Machine Learning for Signal Processing

Sparse and Overcomplete Representations

Bhiksha Raj
(slides from Sourish Chaudhuri)

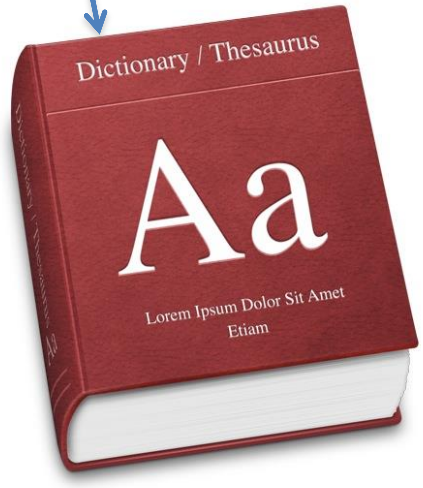
Oct 14, 2014

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?

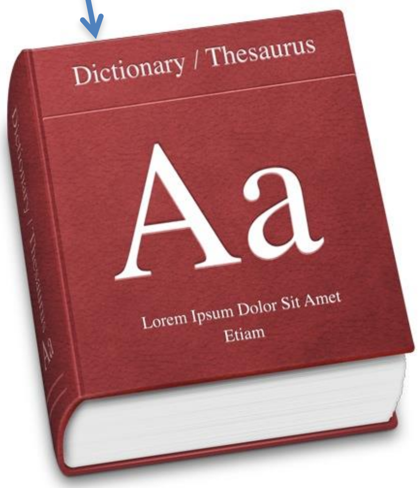
Representing Data

Dictionary (codebook)

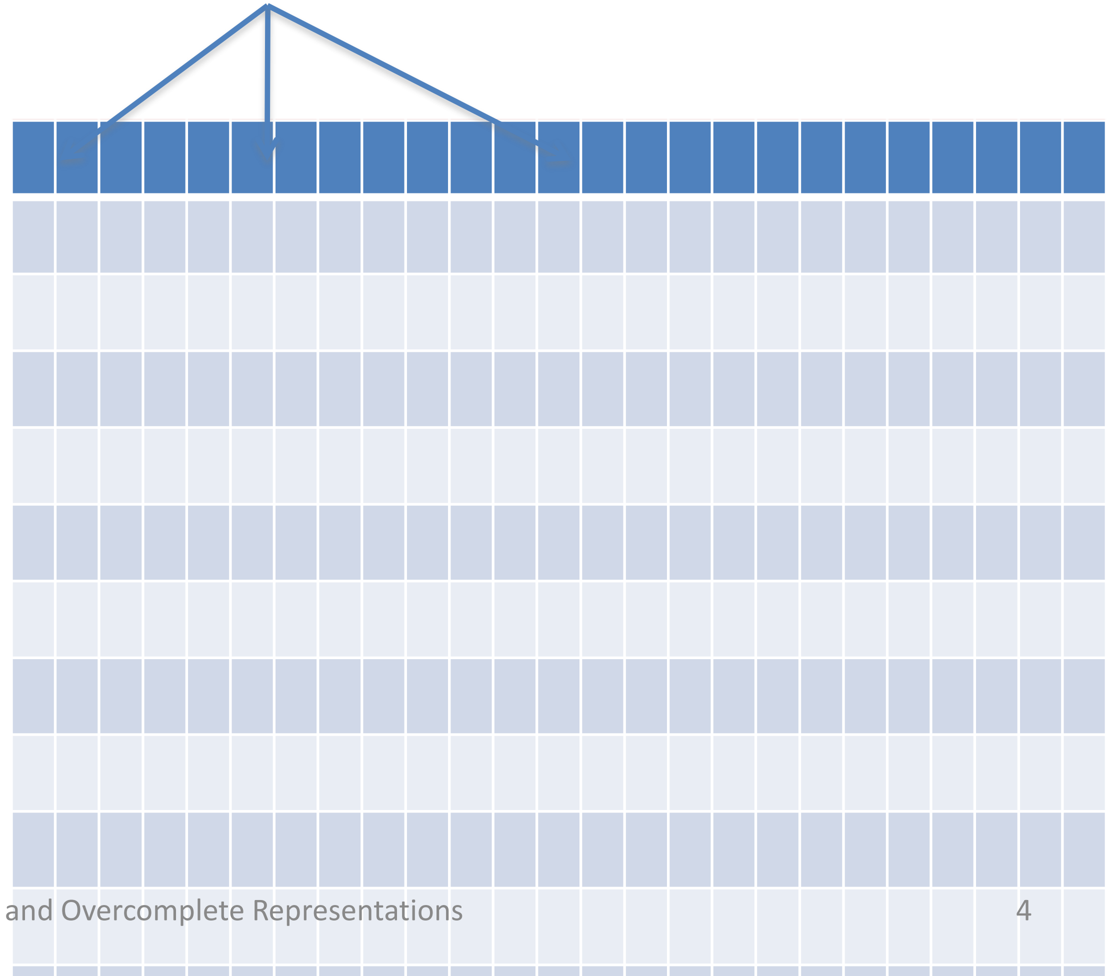


Representing Data

Dictionary

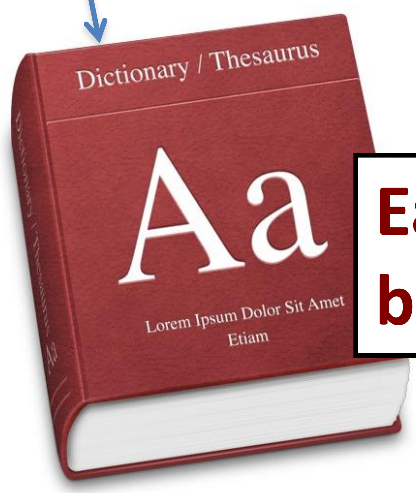


Atoms

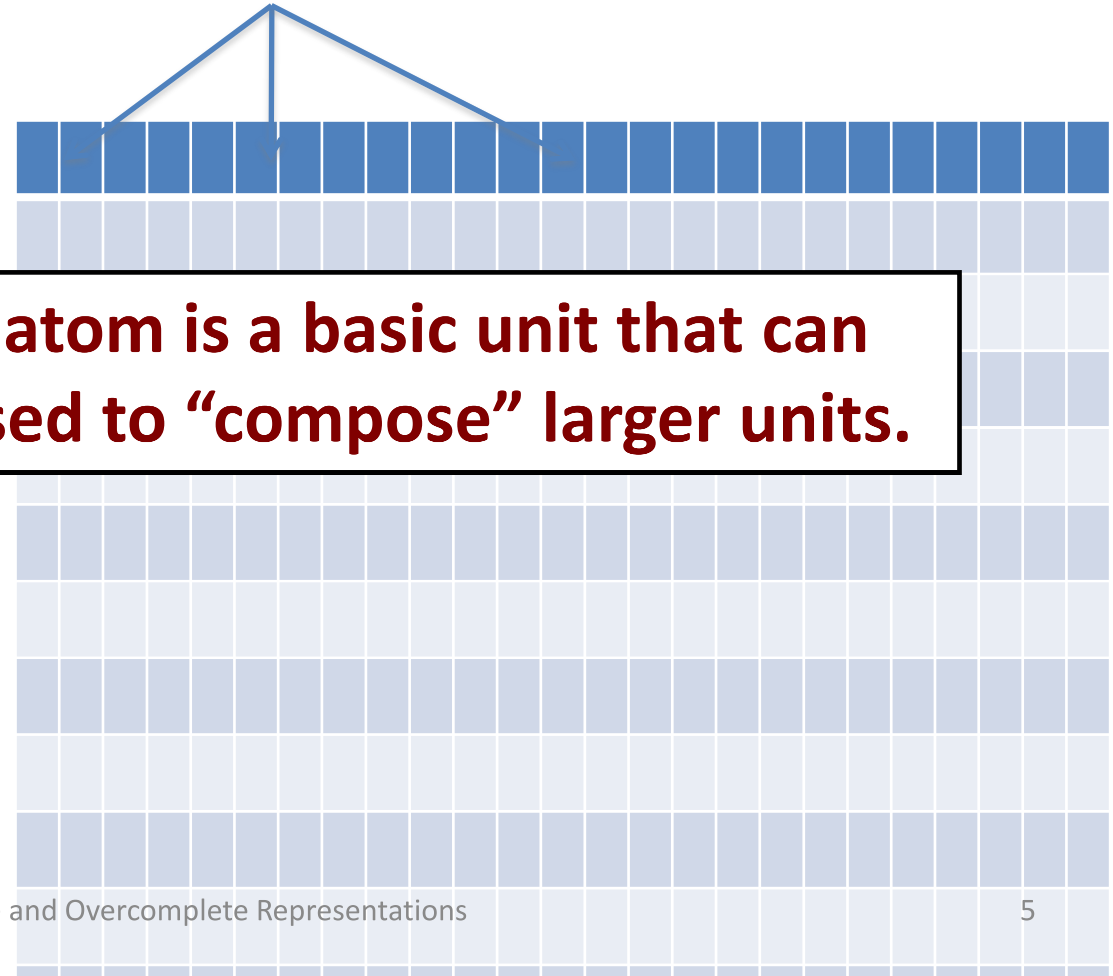


Representing Data

Dictionary

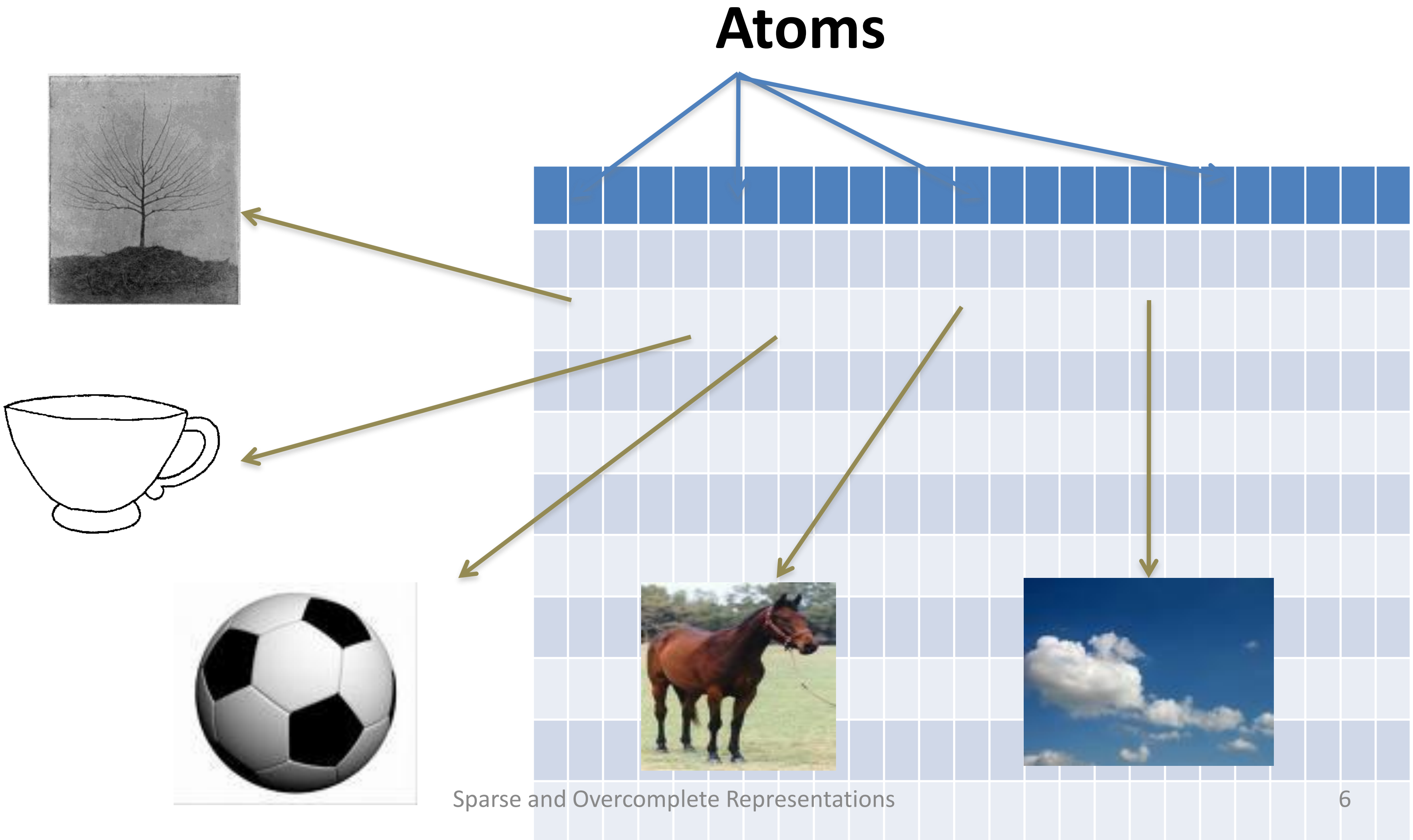


Atoms

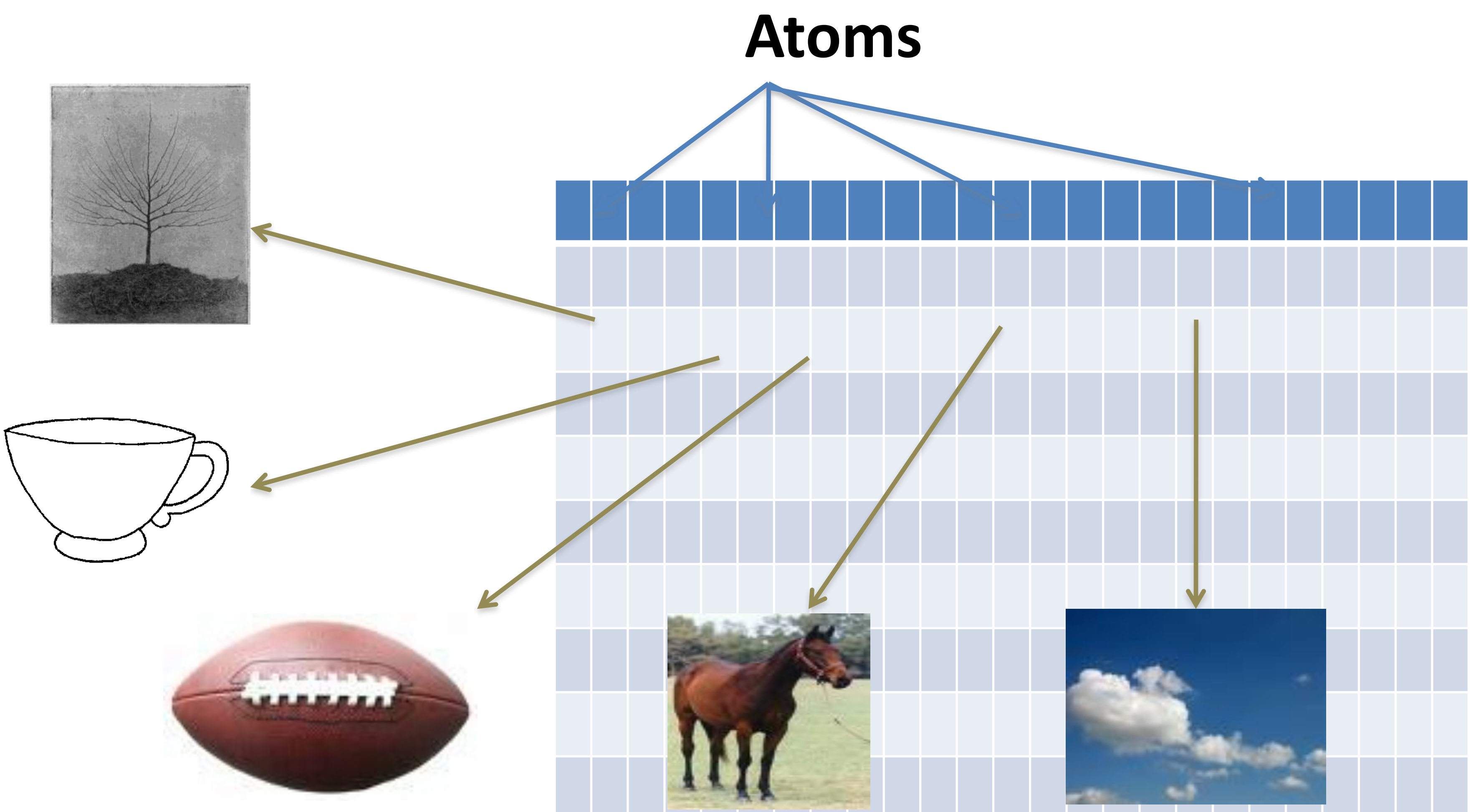


Each atom is a basic unit that can be used to "compose" larger units.

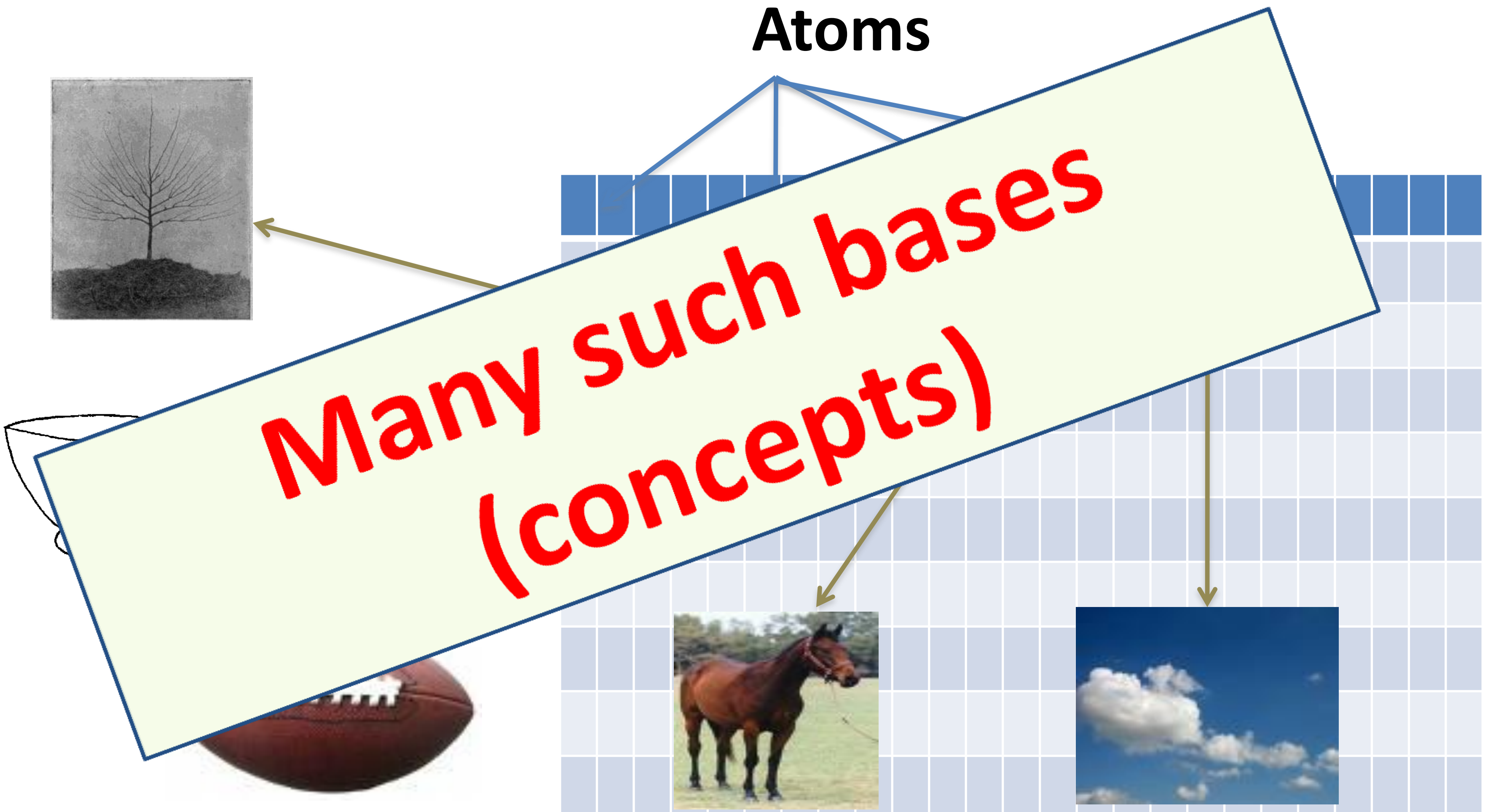
Representing Data



Representing Data



Representing Data



Many such bases (concepts)

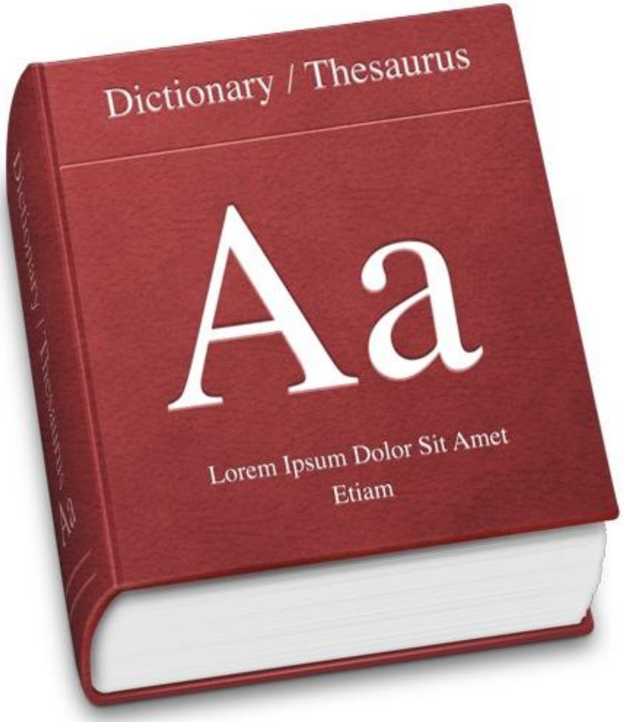
Representing Data

Representing Data

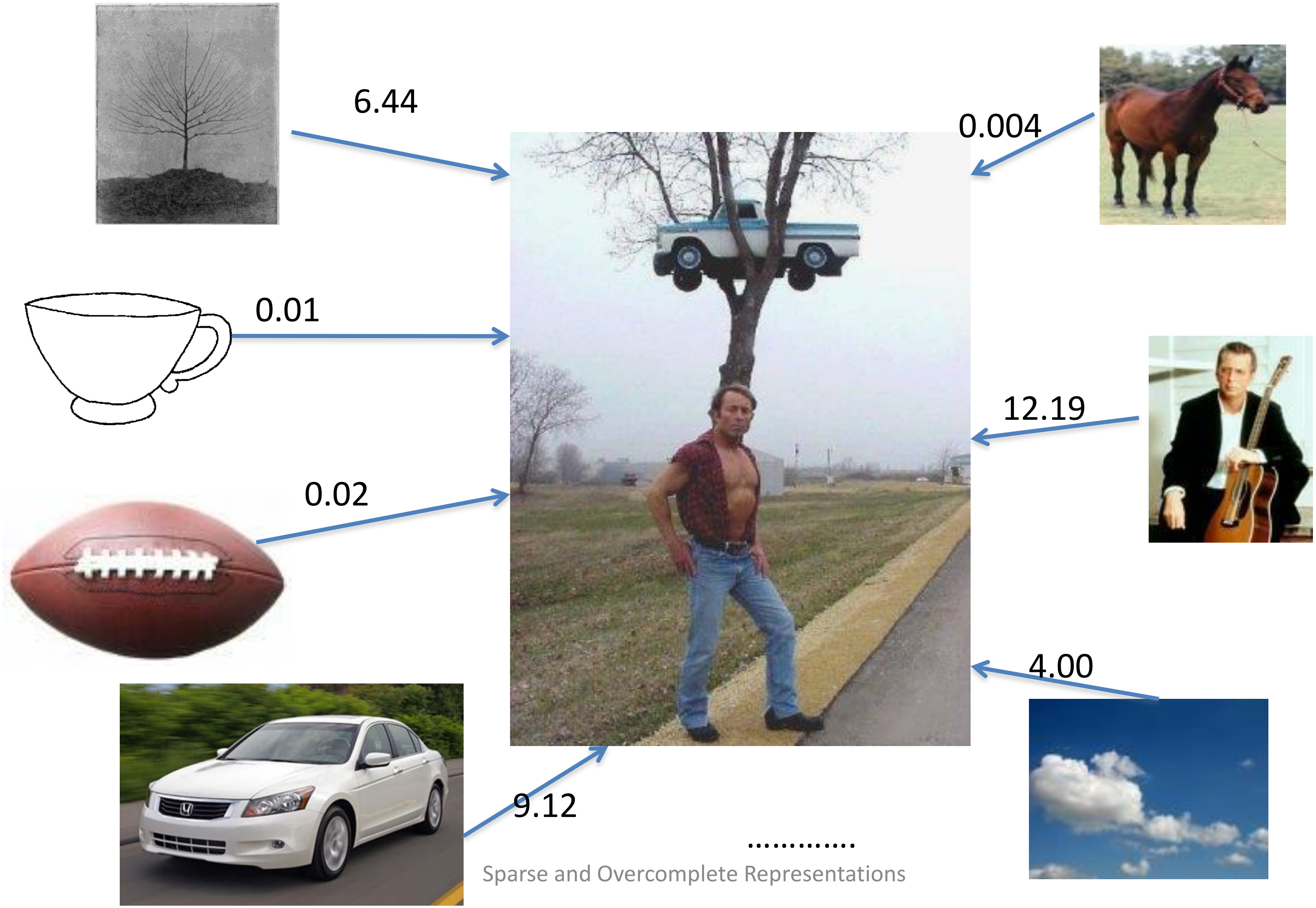
Using concepts that we know…

Representing Data

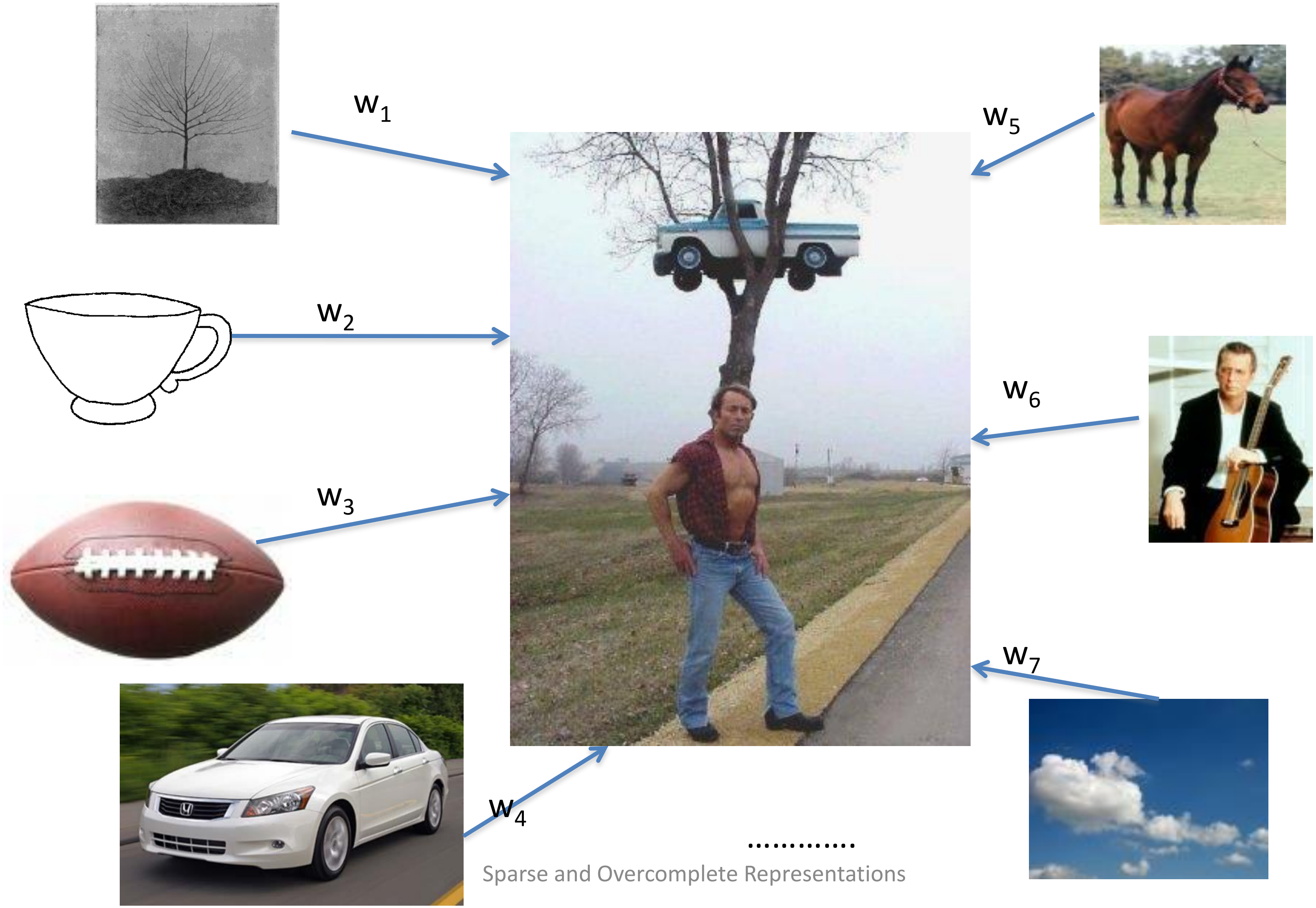
Using concepts that we know…



Representing Data



Representing 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)

    - **N x 4096**

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)

  - **N x 4096** ???

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)

  - **N x 4096** 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)

N x 4096 VERY LARGE!!!

Overcomplete Representations

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

More generally:

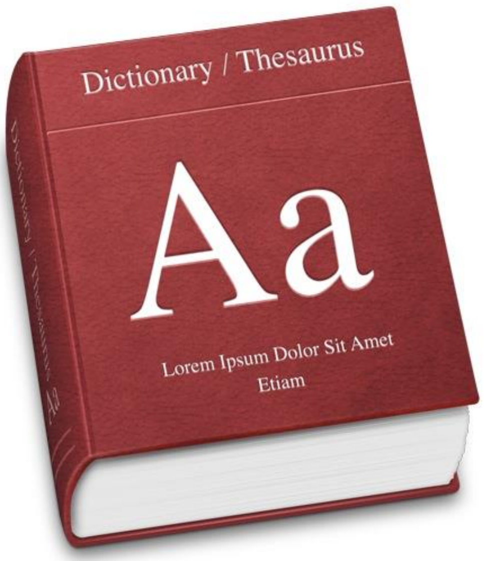
If #(basis vectors) > dimensions of input

we have an **overcomplete** representation

-

➢ N x 4096 VERY LARGE!!!

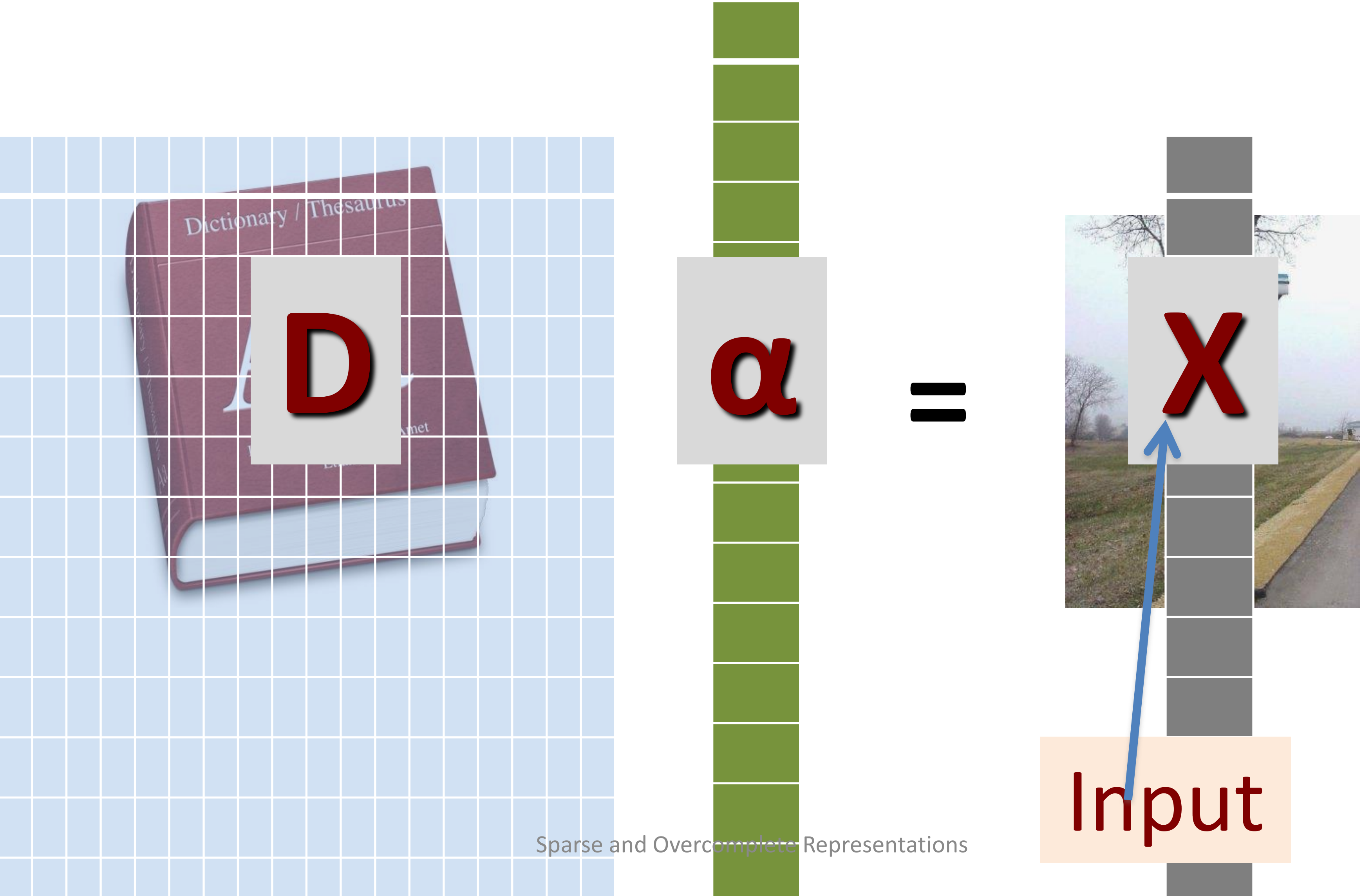
Representing Data



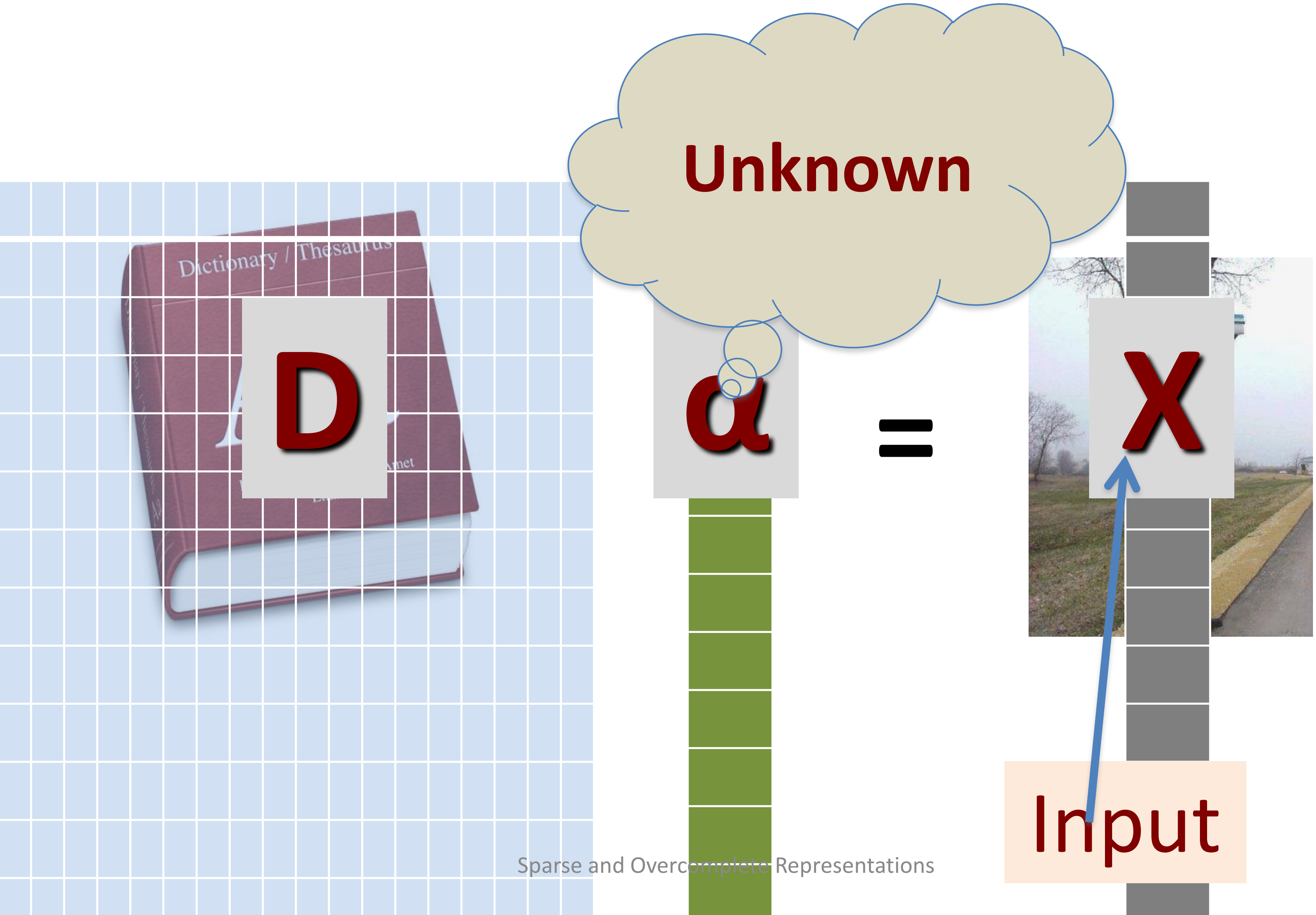
=

Representing Data

Representing Data



Representing Data



Quick Linear Algebra Refresher

- Remember, #(Basis Vectors)= #unknowns

$$D.\alpha = X$$

Basis Vectors

Unknowns

Input data

Quick Linear Algebra Refresher

- Remember, #(Basis Vectors)= #unknowns

$$D.\alpha = X$$

Basis Vectors → D

Unknowns → α

Input data → X

When can we solve for α?

Quick Linear Algebra Refresher

$$D.\alpha = X$$

- When #(Basis Vectors) = dim(Input Data), we have a unique solution
- When #(Basis Vectors) < dim(Input Data), we may have no solution
- When #(Basis Vectors) > dim(Input Data), we have infinitely many solutions

Quick Linear Algebra Refresher

$$D.\alpha = X$$

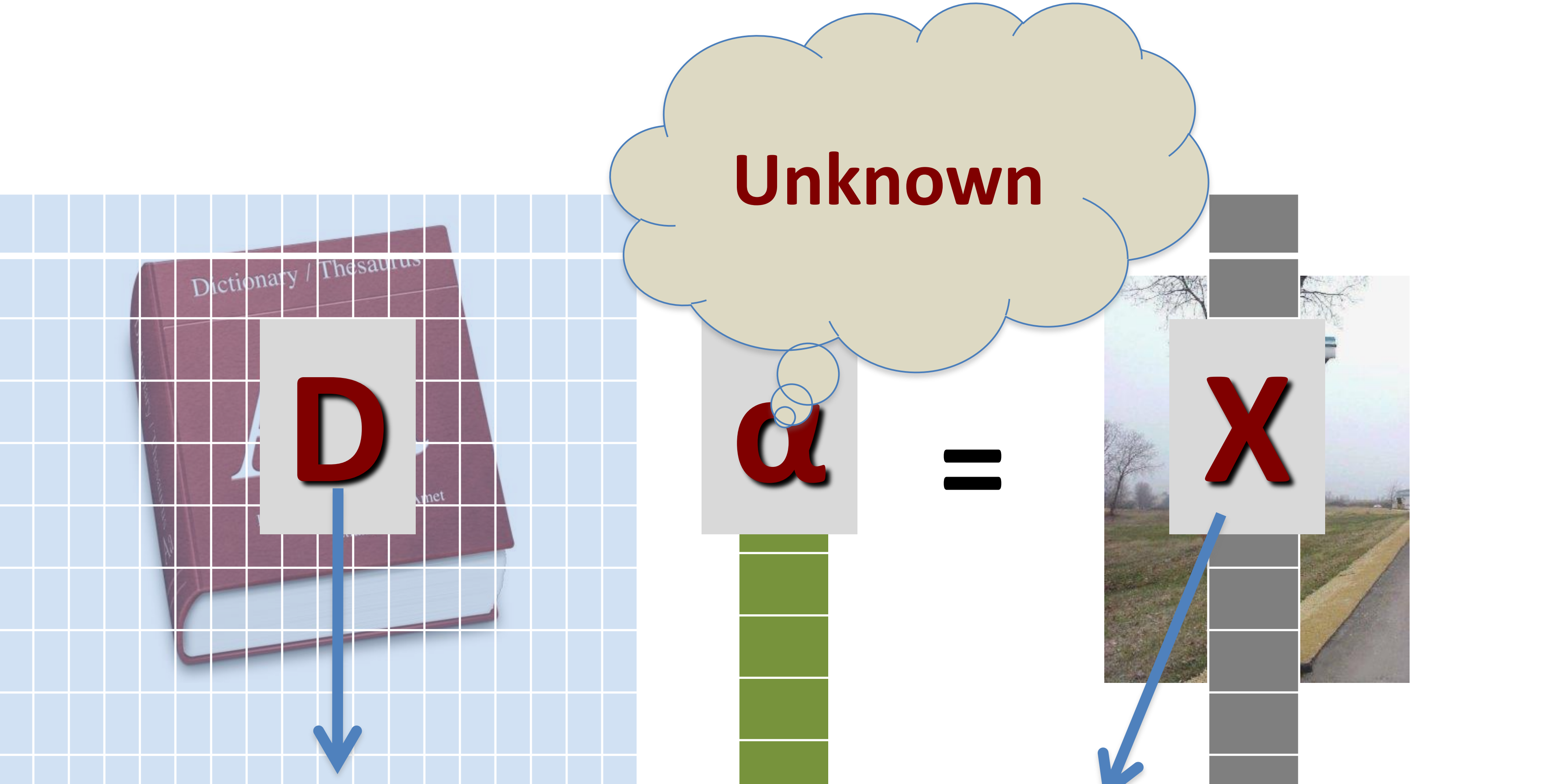
- When #(Basis Vectors) = dim(Input Data), we have a unique solution
- When #(Basis Vectors) < dim(Input Data), we may have no solution
- When #(Basis Vectors) > dim(Input Data), we have infinitely many solutions

Our Case

Overcomplete Representations

#(Basis Vectors) > dimensions of the input

Overcomplete Representation



#(Basis Vectors) > dimensions of the input

Overcomplete Representations

- Why do we use them?
- How do we learn them?

Overcomplete Representations

- Why do we use them?
  - A more natural representation of the real world
  - More flexibility in matching data
  - Can yield a better approximation of the statistical distribution of the data.
- How do we learn them?

Overcompleteness and Sparsity

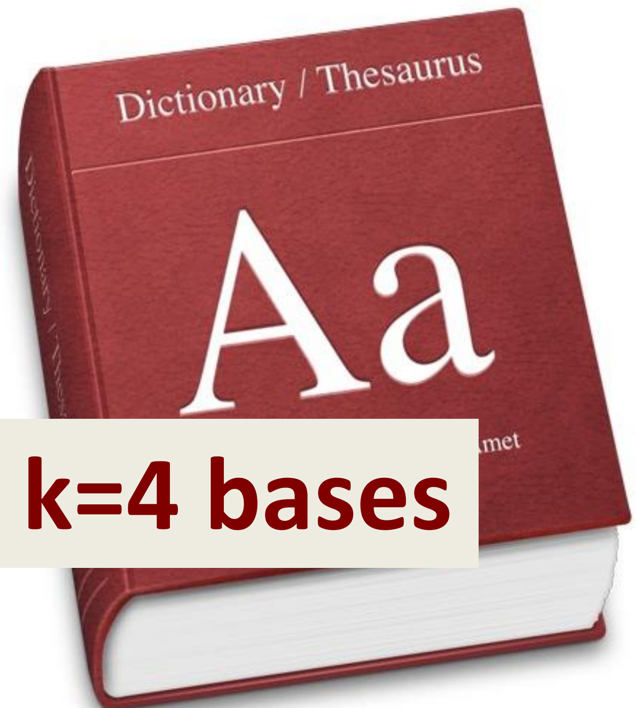
- To solve an overcomplete system of the type:

$$\mathbf{D}.\alpha = \mathbf{X}$$

- 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**.

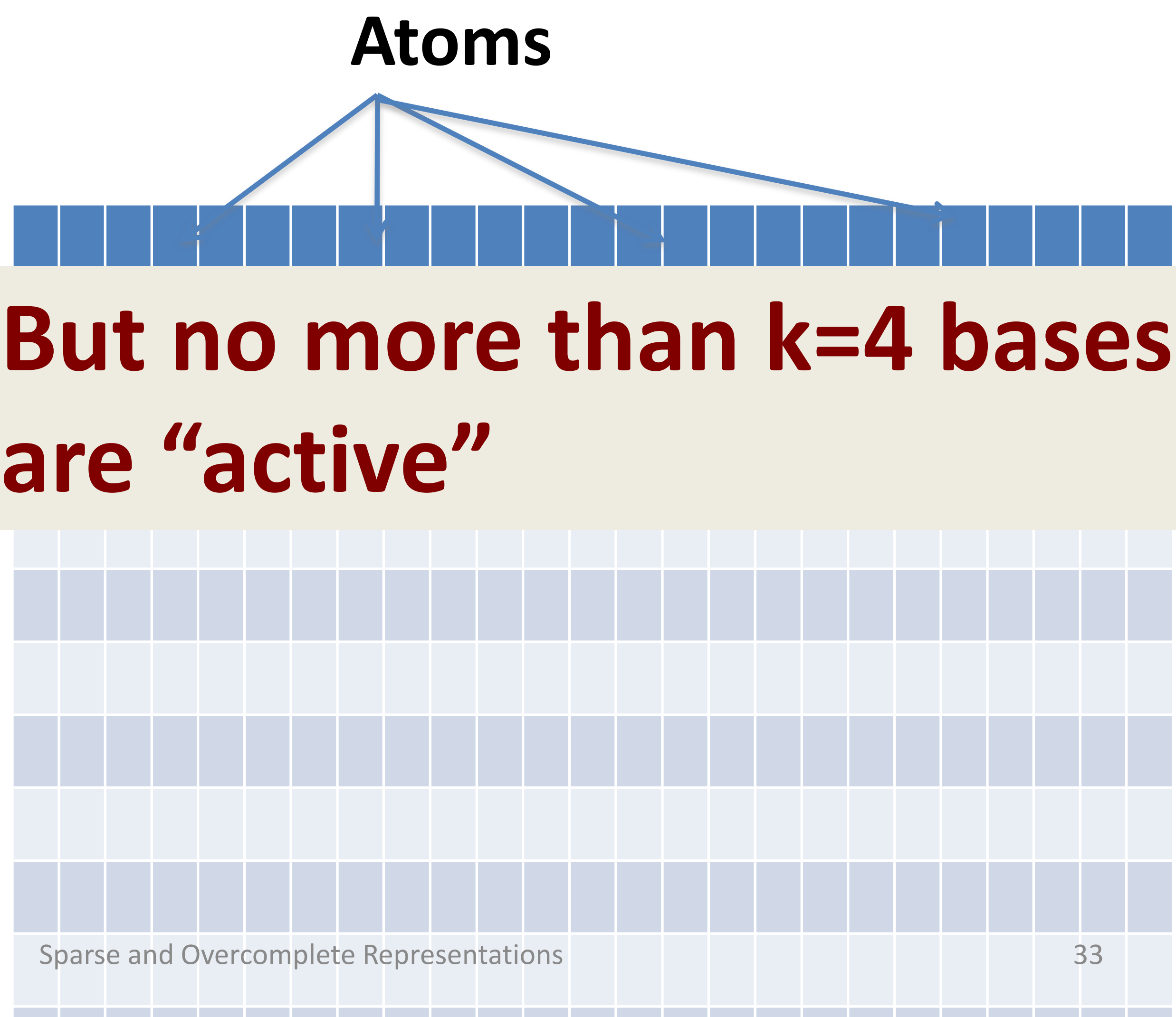
Representing Data

Using bases that we know…



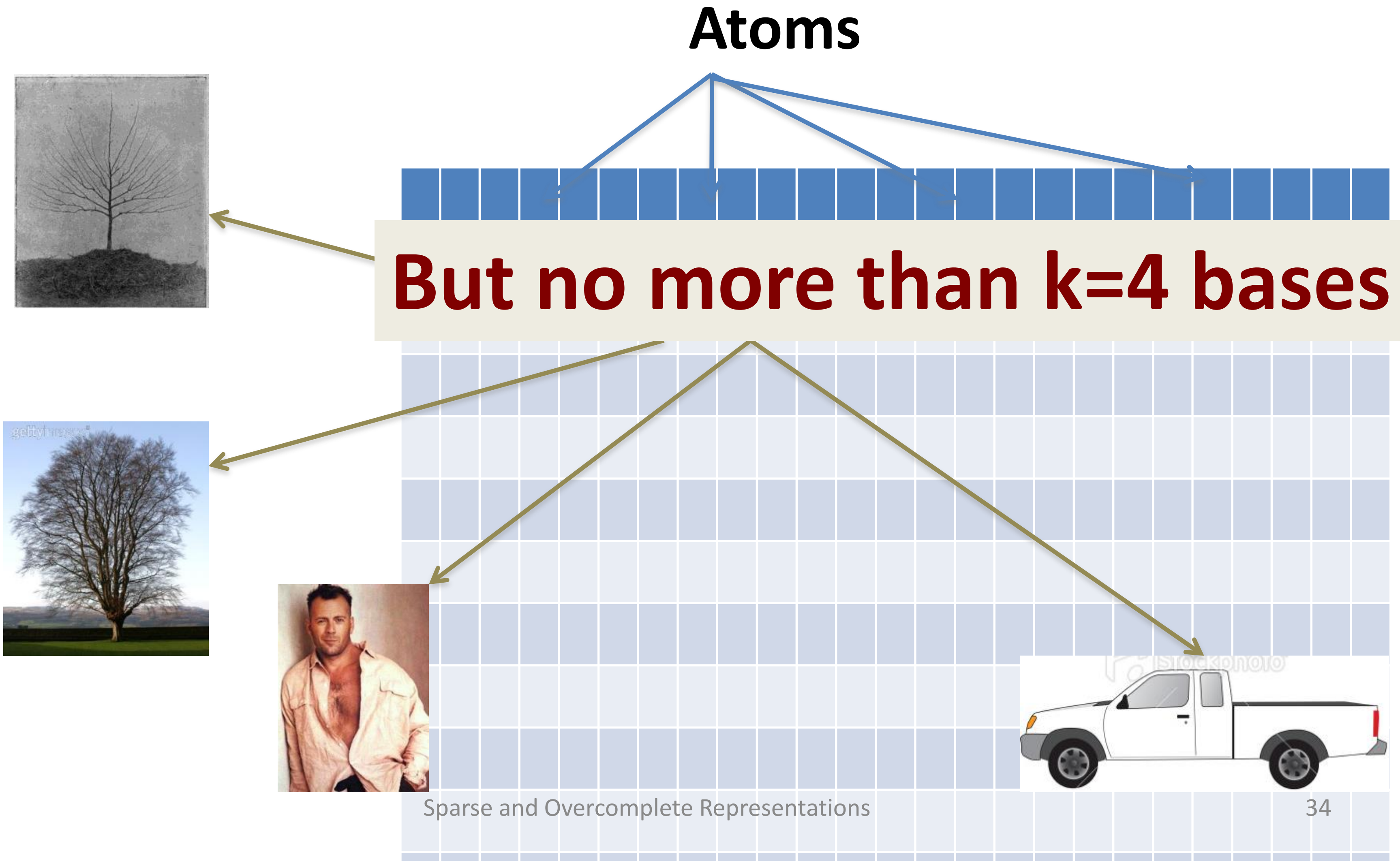
But no more than k=4 bases

Overcompleteness and Sparsity

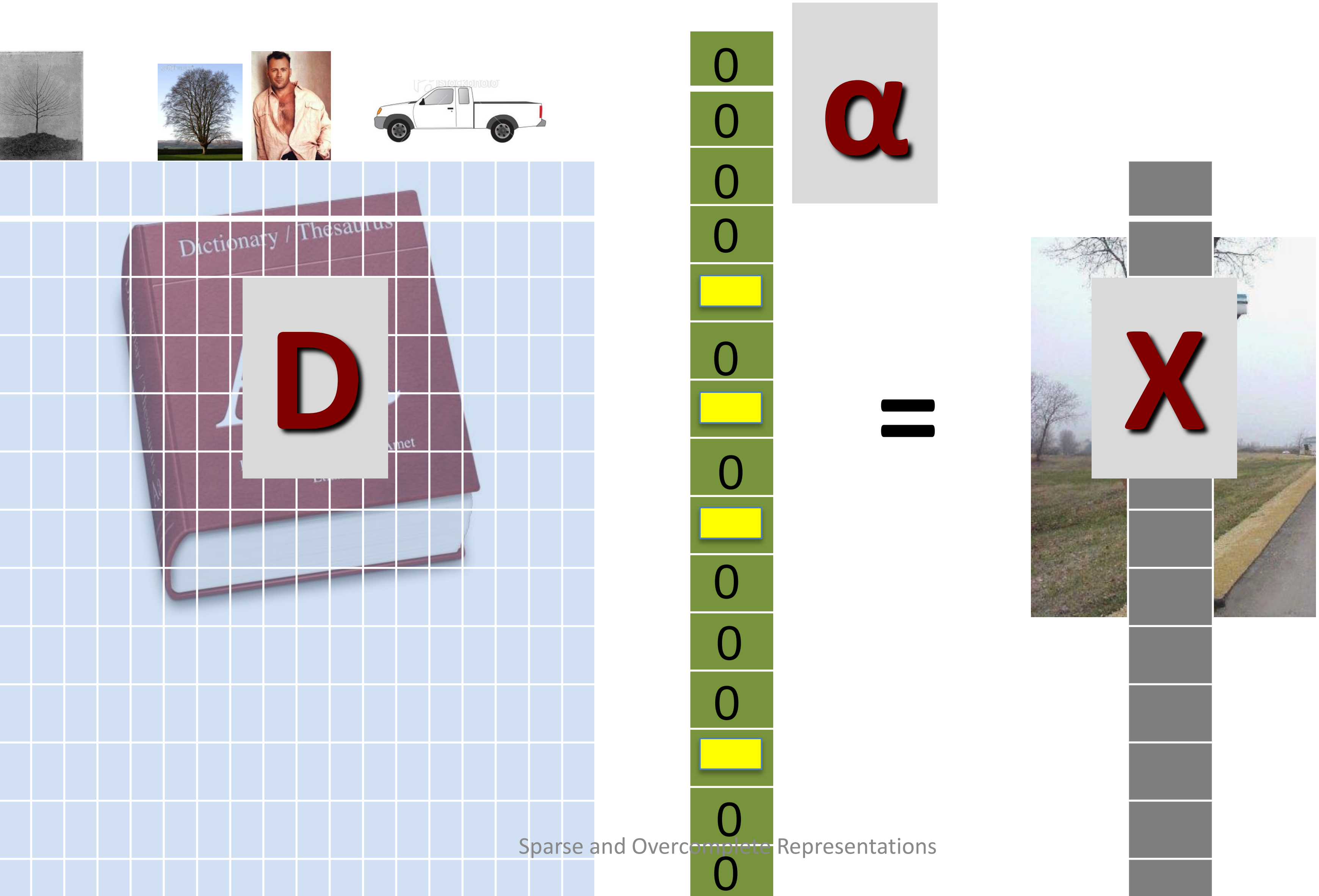


But no more than k=4 bases are "active"

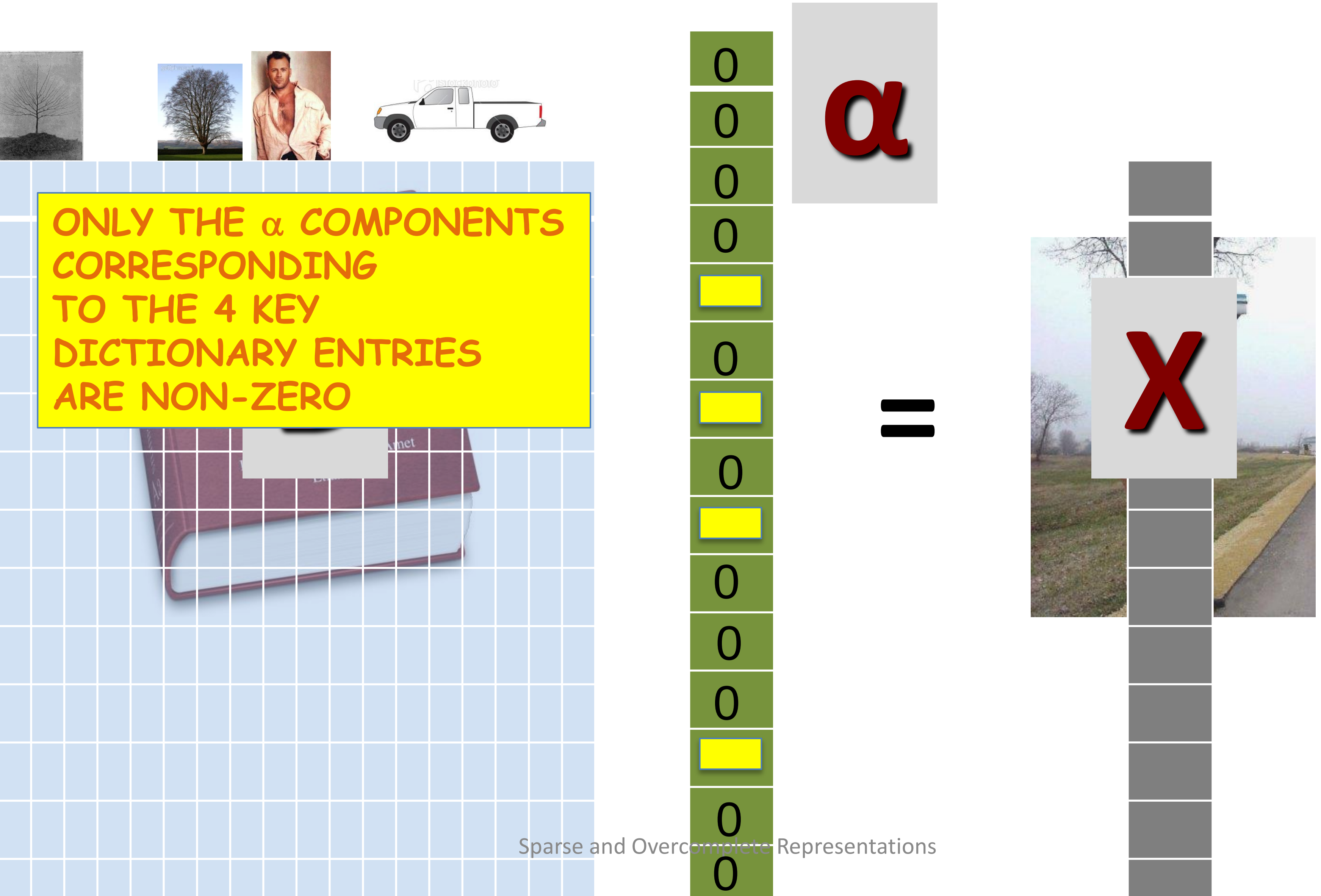
Overcompleteness and Sparsity



No more than 4 bases



No more than 4 bases

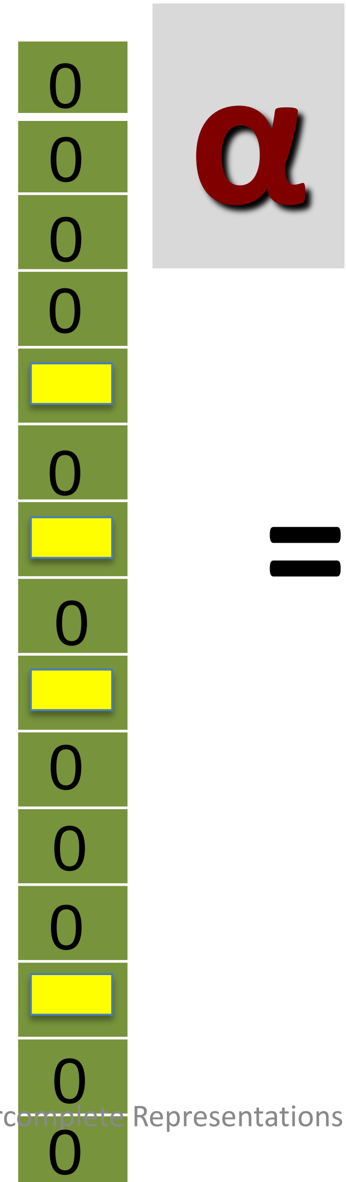


No more than 4 bases

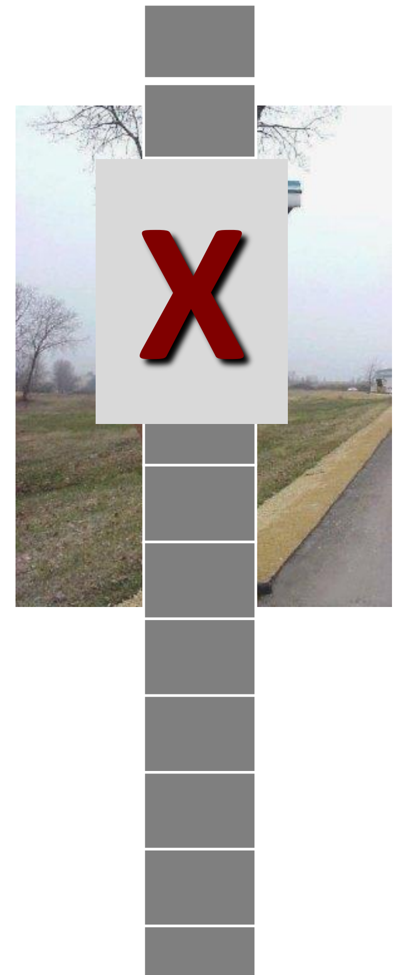
ONLY THE α COMPONENTS CORRESPONDING TO THE 4 KEY DICTIONARY ENTRIES ARE NON-ZERO

MOST OF α IS ZERO!!

α IS *SPARSE*



=



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 basis vectors as possible to do this.

$$\underset{\underline{\alpha}}{Min} \ \left\| \underline{\alpha} \right\|_0$$

$$s.t. \ \ \underline{X} = \mathbf{D}\underline{\alpha}$$

The Sparsity Problem

- We want to use as few basis vectors as possible to do this.

$$\underset{\underline{\alpha}}{Min} \ \|\underline{\alpha}\|_0$$

$$s.t. \ \ \underline{X} = \mathbf{D}\underline{\alpha}$$

Counts the number of non-zero elements in α

The Sparsity Problem

- We want to use as few basis vectors as possible to do this.

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

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

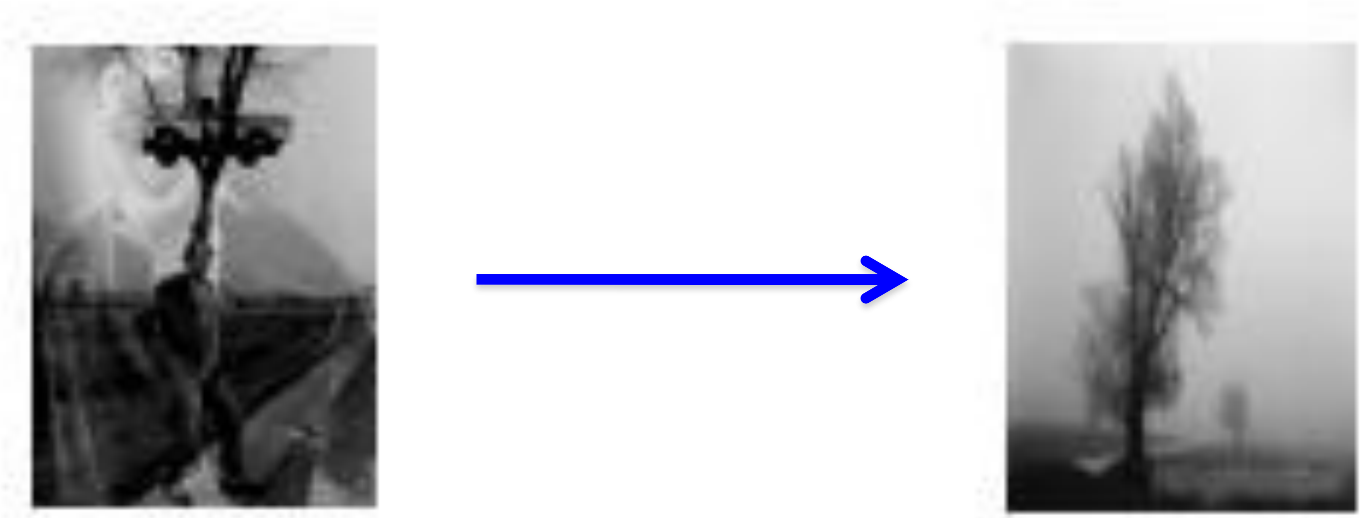
Matching Pursuit

- 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,```

$$\underset{\underline{\alpha}}{Min} \ \left\| \underline{\alpha} \right\|_0$$

$$s.t. \ \ \underline{X} = \mathbf{D}\underline{\alpha}$$

Basis Pursuit

- Remember,

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

In the general case, this is intractable

Basis Pursuit

- Remember,

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

In the general case, this is intractable

Requires combinatorial optimization

Basis Pursuit

- Replace the intractable expression by an expression that is solvable

$$\underset{\underline{\alpha}}{Min} \ \left\| \underline{\alpha} \right\|_1$$

$$s.t. \ \ \underline{X} = \mathbf{D}\underline{\alpha}$$

Basis Pursuit

- Replace the intractable expression by an expression that is solvable

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

This holds when $\mathbf{D}$ obeys the ***Restricted Isometry Property***.

Basis Pursuit

- Replace the intractable expression by an expression that is solvable

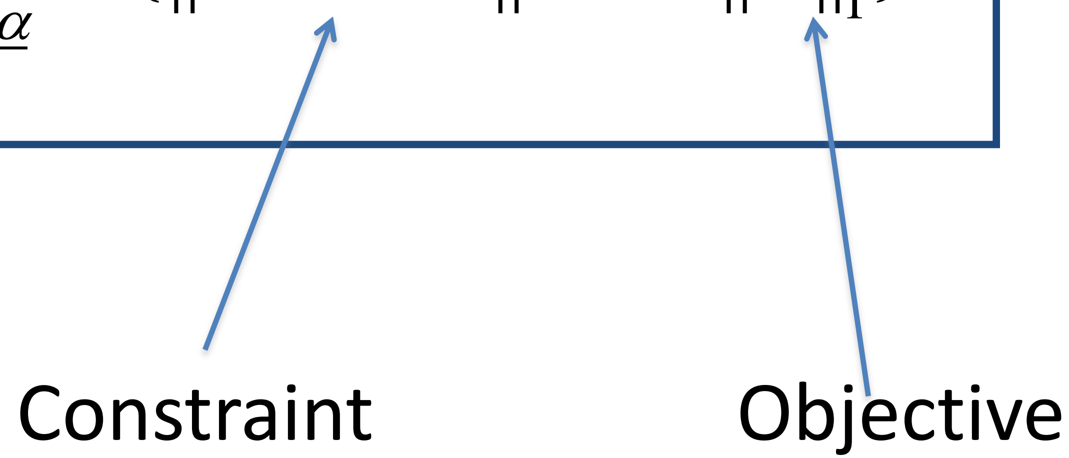
$$\underset{\underline{\alpha}}{Min} \; \|\underline{\alpha}\|_1 \quad \text{(Objective)}$$

$$s.t. \;\; \underline{X} = \mathbf{D}\underline{\alpha} \quad \text{(Constraint)}$$

Basis Pursuit

- We can formulate the optimization term as:

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



Basis Pursuit

- We can formulate the optimization term as:

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

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]

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

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

Basis Pursuit

- There are efficient ways to solve the LASSO formulation. [Link to Matlab code]

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

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

- L_0 constrained optimization

$$\underset{\underline{\alpha}}{Min}\left\|\underline{X} - \mathbf{D}\underline{\alpha}\right\|_2^2 \quad \underline{s.t.\left\|\underline{\alpha}\right\|_0 < C}$$

- L_1 minimization

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

- L_1 constrained optimization

$$\underset{\underline{\alpha}}{Min}\left\|\underline{X} - \mathbf{D}\underline{\alpha}\right\|_2^2 \quad \underline{s.t.\left\|\underline{\alpha}\right\|_1 < C}$$

Many Other Methods..

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

Applications of Sparse Representations

- Many many applications
  - Signal representation
  - Statistical modelling
  - ..
- Two extremely popular signal processing applications:
  - Compressive sensing
  - Denoising

Compressive Sensing

- Recall the Nyquist criterion?
- To reconstruct a signal, you need to sample at twice the maximum frequency of the original signal

Compressive Sensing

- Recall the Nyquist criterion?
- To reconstruct a signal, you need to sample at twice the frequency of the original signal
- Is it possible to reconstruct signals when they have not been sampled so as to satisfy the Nyquist criterion?

Compressive Sensing

- Recall the Nyquist criterion?
- To reconstruct a signal, you need to sample at twice the frequency of the original signal
- Is it possible to reconstruct signals when they have not been sampled so as to satisfy the Nyquist criterion?
- Under specific criteria, yes!!!!

Compressive Sensing

- What criteria?

Compressive Sensing

- What criteria?

Sparsity!

Compressive Sensing

- What criteria?

Sparsity!

- Exploit the structure of the data
- Most signals are sparse, in some domain

Applications of Sparse Representations

- Two extremely popular applications:
  - Compressive sensing
  - Denoising

Applications of Sparse Representations

- Two extremely popular applications:
  - Compressive sensing
  - Denoising

Denoising

- As the name suggests, remove noise!

Denoising

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

Image Denoising

- Here's what we want

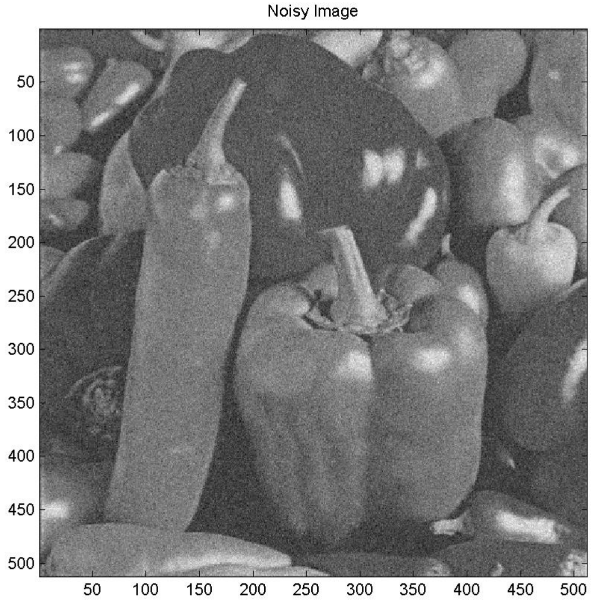
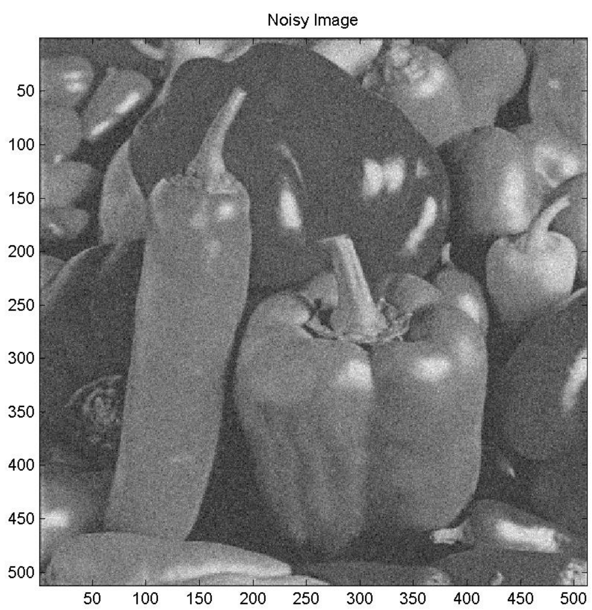


Image Denoising

- Here's what we want



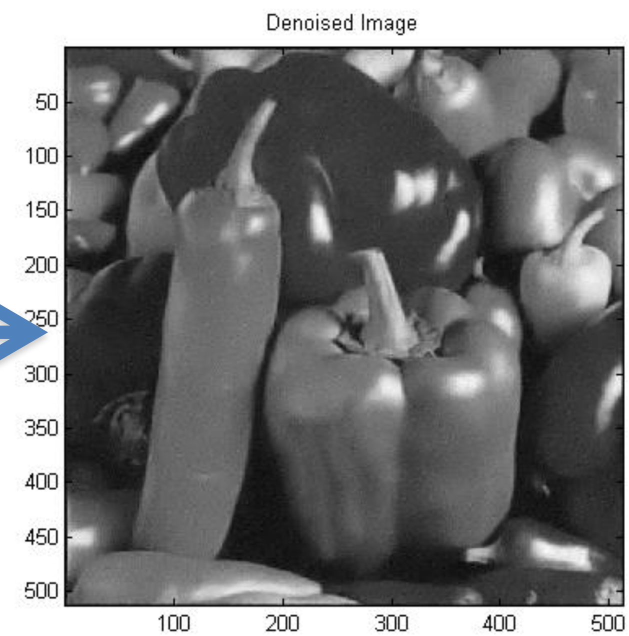


Image Denoising

- Here's what we want

Denoising

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

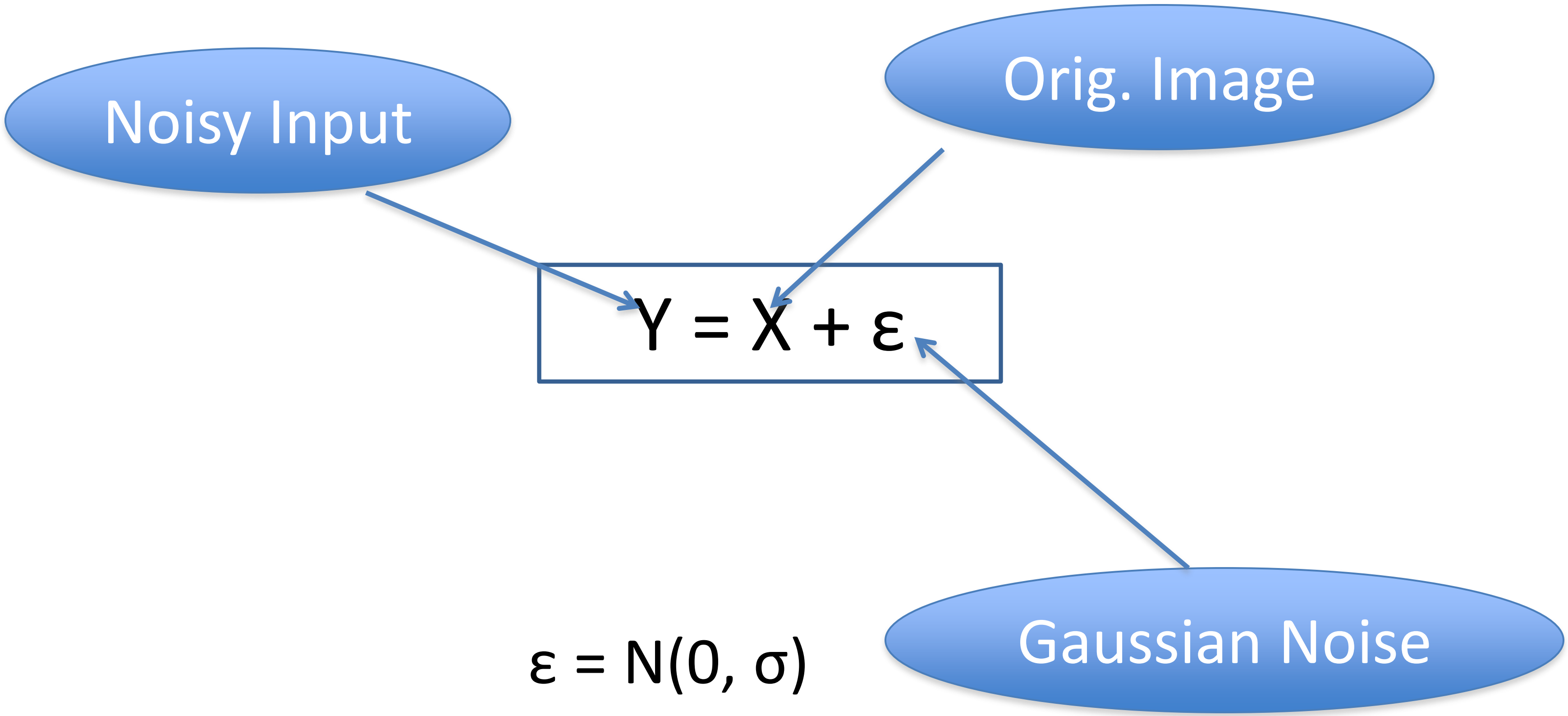
A more general take-away:

How to learn the dictionaries

The Image Denoising Problem

- Given an image
- Remove Gaussian additive noise from it

Image Denoising



$\varepsilon = N(0, \sigma)$

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
- Yes, 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
- Yes, 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 $\sqrt{n} \times \sqrt{n}$ 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:

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

Image Denoising

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

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

Can Matching Pursuit solve this?

Image Denoising

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

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

Can Matching Pursuit solve this? **Yes**

Image Denoising

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

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

Can Matching Pursuit solve this? **Yes**

What constraints does it need?

Image Denoising

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

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

Can Basis Pursuit solve this?

Image Denoising

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

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

But this is intractable!

Image Denoising

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

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

Can Basis Pursuit solve this?

Image Denoising

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

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

Can Basis Pursuit solve this? **Yes**

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

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

$$+ \sum_{ij} \lambda_{ij} \|\underline{\alpha_{ij}}\|_0 \}$$

Image Denoising

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

$$+\sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

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

Image Denoising

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

$$+\sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

Error bounding in each patch

-what is R_{ij}?

-How many terms in the summation?

Image Denoising

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

$$+ \sum_{ij} \lambda_{ij} \|\underline{\alpha_{ij}}\|_0 \}$$

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.

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

$$+ \sum_{ij} \lambda_{ij} \|\underline{\alpha_{ij}}\|_0 \}$$

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

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

$$+ \sum_{ij} \lambda_{ij} \|\underline{\alpha_{ij}}\|_0 \}$$

How do we estimate all 3 at once?

Fix 2, and find the optimal 3rd.

Image Denoising

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

$$+ \sum_{ij} \lambda_{ij} \left\| \underline{\alpha_{ij}} \right\|_0 \}$$

Initialize X = Y

Image Denoising

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

$$+\sum_{ij}\lambda_{ij}\left\|\underline{\alpha_{ij}}\right\|_0\}$$

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

K-SVD vs K-Means

- Kmeans: Given data $\mathbf{Y}$
  - Find $\mathbf{D}$ and α such that
  - Error = $||\mathbf{Y} - \mathbf{D}\alpha||^2$ is minimized, with constraint
  - $|\alpha_i|_0 = 1$

- K-SVD
  - Find $\mathbf{D}$ and α such that
  - Error = $||\mathbf{Y} - \mathbf{D}\ \alpha||^2$ is minimized, with constraint
  - $|\alpha_i|_0 < T$

Image Denoising

- Updating **D**
  - For each basis vector, compute its contribution to the image

$$E_k = Y - \sum_{j \neq k} D_j \alpha_j$$

Image Denoising

- Updating **D**
  - For each basis vector, compute its contribution to the image
  - Eigen decomposition of E_k

$$E_k = U \Delta V^T$$

K-SVD

- Updating **D**
  - For each basis vector, compute its contribution to the image
  - Eigen decomposition of E_k
  - Take the principal eigen vector as the updated basis vector.

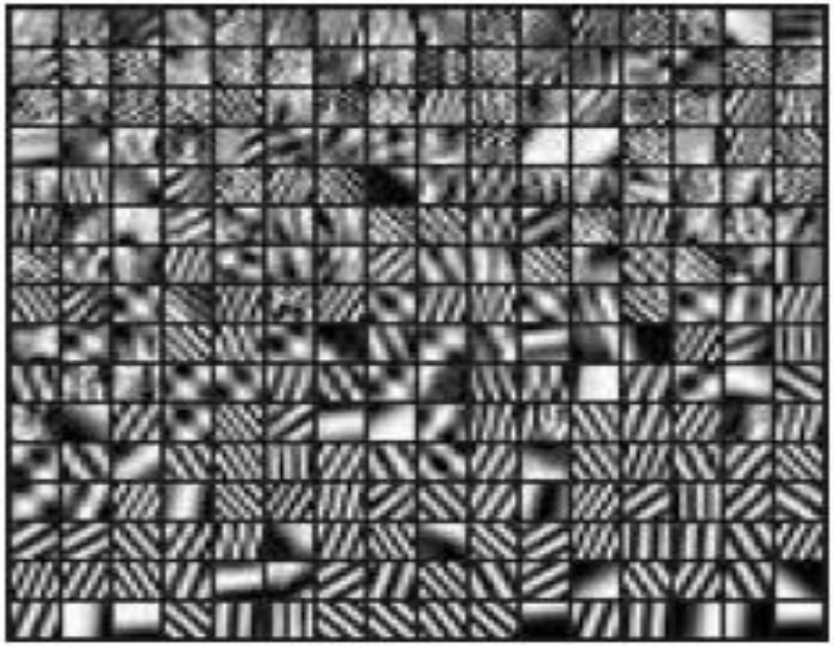
$$D_k = U_1$$

- Update every entry in **D**

K-SVD

- Initialize **D**
- Estimate $\boldsymbol{\alpha}$
- Update every entry in **D**
- Iterate

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

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

$$+\sum_{ij}\lambda_{ij}\left\|\underline{\alpha_{ij}}\right\|_0\}$$

Const. wrt X

We know D and α

The quadratic term above has a closed-form solution

Image Denoising

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

$$+\sum_{ij}\lambda_{ij}\|\underline{\alpha_{ij}}\|_0\}$$

Const. wrt X

We know D and α

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

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

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 $i = 1, 2, \ldots, N$

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

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

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

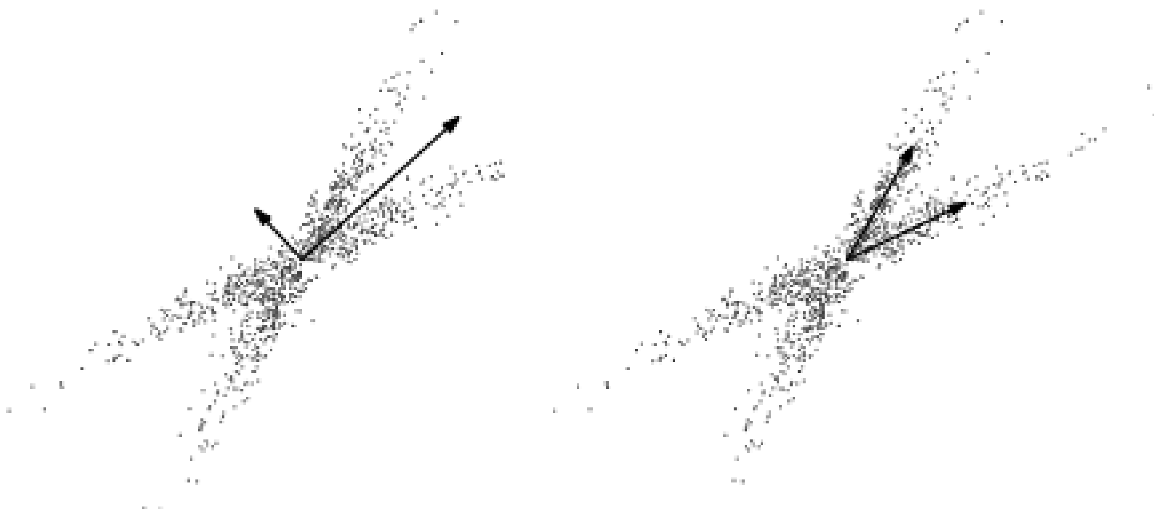
$$\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}^j,$$

- 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 $J = J + 1$.

Comparing to Other Techniques

Non-Gaussian data

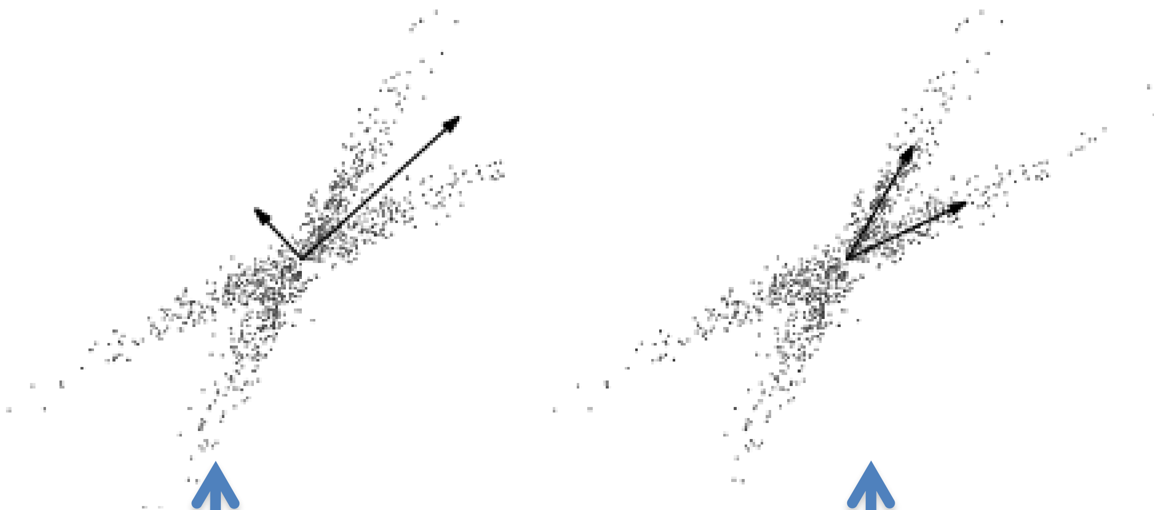


PCA of ICA   Which is which?

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

Comparing to Other Techniques

Non-Gaussian data

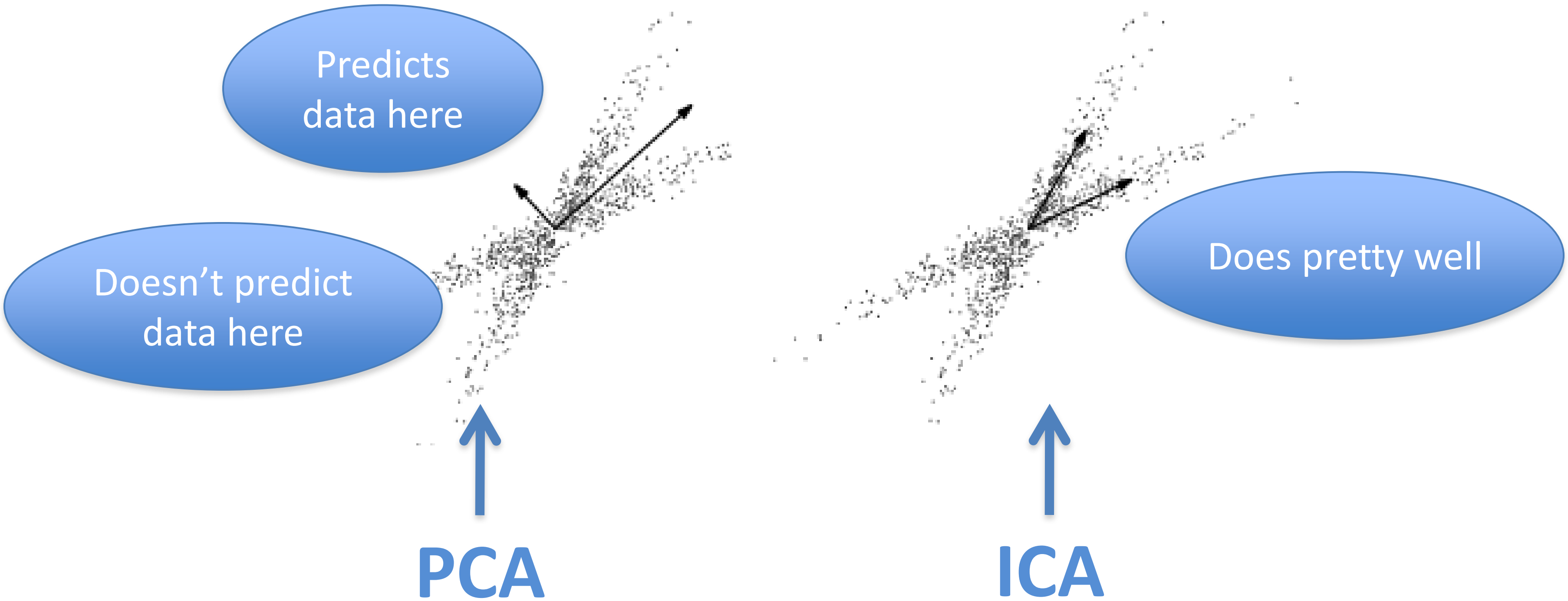


PCA **ICA**

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

Comparing to Other Techniques

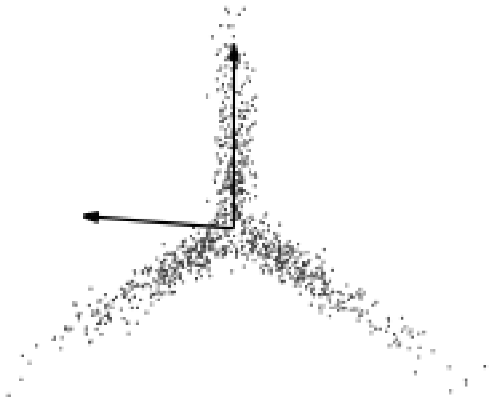
Non-Gaussian data

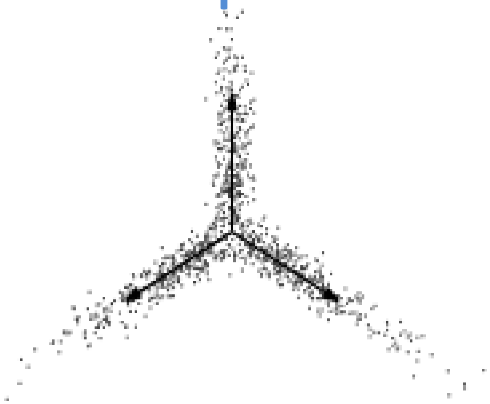


Images from Lewicki and Sejnowski, *Learning Overcomplete Representations*, 2000.

Comparing to Other Techniques

Data still in 2-D space





ICA

Overcomplete

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.