Machine Learning for Signal Processing
Sparse and Overcomplete Representations

Bhiksha Raj
(slides from Sourish Chaudhuri and Abelino Jimenez)

So far

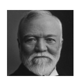
Can we use linear composition to identify **basic units** that compose the signal?

So far

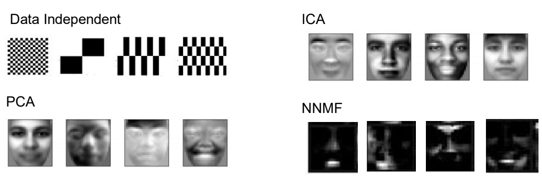
Weights

$$D \cdot \alpha \underset{\approx}{=} X$$

Data



Basis



basis <= dim X

Just in case you missed it..

- Remember, #(Basis Vectors)= #unknowns

$$D \cdot \alpha = X$$

Basis Vectors **Weights** **Input data**

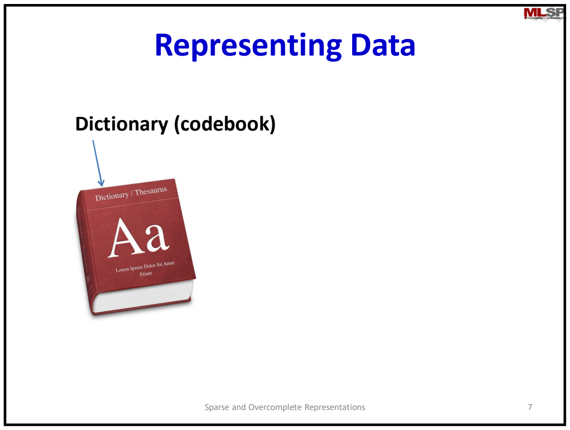
Standard representations: number of bases <= dimension of data

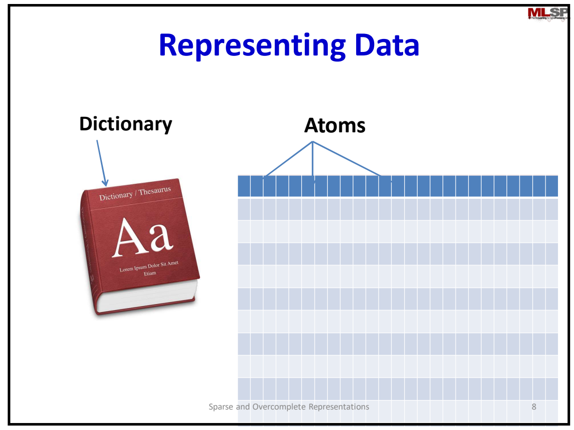
A limitation we saw earlier

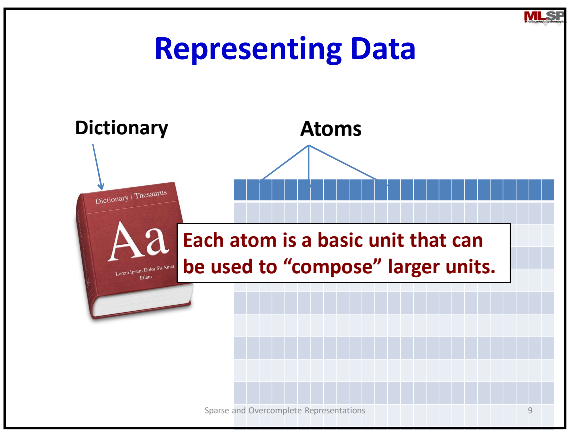
- Mathematical restrictions on the number of bases have no connection to reality
  - Universe does not respect your mathematical representations of the data
  - In reality: number of building blocks that compose any kind of data is unlimited
- Today: Learning linear compositional representations without restrictions on the number of basic units

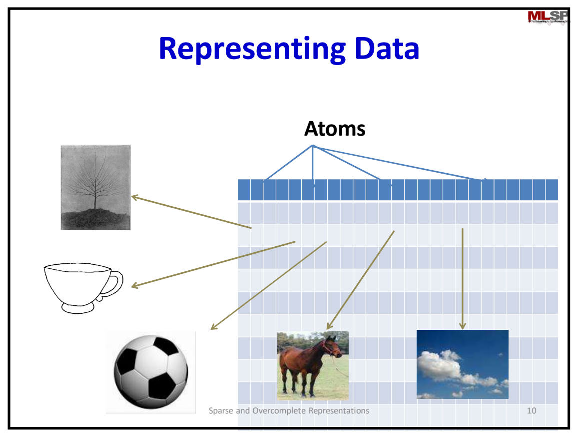
Key Topics in this Lecture

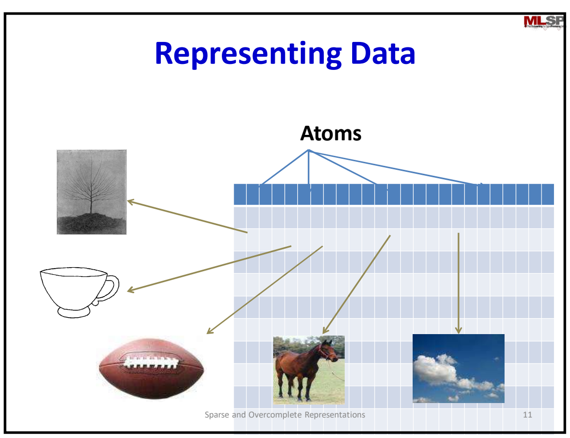
- Basics – Component-based representations
  - ***Overcomplete*** and Sparse Representations,
  - ***Dictionaries***
- Pursuit Algorithms
- How to learn a dictionary
- Why is an overcomplete representation powerful?

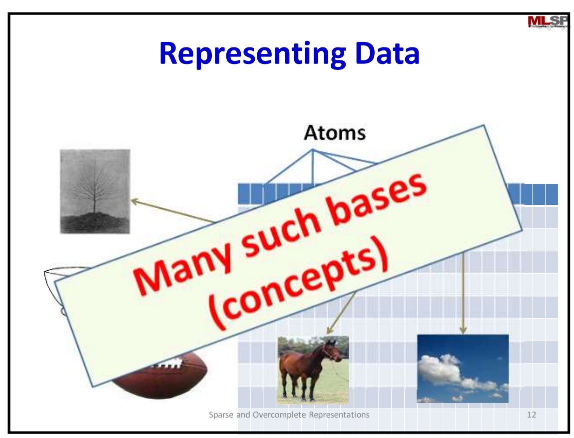

Representing Data
Dictionary (codebook)
Dictionary / Thesaurus
Aa
Sparse and Overcomplete Representations
7


Representing Data
Dictionary
Atoms
Dictionary / Thesaurus
Aa
Sparse and Overcomplete Representations
8

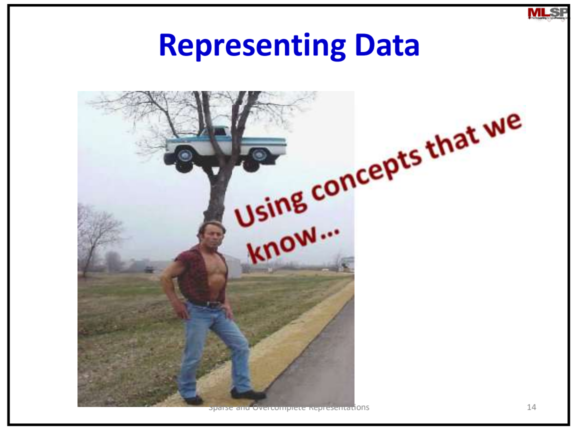

Representing Data
Dictionary
Atoms
Dictionary / Thesaurus
Aa
Each atom is a basic unit that can be used to “compose” larger units.
Sparse and Overcomplete Representations
9

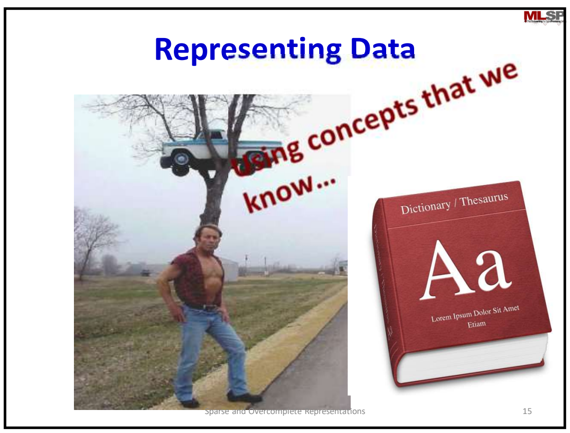

Representing Data
Atoms
Sparse and Overcomplete Representations
10

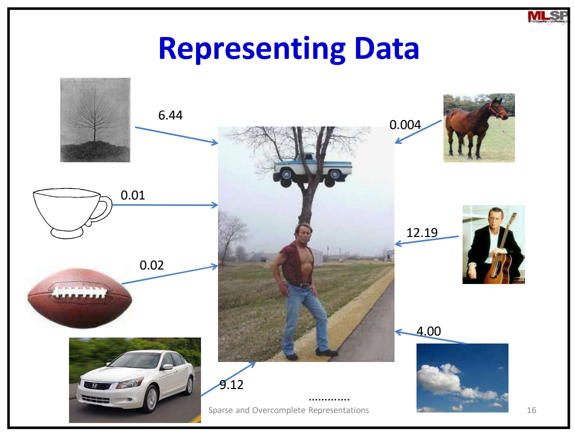

Representing Data
Atoms
Sparse and Overcomplete Representations
11

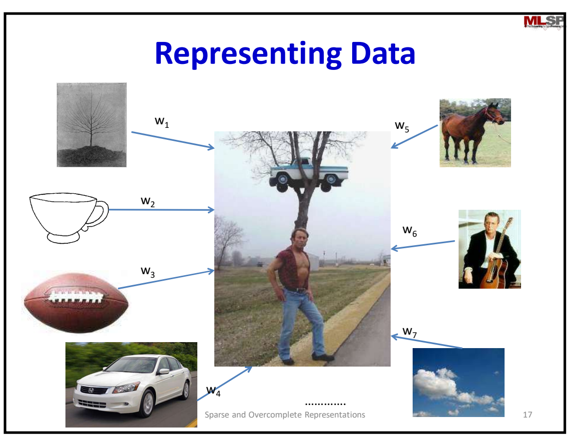

Representing Data
Atoms
Many such bases (concepts)
Sparse and Overcomplete Representations
12

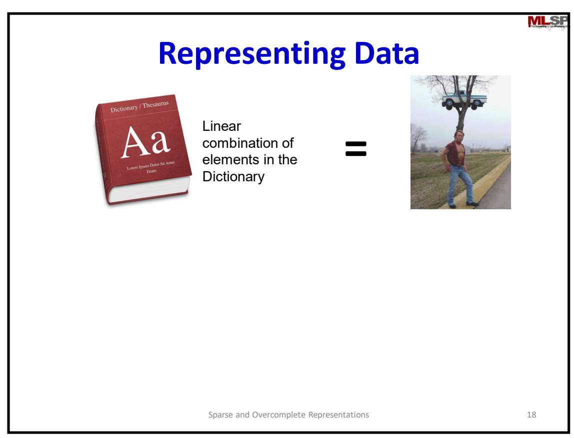

Representing Data
Sparse and Overcomplete Representations
13


Representing Data
Using concepts that we know…
Sparse and Overcomplete Representations
14

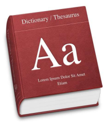

Representing Data
Using concepts that we know…
Dictionary / Thesaurus
Aa
Lorem Ipsum Dolor Sit Amet Etiam
Sparse and Overcomplete Representations
15


Representing Data
6.44
0.01
0.02
9.12
0.004
12.19
4.00
…………
Sparse and Overcomplete Representations
16

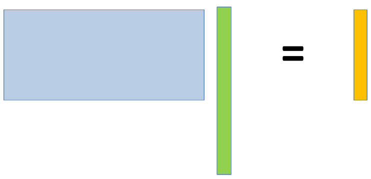

Representing Data
w_1
w_2
w_3
w_4
w_5
w_6
w_7
…………
Sparse and Overcomplete Representations
17


Representing Data
Aa
Linear combination of elements in the Dictionary
=
Sparse and Overcomplete Representations
18

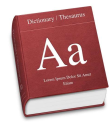
Representing Data



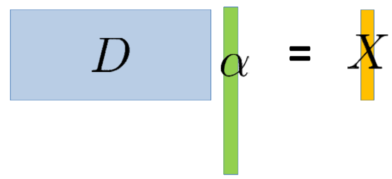
Linear combination of elements in the Dictionary = 



Representing Data



Linear combination of elements in the Dictionary = 



Quick Linear Algebra Refresher

- Remember, #(Basis Vectors)= #unknowns

$$D \cdot \alpha = X$$

Basis Vectors *(from Dictionary)*

Weights

Input data

Overcomplete Representations

- What is the dimensionality of the input image? (say 64x64 image)
  - **4096**
- What is the dimensionality of the dictionary? (each image = 64x64 pixels)
  - **4096 x N**

Overcomplete Representations

- What is the dimensionality of the input image? (say 64x64 image)
  - **4096**
- What is the dimensionality of the dictionary? (each image = 64x64 pixels)
  - **4096 x N** ???

Overcomplete Representations

- What is the dimensionality of the input image? (say 64x64 image)
  - **4096**
- What is the dimensionality of the dictionary? (each image = 64x64 pixels)
  - **4096 x N** VERY LARGE!!!

Overcomplete Representations

- What is the dimensionality of the input image? (say 64x64 image)

If N > 4096 (as it likely is)
we have an **overcomplete** representation

- What is the dimensionality of the dictionary? (each image = 64x64 pixels)

➢ **4096 x N** VERY LARGE!!!

Overcomplete Representations

- What is the dimensionality of the input image? (say 64x64 image)

More generally:

If #(dictionary units) > dimensions of input

we have an **overcomplete** representation

➢ **4096 x N** VERY LARGE!!!

Quick Linear Algebra Refresher

- Remember, #(Basis Vectors)= #unknowns

$$D \cdot \alpha = X$$

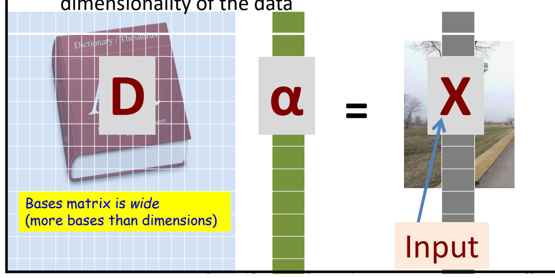
Dictionary Units

Weights

Input data

Dictionary based Representations

- Overcomplete "dictionary"-based representations are linear-composition-based representations with more "atomic building blocks" than the dimensionality of the data

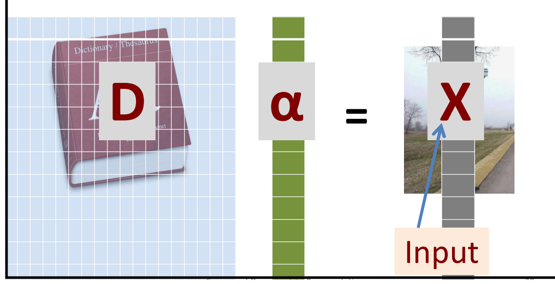


Why Dictionary-based Representations?

- Dictionary based representations are semantically more meaningful
- Enable content-based description
  - Bases can capture entire structures in data
  - E.g. notes in music
  - E.g. image structures (such as faces) in images
- Enable content-based processing
  - Reconstructing, separating, denoising, manipulating speech/music signals
  - Coding, compression, etc.
- Statistical reasons: We will get to that shortly..

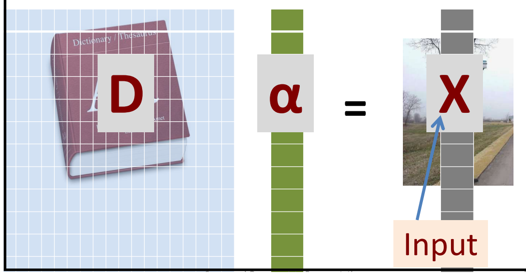
Problems

- How to obtain the dictionary
  - Which will give us meaningful representations
- How to compute the weights?



Problems

- How to obtain the dictionary
  - Which will give us meaningful representations
- How to compute the weights?



Quick Linear Algebra Refresher

- Remember, #(Basis Vectors)= #unknowns

$$D \cdot \alpha = X$$

Dictionary entries

Weights

Input data

When can we solve for α?

Quick Linear Algebra Refresher

$$D \cdot \alpha = X$$

D: full rank

D α = X Unique solution

D α = X We may have no exact solution

D α = X Infinite Solutions

Quick Linear Algebra Refresher

$$D \cdot \alpha = X$$

D: full rank

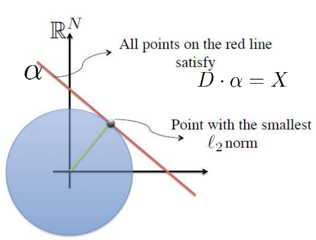
D α = X Unique solution

D α = X We may have no exact solution

Our Case

D α = X Infinite Solutions

Using Pseudo-Inverse?

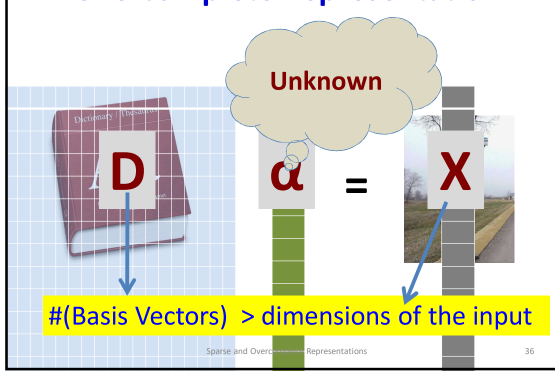


This is equivalent to

minimize $\|\alpha\|_2$ subject to $D\alpha = X$

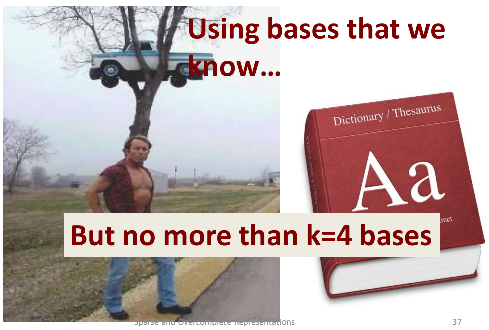
α will generally be "dense"

Overcomplete Representation



#(Basis Vectors) > dimensions of the input

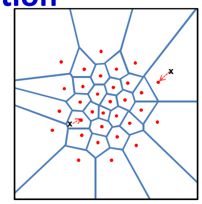
Representing Data



Alternate view: Recall quantization

$$V = \sum_i w_i d_i$$

$$V = \mathbf{Dw} \qquad |\mathbf{w}| = 1$$
$$|\mathbf{w}|_0 = 1$$



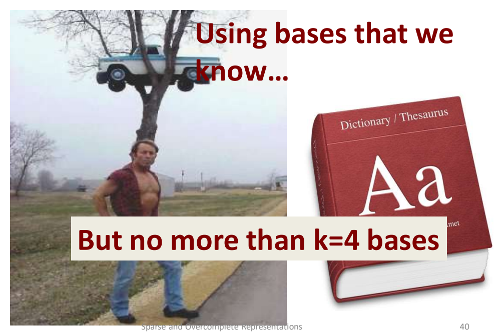
- d_i are the "representative" vectors of each cluster
- Restriction: only one of the w_i is 1, the rest are 0
  - $\sum_i w_i = 0$
  - $\mathbf{w}$ is unit length and one-sparse
- What if we let *more* than one entry of **w** to be non zero?

Overcompleteness and Sparsity

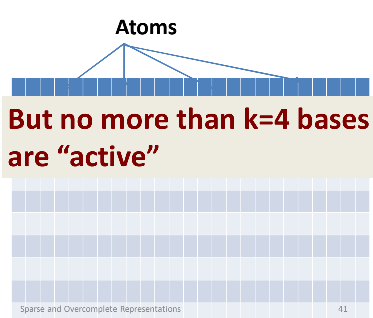
- To solve an overcomplete system of the type:

  D.α = X
- Make assumptions about the data.
- Suppose, we say that **X** is composed of no more than a fixed number (**k**) of "bases" from **D** (k ≤ dim(**X**))
  - The term "bases" is an abuse of terminology..
- Now, we can find the set of **k** bases that best fit the data point, **X**.

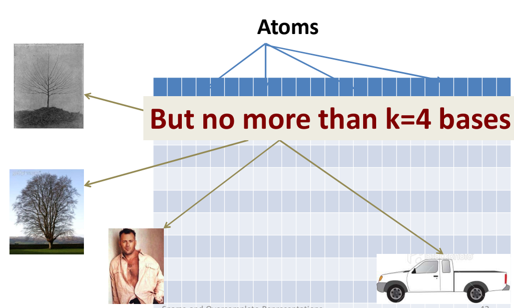
Representing Data



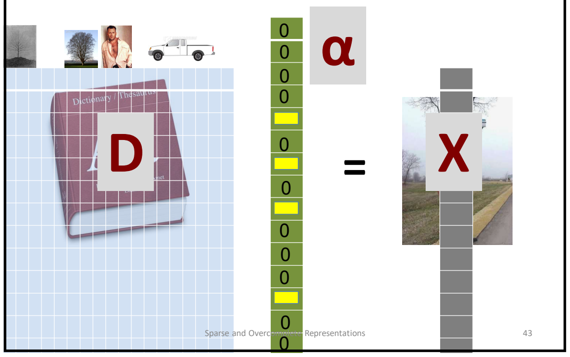
Overcompleteness and Sparsity



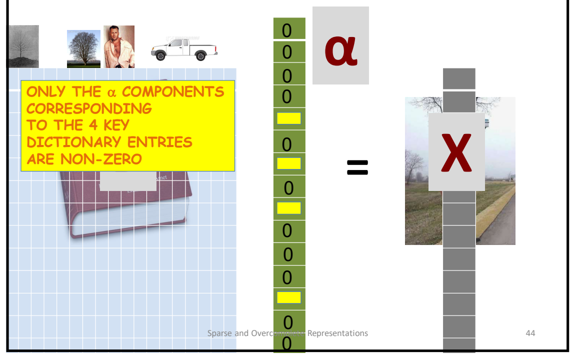
Overcompleteness and Sparsity



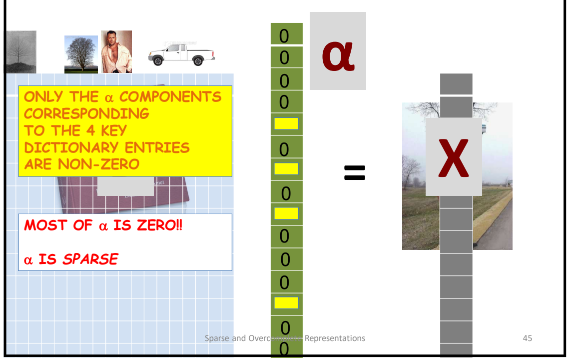
No more than 4 bases



No more than 4 bases



No more than 4 bases



Sparsity- Definition

- *Sparse representations* are representations that account for most or all information of a signal with a linear combination of a small number of atoms.

(*from*: www.see.ed.ac.uk/~tblumens/Sparse/Sparse.html)

The Sparsity Problem

- We don't really know **k**
- You are given a signal **X**
- Assuming **X** was generated using the dictionary, can we find α that generated it?

The Sparsity Problem

- We want to use as few dictionary entries as possible to do this.

$$\underset{\underline{\alpha}}{Min} \ \|\underline{\alpha}\|_0 \quad s.t. \ \ \underline{X} = \mathbf{D}\underline{\alpha}$$

The Sparsity Problem

- We want to use as few dictionary entries as possible to do this.

$$\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad s.t.\ \underline{X} = \mathbf{D}\underline{\alpha}$$

Counts the number of non-zero elements in α

The Sparsity Problem

- We want to use **as few dictionary entries** as possible to do this
  - Ockham's razor: Choose the simplest explanation invoking the fewest variables

$$\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad s.t.\ \underline{X} = \mathbf{D}\underline{\alpha}$$

The Sparsity Problem

- We want to use as few dictionary entries as possible to do this.

$$\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad s.t.\ \underline{X} = \mathbf{D}\underline{\alpha}$$

How can we solve the above?

Obtaining Sparse Solutions

- We will look at 2 algorithms:
  - Matching Pursuit (MP)
  - Basis Pursuit (BP)

Matching Pursuit (MP)

- Greedy algorithm
- Finds an atom in the dictionary that best matches the input signal
- Remove the weighted value of this atom from the signal
- Again, find an atom in the dictionary that best matches the remaining signal.
- Continue till a defined stop condition is satisfied.

Matching Pursuit

- Find the dictionary atom that best matches the given signal.

Weight = w_1

Matching Pursuit

- Remove weighted image to obtain updated signal

Find best match for this signal from the dictionary

Matching Pursuit

- Find best match for updated signal

Matching Pursuit

- Find best match for updated signal

Iterate till you reach a stopping condition, **norm(ResidualInputSignal) < threshold**

Matching Pursuit

Algorithm Matching Pursuit
Input: Signal: $f(t)$.
Output: List of coefficients: (a_n, g_{γ_n}).
Initialization:
  $Rf_1 \leftarrow f(t)$;
Repeat
  find $g_{\gamma_n} \in D$ with maximum inner product $< Rf_n, g_{\gamma_n} >$;
  $a_n \leftarrow < Rf_n, g_{\gamma_n} >$;
  $Rf_{n+1} \leftarrow Rf_n - a_n g_{\gamma_n}$;
  $n \leftarrow n+1$;
Until stop condition (for example: $\|Rf_n\| < threshold$)

From http://en.wikipedia.org/wiki/Matching_pursuit

Matching Pursuit

- Problems ???

Matching Pursuit

- Main Problem
  - Computational complexity
  - The entire dictionary has to be searched at every iteration

Comparing MP and BP

Matching Pursuit	Basis Pursuit
Hard thresholding (remember the equations)	
Greedy optimization at each step	
Weights obtained using greedy rules	

Basis Pursuit (BP)

- Remember,

$$\boxed{\begin{array}{l} \underset{\alpha}{Min} \ \|\alpha\|_0 \\ s.t. \ \ X = \mathbf{D}\alpha \end{array}}$$

Basis Pursuit

- Remember,

$$\boxed{\begin{array}{l} \underset{\alpha}{Min} \ \|\alpha\|_0 \\ s.t. \ \ X = \mathbf{D}\alpha \end{array}}$$

In the general case, this is intractable

Basis Pursuit

- Remember,

$$\boxed{\begin{array}{l} \underset{\alpha}{Min} \ \|\alpha\|_0 \\ s.t. \ \ X = \mathbf{D}\alpha \end{array}}$$

In the general case, this is intractable

Requires combinatorial optimization

Basis Pursuit

- Replace the intractable expression by an expression that is solvable

$$\boxed{\begin{array}{l} \underset{\alpha}{Min} \ \|\alpha\|_1 \\ s.t. \ \ X = \mathbf{D}\alpha \end{array}}$$

Basis Pursuit

- Replace the intractable expression by an expression that is solvable

$$\boxed{\begin{array}{l} \underset{\alpha}{Min} \ \|\alpha\|_1 \\ s.t. \ \ X = \mathbf{D}\alpha \end{array}}$$

This will provide identical solutions when $\mathbf{D}$ obeys the ***Restricted Isometry Property***.

Basis Pursuit

- Replace the intractable expression by an expression that is solvable

$$\underset{\alpha}{Min}\ \|\underline{\alpha}\|_1 \quad \text{Objective}$$
$$s.t.\ \ \underline{X} = \mathbf{D}\underline{\alpha}$$

Constraint

Basis Pursuit

- We can formulate the optimization term as:

$$\underset{\alpha}{Min}\ \{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

Constraint Objective

Basis Pursuit

- We can formulate the optimization term as:

$$\underset{\alpha}{Min}\ \{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

λ is a penalty term on the non-zero elements and promotes sparsity

Basis Pursuit

Equivalent to *LASSO*; for more details, see [this paper by Tibshirani](http://www-stat.stanford.edu/~tibs/ftp/lasso.ps)
http://www-stat.stanford.edu/~tibs/ftp/lasso.ps

$$\underset{\alpha}{Min}\ \{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

λ is a penalty term on the non-zero elements and promotes sparsity

Basis Pursuit

$$\underset{\alpha}{Min}\{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

$$\frac{\partial\|\alpha\|_1}{\partial\alpha_j} = \begin{cases} +1 & \text{at } \alpha_j > 0 \\ [-1,1] & \text{at } \alpha_j = 0 \\ -1 & \text{at } \alpha_j < 0 \end{cases}$$

- $\|\alpha\|_1$ is not differentiable at $\alpha_j = 0$
- Gradient of $\|\alpha\|_1$ for gradient descent update
- At optimum, following conditions hold

$$\nabla_j\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda sign(\alpha_j) = 0, \quad \text{if } |\alpha_j| > 0$$
$$\nabla_j\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 \le \lambda, \quad \text{if } \alpha_j = 0$$

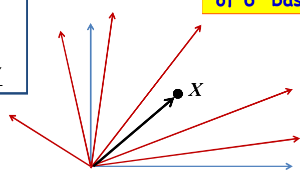
Basis Pursuit

- There are efficient ways to solve the LASSO formulation.
  - http://web.stanford.edu/~hastie/glmnet_matlab/
- Simplest solution: Coordinate descent algorithms
  - On webpage..

L_1 vs L_0

$$\underset{\alpha}{Min} \|\alpha\|_0$$
$$s.t.\ X = \mathbf{D}\alpha$$

Overcomplete set of 6 "bases"

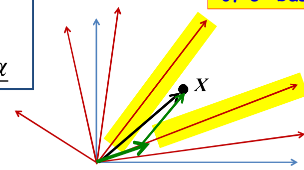


- **L_0 minimization**
  - Two-sparse solution
  - ANY pair of bases can explain X with 0 error

L_1 vs L_0

$$\underset{\alpha}{Min} \|\alpha\|_1$$
$$s.t.\ X = \mathbf{D}\alpha$$

Overcomplete set of 6 "bases"



- **L_1 minimization**
  - Two-sparse solution
  - All else being equal, the two closest bases are chosen

Comparing MP and BP

Matching Pursuit	Basis Pursuit
Hard thresholding	Soft thresholding
(remember the equations)	
Greedy optimization at each step	Global optimization
Weights obtained using greedy rules	Can force N-sparsity with appropriately chosen weights

General Formalisms

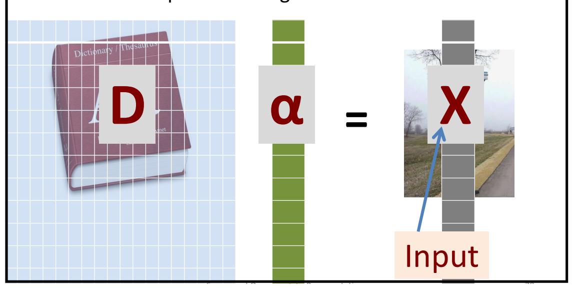
- L_0 minimization $\quad \underset{\alpha}{Min} \|\alpha\|_0 \quad s.t.\ X = \mathbf{D}\alpha$
- L_0 constrained optimization $\quad \underset{\alpha}{Min} \|X - \mathbf{D}\alpha\|_2^2 \quad s.t.\ \|\alpha\|_0 < C$
- L_1 minimization $\quad \underset{\alpha}{Min} \|\alpha\|_1 \quad s.t.\ X = \mathbf{D}\alpha$
- L_1 constrained optimization $\quad \underset{\alpha}{Min} \|X - \mathbf{D}\alpha\|_2^2 \quad s.t.\ \|\alpha\|_1 < C$

Many Other Methods..

- Iterative Hard Thresholding (IHT)
- CoSAMP
- OMP
- …

Problems

- How to obtain the dictionary
  - Which will give us meaningful representations
- How to compute the weights?

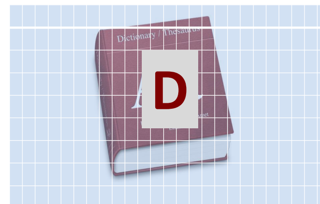


Trivial Solution

- D = Training data
- Impractical in most situations
  - Popular approach: sample random vectors from training data

Dictionaries: Compressive Sensing

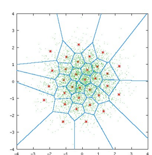
- Just random vectors!



More Structured ways of Constructing Dictionaries

- Dictionary entries must be structurally "meaningful"
  - Represent true compositional units of data
- Have already encountered two ways of building dictionaries
  - NMF for non-negative data
  - K-means ..

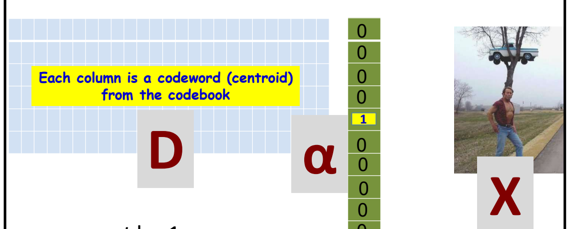
K-Means for Composing Dictionaries



Train the codebook from training data using K-means

- Every vector is approximated by the centroid of the cluster it falls into
- Cluster means are "codebook" entries
  - Dictionary entries
  - Also compositional units the compose the data

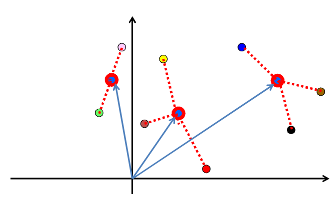
K-Means for Dictionaries



- α *must* be 1 sparse
- Only α entry must 1

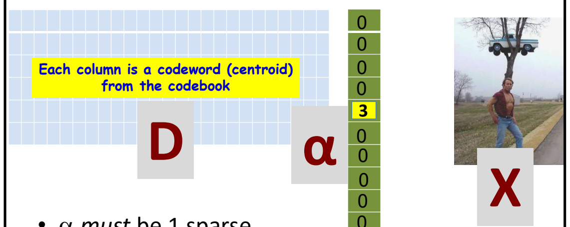
$|| \propto ||_0 = \mathbf{1}$ $|| \propto ||_1 = \mathbf{1}$

K-Means



- Learn Codewords to minimize the total squared length of the training vectors from the closest codeword

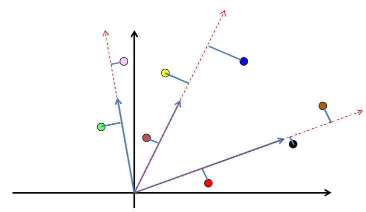
Length-unconstrained K-Means for Dictionaries



- α *must* be 1 sparse
- No restriction on α value

$||\propto||_0 = 1$

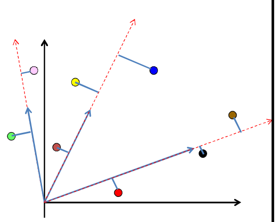
SVD K-Means



- Learn Codewords to minimize the total squared *projection error* of the training vectors from the closest codeword

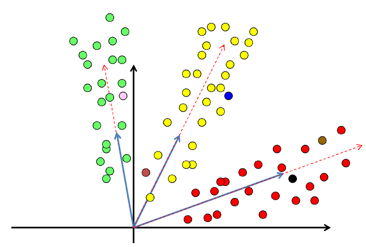
SVD K–means

1. Initialize a set of centroids randomly
2. For each data point x, find the projection from the centroid for each cluster
   - $p_{cluster} = \left|x^T m_{cluster}\right|$
3. Put data point in the cluster of the closest centroid
   - Cluster for which $p_{cluster}$ is *maximum*
4. When all data points are clustered, recompute centroids



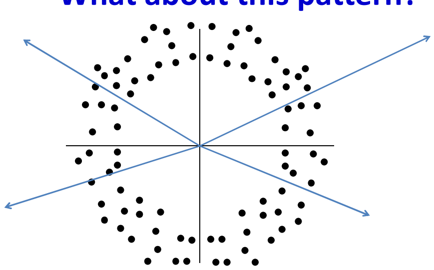
$$m_{cluster} = \text{Principal Eigenvector}(\{x \mid x \in cluster\})$$

Problem



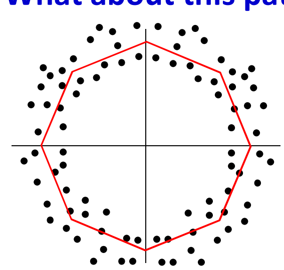
- Only represents *Radial* patterns

What about this pattern?



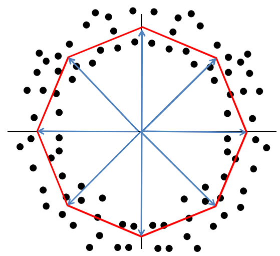
- Dictionary entries that represent radial patterns will not capture this structure
  - 1-sparse representations will not do

What about this pattern?



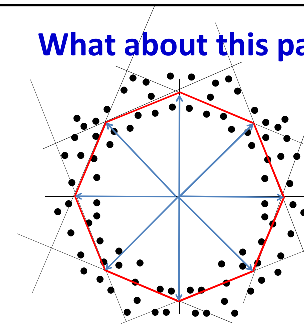
- We need ***AFFINE*** *patterns*

What about this pattern?



- We need ***AFFINE*** *patterns*
- *Each vector is modeled by a linear combination of K (here 2) bases*

What about this pattern?



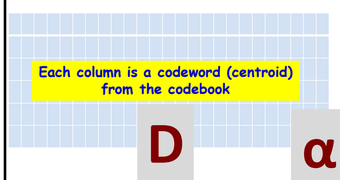
Every line is a (constrined) combination of two bases

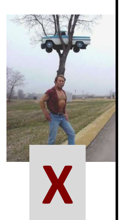
2-sparse

Constraint:
Line = a.b$_1$ + (1-a)b$_2$

- We need ***AFFINE*** *patterns*
- *Each vector is modeled by a linear combination of K (here 2) bases*

Codebooks for K sparsity?





- α *must* be *k* sparse
- No restriction on α value

$$|| \propto ||_0 = \boldsymbol{k}$$

Formalizing

Given training data

$$\{X_1, X_2, ..., X_T\}$$

We want to find a dictionary D, such that

$$D\alpha_i = X_i$$

With α_i sparse

Formalizing

Two objectives:

- Approximation $\|D\alpha_i - X_i\|$
- Sparsity in coefficients $\|\alpha_i\|_1$

$$\min_{D,\alpha_i} \sum_{i=1}^{T} \|X_i - D\alpha_i\|^2 + \|\alpha_i\|_1$$

NON-Convex!!!

An iterative method

- Given D, estimate α_i to get sparse solution
  - We can use any method

$$\min_{\alpha_i} \sum_{i=1}^{T} \|X_i - D\alpha_i\|^2 + \|\alpha_i\|_1$$

- Given α_i, estimate D

$$\min_{D} \sum_{i=1}^{T} \|X_i - D\alpha_i\|^2$$

Difficult!

K SVD

- Initialize Codebook

D =

1. For every vector, compute K-sparse alphas
   – Using any pursuit algorithm

α =

0	0	2	0	0
1	0	0	0	0
0	0	0	1	0
0	7	0	0	0
0	0	0	1	0
3	0	0	0	0
0	5	0	0	1
0	0	1	0	2

K-SVD

2. For each codeword (k):
- For each vector $\boldsymbol{x}$
  - Subtract the contribution of all other codewords to obtain $\mathbf{e}_k(\boldsymbol{x})$
    - Codeword-specific residual
- Compute the principal Eigen vector of $\{\mathbf{e}_k(\boldsymbol{x})\}$

3. Return to step 1

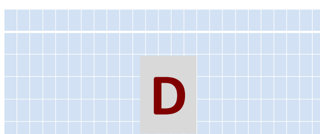
D = ($\mathrm{D_j}$ j != 1)

α = ($\alpha_j(x)$ j != 1)

0	0	2	0	0
1	0	0	0	0
0	0	0	1	0
0	7	0	0	0
0	0	0	1	0
3	0	0	0	0
0	5	0	0	1
0	0	1	0	2

$e_k(x) = x - \sum_{j \neq k} \propto_j D_j$

K-SVD



- Termination of each iteration: Updated dictionary
- Conclusion: A dictionary where any data vector can be composed of at most K dictionary entries
  – More generally, sparse composition

K-SVD algorithm (skip)

Initialization : Set the random normalized dictionary matrix $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times K}$. Set $J = 1$.
Repeat until convergence,
Sparse Coding Stage: Use any pursuit algorithm to compute $\mathbf{x}_i$ for $i = 1, 2, \ldots, N$

$$\min_{\mathbf{x}} \{\|\mathbf{y}_i - \mathbf{D}\mathbf{x}\|_2^2\} \quad \text{subject to} \quad \|\mathbf{x}\|_0 \le T_0.$$

Codebook Update Stage: For $k = 1, 2, \ldots, K$

- Define the group of examples that use $\mathbf{d}_k$, $\omega_k = \{i | \, 1 \le i \le N, \, \mathbf{x}_i(k) \neq 0\}$.
- Compute

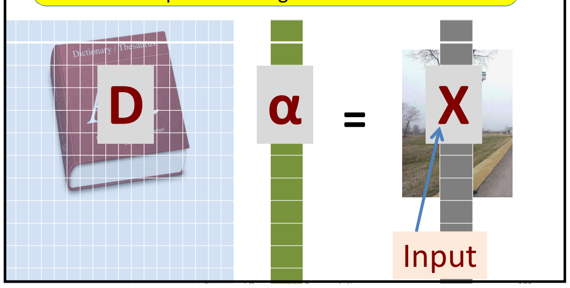
$$\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}^j,$$

- Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to those elements that initially used $\mathbf{d}_k$ in their representation, and obtain $\mathbf{E}_k^R$.
- Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T$. Update: $\mathbf{d}_k = \mathbf{u}_1, \mathbf{x}_R^k = \boldsymbol{\Delta}(1,1) \cdot \mathbf{v}_1$

Set $J = J + 1$.

Problems

- How to obtain the dictionary
  – Which will give us meaningful representations
- How to compute the weights?



Applications of Sparse Representations

- Many many applications
  – Signal representation
  – Statistical modelling
  – ..
  – We've seen one: Compressive sensing
- Another popular use
  – **Denoising**

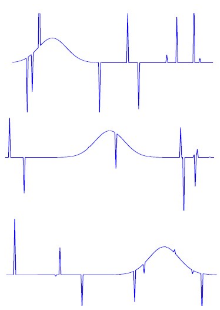
Denoising

- As the name suggests, remove noise!

Denoising

- As the name suggests, remove noise!
- We will look at image denoising as an example

A toy example



A toy example

$$D = [I \; G]$$

I Identity matrix

G Translation of a Gaussian pulse

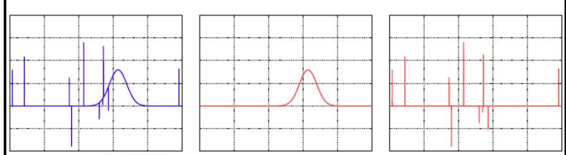


Image Denoising

- Here's what we want

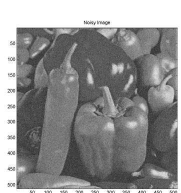


Image Denoising

- Here's what we want

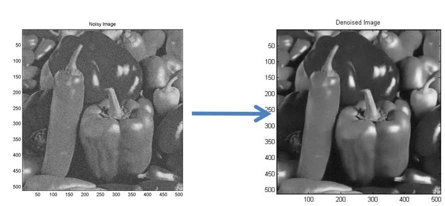
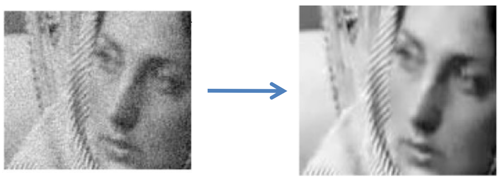


Image Denoising

- Here's what we want



The Image Denoising Problem

- Given an image
- Remove Gaussian additive noise from it

Image Denoising

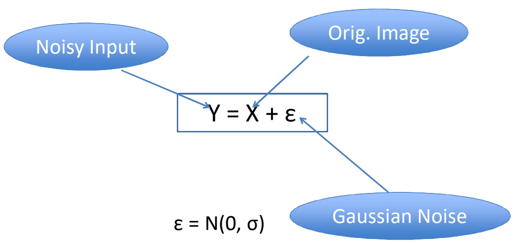


Image Denoising

- Remove the noise from **Y**, to obtain **X** as best as possible.

Image Denoising

- Remove the noise from **Y**, to obtain **X** as best as possible
- Using sparse representations over learned dictionaries

Image Denoising

- Remove the noise from **Y**, to obtain **X** as best as possible
- Using sparse representations over learned dictionaries
- We will *learn* the dictionaries

Image Denoising

- Remove the noise from **Y**, to obtain **X** as best as possible
- Using sparse representations over learned dictionaries
- We will *learn* the dictionaries
- What data will we use? *The corrupted image itself!*

Image Denoising

- We use the data to be denoised to learn the dictionary.
- Training and denoising become an iterated process.
- We use image patches of size √n x √n pixels (i.e. if the image is 64x64, patches are 8x8)

Image Denoising

- The data dictionary D
  - Size = n x k (k > n)
  - This is known and fixed, to start with
  - Every image patch can be sparsely represented using D

Image Denoising

- Recall our equations from before.
- We want to find α so as to minimize the value of the equation below:

$$\underset{\underline{\alpha}}{Min}\ \{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_0\}$$

$$\underset{\underline{\alpha}}{Min}\{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

Image Denoising

$$\underset{\underline{\alpha}}{Min}\ \{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

- In the above, *X* is a patch.

Image Denoising

$$\underset{\underline{\alpha}}{Min}\ \{\|\underline{X} - \mathbf{D}\underline{\alpha}\|^2 + \lambda\|\underline{\alpha}\|_1\}$$

- In the above, *X* is a patch.
- If the larger image is fully expressed by the every patch in it, how can we go from patches to the image?

Image Denoising

$$\underset{\alpha_{ij},X}{Min}\ \{\mu\|X-Y\|_2^2+\sum_{ij}\|R_{ij}X-\mathbf{D}\alpha_{ij}\|_2^2$$
$$+\sum_{ij}\lambda_{ij}\|\alpha_{ij}\|_0\}$$

Image Denoising

$$\underset{\alpha_{ij},X}{Min}\ \{\mu\|X-Y\|_2^2+\sum_{ij}\|R_{ij}X-\mathbf{D}\alpha_{ij}\|_2^2$$
$$+\sum_{ij}\lambda_{ij}\|\alpha_{ij}\|_0\}$$

(X – Y) is the error between the input and denoised image. μ is a penalty on the error.

Image Denoising

$$\underset{\alpha_{ij},X}{Min}\ \{\mu\|X-Y\|_2^2+\sum_{ij}\|R_{ij}X-\mathbf{D}\alpha_{ij}\|_2^2$$
$$+\sum_{ij}\lambda_{ij}\|\alpha_{ij}\|_0\}$$

Error bounding in each patch
- R_{ij} selects the $(ij)^{th}$ patch
- Terms in summation = no. of patches

Image Denoising

$$\underset{\alpha_{ij},X}{Min}\ \{\mu\|X-Y\|_2^2+\sum_{ij}\|R_{ij}X-\mathbf{D}\alpha_{ij}\|_2^2$$
$$+\sum_{ij}\lambda_{ij}\|\alpha_{ij}\|_0\}$$

λ forces sparsity

Image Denoising

- But, we don't "***know***" our dictionary D.
- We want to estimate D as well.

Image Denoising

- But, we don't "***know***" our dictionary D.
- We want to estimate D as well.

$$\underset{D,\alpha_{ij},X}{Min}\ \{\mu\|X-Y\|_2^2+\sum_{ij}\|R_{ij}X-\mathbf{D}\alpha_{ij}\|_2^2$$
$$+\sum_{ij}\lambda_{ij}\|\alpha_{ij}\|_0\}$$

We can use the previous equation itself!!!

Image Denoising

$$\underset{D,\alpha_{ij},X}{\underline{Min}}\ \{\mu\|\underline{X}-Y\|_2^2+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2 + \sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

How do we estimate all 3 at once?

Image Denoising

$$\underset{D,\alpha_{ij},X}{\underline{Min}}\ \{\mu\|\underline{X}-Y\|_2^2+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2 + \sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

How do we estimate all 3 at once?

We cannot estimate them at the same time!

Image Denoising

$$\underset{D,\alpha_{ij},X}{\underline{Min}}\ \{\mu\|\underline{X}-Y\|_2^2+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2 + \sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

How do we estimate all 3 at once?

Fix 2, and find the optimal 3rd.

Image Denoising

$$\underset{D,\alpha_{ij},X}{\underline{Min}}\ \{\mu\|\underline{X}-Y\|_2^2+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2 + \sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

Initialize X = Y

Image Denoising

$$\underset{\alpha_{ij}}{\underline{Min}}\ \{\underbrace{\mu\|\underline{X}-Y\|_2^2}_{0}+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2 + \sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

Initialize X = Y, initialize D

You know how to solve the remaining portion for α – MP, BP!

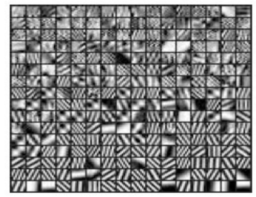
Image Denoising

- Now, update the dictionary D.
- Update D one column at a time, following the K-SVD algorithm
- K-SVD maintains the sparsity structure

Image Denoising

- Now, update the dictionary D.
- Update D one column at a time, following the [K-SVD algorithm]
- K-SVD maintains the sparsity structure
- Iteratively update α and D

Image Denoising



Learned Dictionary for Face Image denoising

From: M. Elad and M. Aharon, *Image denoising via learned dictionaries and sparse representation*, CVPR, 2006.

Image Denoising

$$\underset{\underline{X}}{Min}\ \{\mu\|\underline{X}-Y\|_2^2+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2$$

$$+\sum_{ij}\lambda_{ij}\|\underline{\alpha}_{ij}\|_0\} \longrightarrow \text{Const. wrt X}$$

We know D and α

The quadratic term above has a closed-form solution

Image Denoising

$$\underset{\underline{X}}{Min}\ \{\mu\|\underline{X}-Y\|_2^2+\sum_{ij}\|\underline{R_{ij}X}-\mathbf{D}\underline{\alpha}_{ij}\|_2^2$$

$$+\sum_{ij}\lambda_{ij}\|\underline{\alpha}_{ij}\|_0\} \longrightarrow \text{Const. wrt X}$$

We know D and α

$$X=(\mu I+\sum_{ij}R_{ij}^T R_{ij})^{-1}(\mu Y+\sum_{ij}R_{ij}^T D\alpha_{ij})$$

Image Denoising

- Summarizing… We wanted to obtain 3 things

Image Denoising

- Summarizing… We wanted to obtain 3 things

➢ Weights **α**
➢ Dictionary **D**
➢ Denoised Image **X**

Image Denoising

- Summarizing… We wanted to obtain 3 things

➢ Weights **α** – Your favorite pursuit algorithm
➢ Dictionary **D** – Using K-SVD
➢ Denoised Image **X**

Image Denoising

- Summarizing… We wanted to obtain 3 things

➢ Weights **α** – Your favorite pursuit algorithm
➢ Dictionary **D** – Using K-SVD
➢ Denoised Image **X**

Iterating

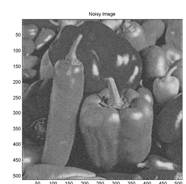
Image Denoising

- Summarizing… We wanted to obtain 3 things

➢ Weights **α**
➢ Dictionary **D**
➢ Denoised Image **X**- Closed form solution

Image Denoising

- Here's what we want



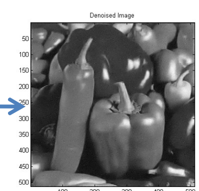


Image Denoising

- Here's what we want

Comparing to Other Techniques

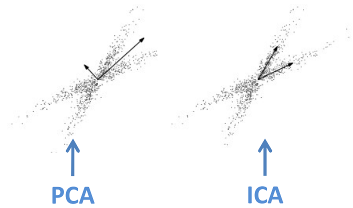
Non-Gaussian data

PCA of ICA Which is which?

Images from Lewicki and Sejnowski, *Learning Overcomplete Representations*, 2000.

Comparing to Other Techniques

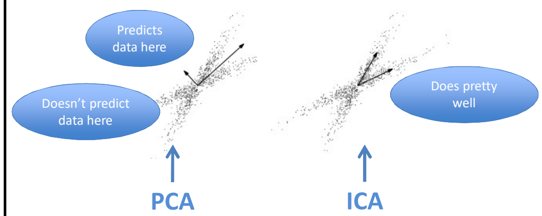
Non-Gaussian data



Images from Lewicki and Sejnowski, *Learning Overcomplete Representations*, 2000.

Comparing to Other Techniques

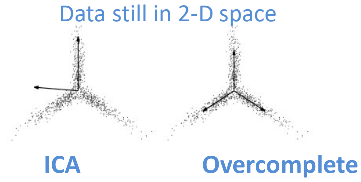
Non-Gaussian data



Images from Lewicki and Sejnowski, *Learning Overcomplete Representations*, 2000.

Comparing to Other Techniques

Data still in 2-D space



Doesn't capture the underlying representation, which Overcomplete representations can do…

Summary

- Overcomplete representations can be more powerful than component analysis techniques.
- Dictionary can be learned from data.
- Relative advantages and disadvantages of the pursuit algorithms.