=1}^K d_j y_T^j\|_F^2$$

$$\left\| \left( X - \sum_{j \neq k} d_j y_T^j \right) - d_k y_T^k \right\|_F^2$$

$E_k$

Consider effect of minimizing w.r.t.  $k$  th row of  $Y$  associated with code vector  $d_k$  keeping other terms fixed. Perform SVD:  $E_k = UDV^T$  and take columns of  $U$  and  $V$  for max singular value (say  $D(1,1)$ ).

1<sup>st</sup> Column of  $U$ :  $d_k$   
 $D(1,1) \times$  1<sup>st</sup> column of  $V$ :  $y_T^k$

But the column vector may not be sparse.

Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4322.

So, this is the single atom  $d_k$  is the contribution of atom  $d_k$  and this is a contribution of rest of the atoms towards formation of DY and which we would like to get as close as possible to x. So, if we have this then as I then the idea is that since we are separated out. So, we can observe just what the effect of a single atom is.

So, if I keep all other thing constant then what is the best  $d_k$  and  $y_T^k$  that should approximate this particular factor let me call it as E. So, we can compute it because everything is given in your in your hand at this stage you have the input data vectors, you

have the set of atoms that is the safe suppose you are doing iteratively. So,  $K - 1$  th set of atoms you have also the sparse representation in the previous  $K - 1$  th sparse representation.

So, you can find out what is the value of this error of approximation using except  $K$  th atom this is an error approximation. So, the error can be this minus this. So, if I can update  $d_k$  and  $y^T_k$  such that it is close to  $e$  then my error would be minimized. So, that is a idea and that is what this algorithm does that keeping other transfixed. So, it gets  $E_k$ .

So, to do that what we can do? We can perform the SVD of  $E_k$ . So, this is I am mentioning as  $E_K$  and this is what is written. So, let me rub the my writing. So, if I perform SVD of  $E_K$  and then you know that with singular value decomposition you get set of or thermal vector column vectors of  $U$  and  $V$  and considered the columns of  $U$  the first column of  $u$  with the maximum singular value and column of  $b$  of the same maximum singular value.

So, you take the column of  $U$  and you can normalize it that is what is your  $d_k$  and the maximum singular value into the column of  $V$  we will give you the  $y_k^T$  that representation that could be 1 possible close approximation it is not exactly equal to  $E_K$ . It is a best approximation of rank 1 approximation from the theory of approximation of matrix using this kind of singular value decomposition.

Now, this is one best representation, problem here this that here you are not ensuring that  $y^T_k$  of that coefficients would be sparse, so this is a problem that. So, as I mentioned we can consider 1st column of  $U$  as  $d_k$  and the 1st column of  $V$  multiplied with the singular value  $d_1$ , 1 which is the maximum singlar value as  $y^T_k$ , but the column vector this column vector may not be sparse. So, what we should do in that case?

(Refer Slide Time: 29:28)

### K-SVD: Enforcing sparsity

- Choose only samples from  $X$  which have a nonzero component along  $d_k$ .
- Form reduced  $E_k$  (denoted  $E_{kR}$ ) and  $y_T^k$  by  $y_R^k$ .
- Perform SVD of  $E_{kR}$  to get  $d_k$  and  $y_R^k$ .
- Update  $d_k$  and  $y_T^k$ .
- Repeat for all  $d_j$ 's and obtain updated  $D$  and  $Y$ .
- Repeat till convergence

$$\left\| \left( X - \sum_{j \neq k} d_j y_T^j \right) - d_k y_T^k \right\|_F^2$$

$y_T^j$ :  $j$  th row of  $Y$ .

Performing SVD  $K$  times for  $K$  atoms in each iteration.

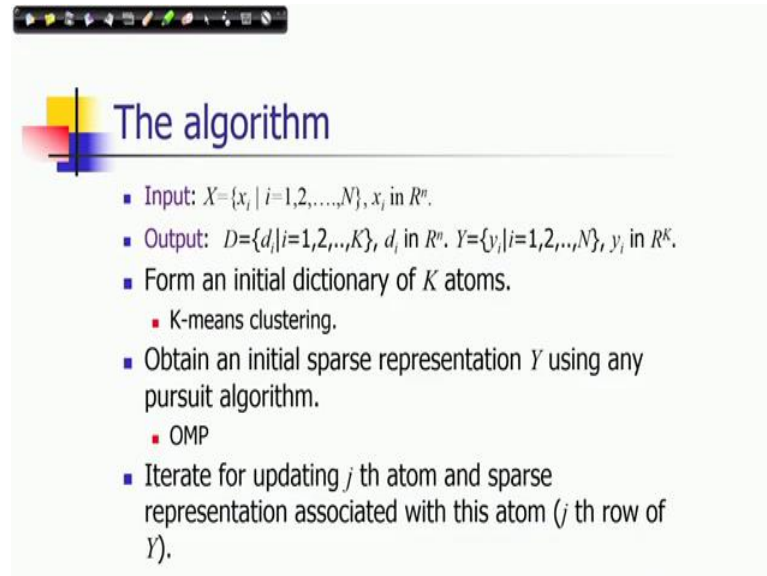
Michal Aharon, Michael Elad, and Alfred Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006, 4311-4322

$$\left\| \left( X - \sum_{j \neq k} d_j y_T^j \right) - d_k y_T^k \right\|_F^2$$

So, this is a tree that we can enforce sparsity by considering only those samples of the input which has nonzero contributions towards  $d_k$ ; that means, I have nonzero coefficient. So, choose only samples from  $x$  which have a nonzero component along  $d_k$  and from there you can reduce the error matrix  $E_k$  to  $E_{kR}$ . So, you are only considering those input samples related to that only and from there you can rewrite the equation. And, you can get  $E_{kR}$  you can get the reduced representation of the row  $y_{Rk}$  and that you can use the once again singular value decomposition of  $E_{kR}$  and then update  $d_k$  and  $y_{Tk}$  from there.

So, you can repeat for all  $d_j$ 's and update obtain the updated  $d$  and  $y$  in this form and you should repeat till repeat this thing till convergence. Actually convergence like  $K$  means clustering here also the problem here that it can get local it can start at a local optimum point in this case. So, performing SVD since you are performing this singular value decomposition  $K$  times for  $K$  atoms in each iteration we call this algorithm as  $K$  SVD.

(Refer Slide Time: 30:53)



### The algorithm

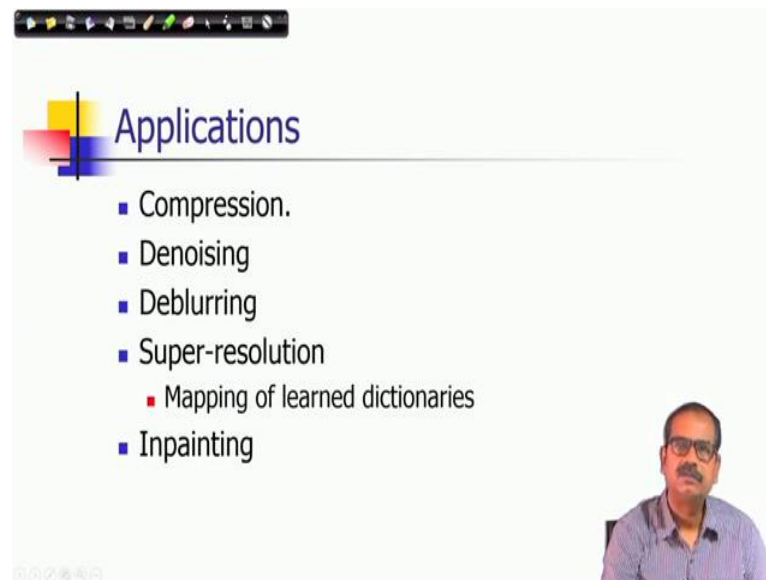
- Input:  $X = \{x_i \mid i=1,2,\dots,N\}$ ,  $x_i$  in  $R^n$ .
- Output:  $D = \{d_i \mid i=1,2,\dots,K\}$ ,  $d_i$  in  $R^n$ .  $Y = \{y_i \mid i=1,2,\dots,N\}$ ,  $y_i$  in  $R^K$ .
- Form an initial dictionary of  $K$  atoms.
  - K-means clustering.
- Obtain an initial sparse representation  $Y$  using any pursuit algorithm.
  - OMP
- Iterate for updating  $j$  th atom and sparse representation associated with this atom ( $j$  th row of  $Y$ ).

So, finally, if we like to give the overview of this algorithm from total perspective at your input is  $x$  data points sorry  $n$  capital  $N$  data points and represented by this set  $X$  or which can be also represented by a data matrix  $X$  and your output is your  $K$  number of atoms of the dictionary  $D$  and also the sparse representation of each input sample  $Y$ .

So, the first step is that you have to form an initial dictionary of  $K$  atoms and that any method you can use  $K$  means clustering can we also be used, then obtain an initial sparse representation  $y$  using any sparsity algorithm. So, you can use  $K$  means clustering for that or you can use orthogonal matching pursuit for this sparse representation, then you iterate for updating  $j$  th atom and sparse representation associated with this atom and you go on to in this iterations till there is a conventions.



(Refer Slide Time: 31:59)

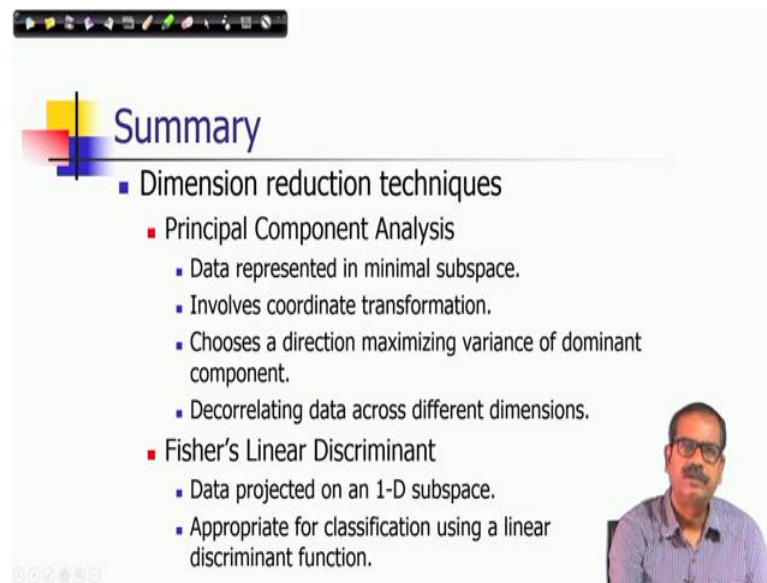


So, there are some applications of the K- SVD algorithms I mean it can be applied for data compression there are various image processing operations where denoising is applied. Actually the idea here is that since once again you have factorized the input data you only retain those factors which are important and where the coefficients are negligible then you can reject it. By doing this things you can compressed data, you can denoise data, you can deblur data. The other applications that kind of computation could be you can learn dictionaries of different levels of representation of a signal.

For example high level representation and low level representation and then if you establish the mapping between two dictionaries that it can give you an algorithm for increasing the resolution of signal. So, super resolution is that task which is called. So, mapping of learn dictionaries can be used in this case. In painting also another task of image processing where you take out some area and you feel that area by looking at the content of the image. So, that the discontinuities of extraction of the object from the image is not felt by everyone. So, they are also this kind of dictionary learning and mapping could be used.



(Refer Slide Time: 33:23)



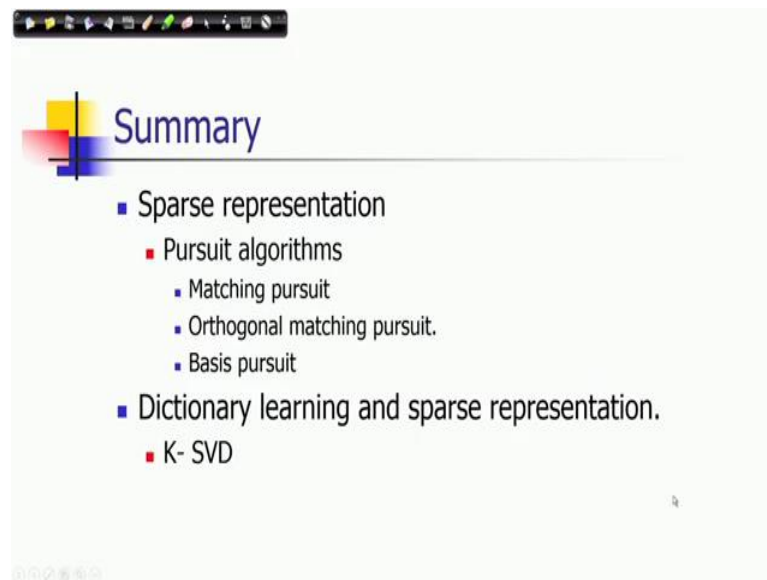
**Summary**

- Dimension reduction techniques
  - Principal Component Analysis
    - Data represented in minimal subspace.
    - Involves coordinate transformation.
    - Chooses a direction maximizing variance of dominant component.
    - Decorrelating data across different dimensions.
  - Fisher's Linear Discriminant
    - Data projected on an 1-D subspace.
    - Appropriate for classification using a linear discriminant function.

So, let me summarize the topics what we discussed under the title of dimension reduction and sparse representation. So, in dimension reduction we discussed about the technique of principal component analysis and also a technique Fisher's linear discriminant with respect to classification task.

So, far principal component analysis the objectives are to represent data in a minimal subspace and it involves we have seen it involves co ordinate transformation, it chooses a direction maximize in variance of dominance component and it decorrelates data across different dimensions. Whereas, in Fisher's linear discriminant the task was to data project the data in a 1-D subspace and then use it for classification and this projection itself gives severe linear discriminant function.

(Refer Slide Time: 34:19)



And, in sparse representation we have discussed about different pursuit algorithms like matching pursuit, then orthogonal matching pursuit, then basis pursuit we have given just a problem statement, but we did not discuss the algorithmic steps. And, then we also discussed technique for dictionary learning and using it how to derive sparse representation in particular we discussed about this algorithm K- SVD. So, with this let me stop and this is the end of this particular topic we will start a new topic in my next lecture.

Thank you very much.

Keywords: K-SVD, K-means algorithm, denoising, deblurring.