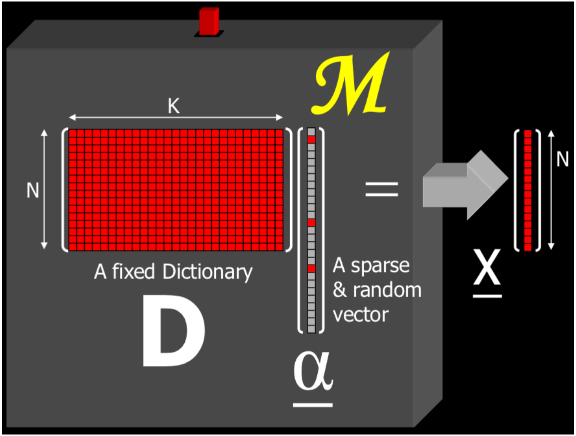
Sparse Coding: An Overview

Brian Booth

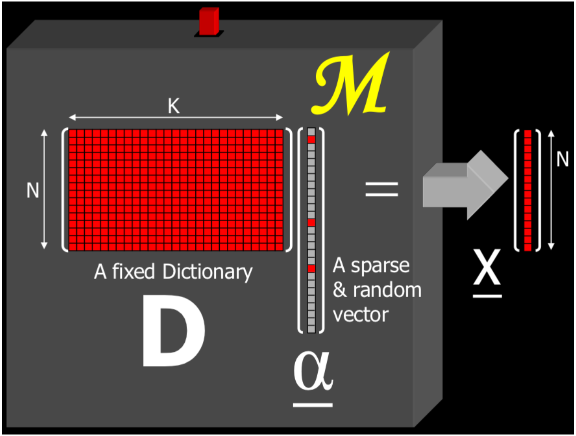
SFU Machine Learning Reading Group

November 12, 2013

The aim of sparse coding

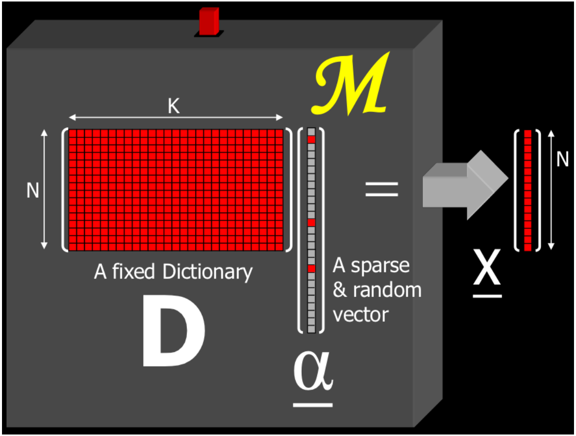


The aim of sparse coding



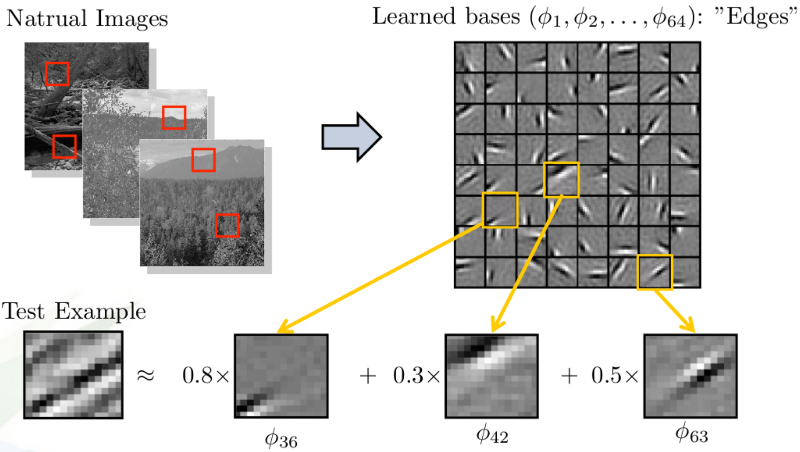
- Every column of **D** is a prototype

The aim of sparse coding



- Every column of **D** is a prototype
- Similar to, but more general than, PCA

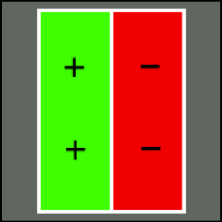
Example: Sparse Coding of Images



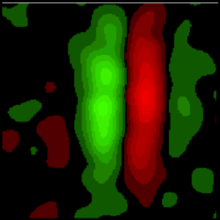
$$[\alpha_1, \ldots, \alpha_{64}] = [0, \ldots, 0.8, \ldots, 0.3, \ldots, 0.5, \ldots, 0]$$

Sparse Coding in V1

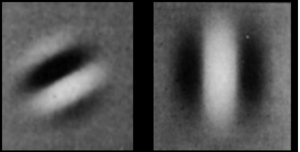
The first stage of visual processing in the brain (V1) does "edge detection."



Schematic of simple cell

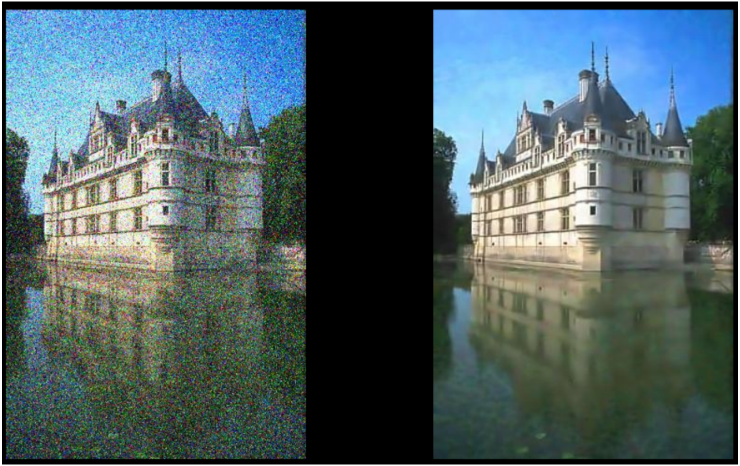


Actual simple cell

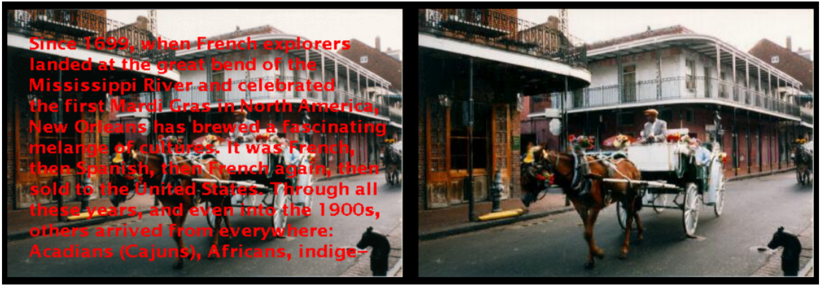


"Gabor functions."

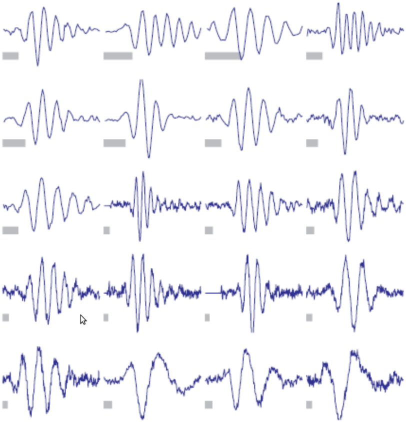
Example: Image Denoising



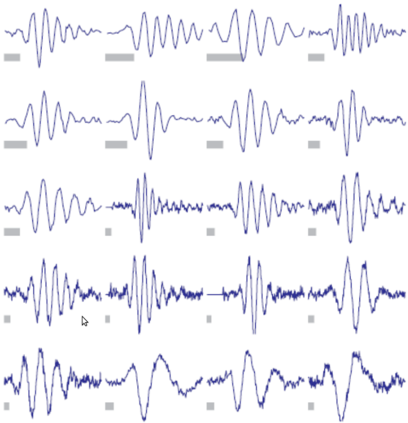
Example: Image Restoration

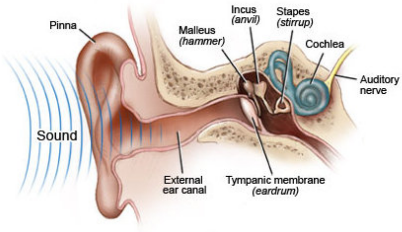


Sparse Coding and Acoustics



Sparse Coding and Acoustics





- Inner ear (cochlea) also does sparse coding of frequencies

Sparse Coding and Natural Language Processing

$$
\begin{array}{r}
\text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \text{able} \\ \text{account} \\ \text{acid} \\ \text{across} \\ \dots \\ \text{baby} \\ \text{back} \\ \dots \\ \text{cradle} \\ \dots \\ \dots \\ \dots \\ \text{zylophone}
\end{array}
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\approx
0.7 \times \underset{\phi_{36}}{\begin{bmatrix} .1 \\ .2 \\ 0 \\ 0 \\ .1 \\ .1 \\ 0 \\ 0 \\ 1.2 \\ 0 \\ 0 \\ 1 \\ 0 \\ .1 \\ 0 \\ .1 \end{bmatrix}}
\quad 0.4 \times \underset{\phi_{42}}{\begin{bmatrix} 1.8 \\ 0 \\ .1 \\ 0 \\ 0 \\ .1 \\ 0 \\ .2 \\ 0 \\ 0 \\ .2 \\ .1 \\ 0 \\ .1 \\ .1 \\ 0 \end{bmatrix}}
\quad 0.1 \times \underset{\phi_{63}}{\begin{bmatrix} .1 \\ 0 \\ 0 \\ 0.8 \\ 0 \\ 0 \\ .1 \\ .2 \\ .1 \\ .2 \\ 0 \\ .3 \\ 0 \\ 0 \\ 0 \\ 1.2 \end{bmatrix}}
$$

$$\text{Document} \approx 0.7 \times \text{Topic36} + 0.4 \times \text{Topic42} + 0.1 \times \text{Topic63}$$

Outline

Outline

1. Introduction: Why Sparse Coding?
2. Sparse Coding: The Basics
3. Adding Prior Knowledge

Outline

1. Introduction: Why Sparse Coding?
2. Sparse Coding: The Basics
3. Adding Prior Knowledge
4. Conclusions

Outline

Outline

The aim of sparse coding, revisited

We assume our data $\mathbf{x}$ satisfies

$$\mathbf{x} \approx \sum_{i=1}^{n} \alpha_i \mathbf{d}_i = \alpha \mathbf{D}$$

The aim of sparse coding, revisited

We assume our data **x** satisfies

$$\mathbf{x} \approx \sum_{i=1}^{n} \alpha_i \mathbf{d}_i = \alpha \mathbf{D}$$

Learning:

- Given training data $\mathbf{x}^j, j \in \{1, \cdots, m\}$
- Learn dictionary **D** and sparse code α

The aim of sparse coding, revisited

We assume our data **x** satisfies

$$\mathbf{x} \approx \sum_{i=1}^{n} \alpha_i \mathbf{d}_i = \alpha \mathbf{D}$$

Learning:

- Given training data $\mathbf{x}^j, j \in \{1, \cdots, m\}$
- Learn dictionary **D** and sparse code α

Encoding:

- Given test data **x**, dictionary **D**
- Learn sparse code α

Learning: The Objective Function

Dictionary learning involves optimizing:

$$\arg \min_{\{\mathbf{d}_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|\mathbf{x}^j - \sum_{i=1}^{n} \alpha_i^j \mathbf{d}_i\|^2$$

Learning: The Objective Function

Dictionary learning involves optimizing:

$$\arg \min_{\{\mathbf{d}_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|\mathbf{x}^j - \sum_{i=1}^{n} \alpha_i^j \mathbf{d}_i\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha_i^j|$$

Learning: The Objective Function

Dictionary learning involves optimizing:

$$\arg\min_{\{\mathbf{d}_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|\mathbf{x}^j - \sum_{i=1}^{n} \alpha_i^j \mathbf{d}_i\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha_i^j|$$

$$\text{subject to } \|\mathbf{d}_i\|^2 \leq c, \quad \forall i = 1, \cdots, n.$$

Learning: The Objective Function

Dictionary learning involves optimizing:

$$\arg\min_{\{\mathbf{d}_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|\mathbf{x}^j - \sum_{i=1}^{n} \alpha_i^j \mathbf{d}_i\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha_i^j|$$

$$\text{subject to } \|\mathbf{d}_i\|^2 \leq c, \quad \forall i = 1, \cdots, n.$$

In matrix notation:

$$\arg\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{AD}\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$

$$\text{subject to } \sum_i \mathbf{D}_{i,j}^2 \leq c, \quad \forall i = 1, \cdots, n.$$

Learning: The Objective Function

Dictionary learning involves optimizing:

$$\arg\min_{\{\mathbf{d}_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|\mathbf{x}^j - \sum_{i=1}^{n} \alpha_i^j \mathbf{d}_i\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha_i^j|$$
$$\text{subject to } \|\mathbf{d}_i\|^2 \leq c, \quad \forall i = 1, \cdots, n.$$

In matrix notation:

$$\arg\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{A}\mathbf{D}\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$
$$\text{subject to } \sum_i \mathbf{D}_{i,j}^2 \leq c, \quad \forall i = 1, \cdots, n.$$

Split the optimization over **D** and **A** in two.

Step 1: Learning the Dictionary

Reduced optimization problem:

$$\arg\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{A}\mathbf{D}\|_F^2$$

$$\text{subject to } \sum_i \mathbf{D}_{i,j}^2 \leq c, \quad \forall i = 1, \cdots, n.$$

Step 1: Learning the Dictionary

Reduced optimization problem:

$$\arg\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{A}\mathbf{D}\|_F^2$$
$$\text{subject to } \sum_i \mathbf{D}_{i,j}^2 \leq c, \quad \forall i = 1, \cdots, n.$$

Introduce Lagrange multipliers:

$$\mathcal{L}(\mathbf{D}, \lambda) = \mathrm{tr}\left((\mathbf{X} - \mathbf{A}\mathbf{D})^T (\mathbf{X} - \mathbf{A}\mathbf{D})\right) + \sum_{j=1}^{n} \lambda_j \left(\sum_i \mathbf{D}_{i,j} - c\right)$$

Step 1: Learning the Dictionary

Reduced optimization problem:

$$\arg\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{AD}\|_F^2$$
$$\text{subject to } \sum_i \mathbf{D}_{i,j}^2 \leq c, \quad \forall i = 1, \cdots, n.$$

Introduce Lagrange multipliers:

$$\mathcal{L}(\mathbf{D}, \lambda) = \mathrm{tr}\left((\mathbf{X} - \mathbf{AD})^T (\mathbf{X} - \mathbf{AD})\right) + \sum_{j=1}^{n} \lambda_j \left(\sum_i \mathbf{D}_{i,j} - c\right)$$

where each $\lambda_j \geq 0$ is a dual variable...

Step 1: Moving to the dual

From the Lagrangian

$$\mathcal{L}\left(\mathbf{D},\lambda\right)=\mathrm{tr}\left(\left(\mathbf{X}-\mathbf{AD}\right)^{T}\left(\mathbf{X}-\mathbf{AD}\right)\right)+\sum_{j=1}^{n}\lambda_{j}\left(\sum_{i}\mathbf{D}_{i,j}^{2}-c\right)$$

Step 1: Moving to the dual

From the Lagrangian

$$\mathcal{L}\left(\mathbf{D}, \lambda\right) = \mathrm{tr}\left(\left(\mathbf{X} - \mathbf{AD}\right)^T \left(\mathbf{X} - \mathbf{AD}\right)\right) + \sum_{j=1}^{n} \lambda_j \left(\sum_i \mathbf{D}_{i,j}^2 - c\right)$$

minimize over **D** to obtain Lagrange dual

$$\mathbf{D}\left(\lambda\right) = \min_{\mathbf{D}} \mathcal{L}\left(\mathbf{D}, \lambda\right) =$$

Step 1: Moving to the dual

From the Lagrangian

$$\mathcal{L}(\mathbf{D}, \lambda) = \mathrm{tr}\left((\mathbf{X} - \mathbf{AD})^T (\mathbf{X} - \mathbf{AD})\right) + \sum_{j=1}^{n} \lambda_j \left(\sum_i \mathbf{D}_{i,j}^2 - c\right)$$

minimize over **D** to obtain Lagrange dual

$$\mathbf{D}(\lambda) = \min_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \lambda) = \mathrm{tr}\left(\mathbf{X}^T\mathbf{X} - \mathbf{XA}^T\left(\mathbf{AA}^T + \Lambda\right)^{-1}\left(\mathbf{XA}^T\right)^T - c\Lambda\right)$$

Step 1: Moving to the dual

From the Lagrangian

$$\mathcal{L}(\mathbf{D}, \lambda) = \mathrm{tr}\left((\mathbf{X} - \mathbf{AD})^T (\mathbf{X} - \mathbf{AD}) \right) + \sum_{j=1}^{n} \lambda_j \left(\sum_i \mathbf{D}_{i,j}^2 - c \right)$$

minimize over **D** to obtain Lagrange dual

$$\mathbf{D}(\lambda) = \min_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \lambda) = \mathrm{tr}\left(\mathbf{X}^T\mathbf{X} - \mathbf{XA}^T \left(\mathbf{AA}^T + \Lambda \right)^{-1} \left(\mathbf{XA}^T \right)^T - c\Lambda \right)$$

- The dual can be optimized using congugate gradient

Step 1: Moving to the dual

From the Lagrangian

$$\mathcal{L}\left(\mathbf{D}, \lambda\right) = \mathrm{tr}\left(\left(\mathbf{X} - \mathbf{A}\mathbf{D}\right)^T \left(\mathbf{X} - \mathbf{A}\mathbf{D}\right)\right) + \sum_{j=1}^{n} \lambda_j \left(\sum_i \mathbf{D}_{i,j}^2 - c\right)$$

minimize over **D** to obtain Lagrange dual

$$\mathbf{D}\left(\lambda\right) = \min_{\mathbf{D}} \mathcal{L}\left(\mathbf{D}, \lambda\right) = \mathrm{tr}\left(\mathbf{X}^T\mathbf{X} - \mathbf{X}\mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T + \Lambda\right)^{-1}\left(\mathbf{X}\mathbf{A}^T\right)^T - c\Lambda\right)$$

- The dual can be optimized using congugate gradient
- Only n, λ values compared to **D** being $n \times k$

Step 1: Dual to the Dictionary

With the optimal Λ, our dictionary is

$$\mathbf{D}^T = \left(\mathbf{A}\mathbf{A}^T + \Lambda\right)^{-1} \left(\mathbf{X}\mathbf{A}^T\right)^T$$

Step 1: Dual to the Dictionary

With the optimal Λ, our dictionary is

$$\mathbf{D}^T = \left(\mathbf{A}\mathbf{A}^T + \Lambda\right)^{-1} \left(\mathbf{X}\mathbf{A}^T\right)^T$$

Key point: Moving to the dual reduces the number of optimization variables, speeding up the optimization.

Step 2: Learning the Sparse Code

With **D** now fixed, optimize for **A**

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{AD}\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$

Step 2: Learning the Sparse Code

With **D** now fixed, optimize for **A**

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{AD}\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$

- Unconstrained, convex quadratic optimization

Step 2: Learning the Sparse Code

With **D** now fixed, optimize for **A**

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{AD}\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$

- Unconstrained, convex quadratic optimization
- Many solvers for this (e.g. interior point methods, in-crowd algorithm, fixed-point continuation)

Step 2: Learning the Sparse Code

With **D** now fixed, optimize for **A**

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{AD}\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$

- Unconstrained, convex quadratic optimization
- Many solvers for this (e.g. interior point methods, in-crowd algorithm, fixed-point continuation)

Note:

- Same problem as the encoding problem.
- Runtime of optimization in the encoding stage?

Speeding up the testing phase

Fair amount of work on speeding up the encoding stage:

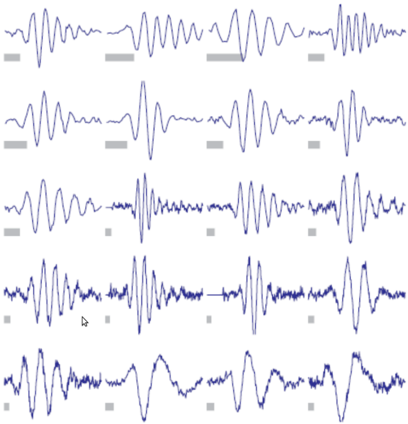
- H. Lee et al., *Efficient sparse coding algorithms*
  `http://ai.stanford.edu/~hllee/nips06-sparsecoding.pdf`
- K. Gregor and Y. LeCun, *Learning Fast Approximations of Sparse Coding*
  `http://yann.lecun.com/exdb/publis/pdf/gregor-icml-10.pdf`
- S. Hawe et al., *Separable Dictionary Learning*
  `http://arxiv.org/pdf/1303.5244v1.pdf`

Outline

1. Introduction: Why Sparse Coding?
2. Sparse Coding: The Basics
3. Adding Prior Knowledge
4. Conclusions

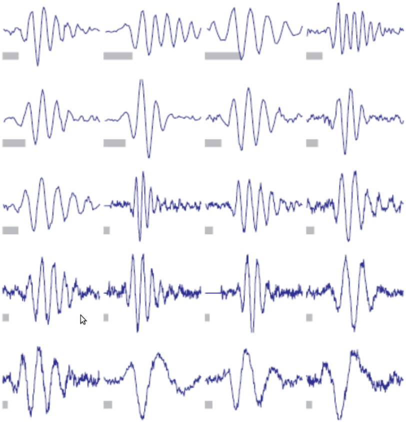
Outline

Relationships between Dictionary atoms



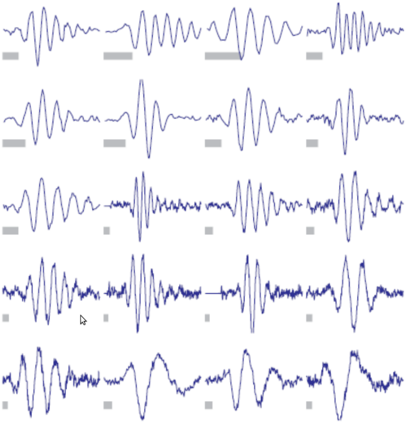
- Dictionaries are over-complete bases

Relationships between Dictionary atoms



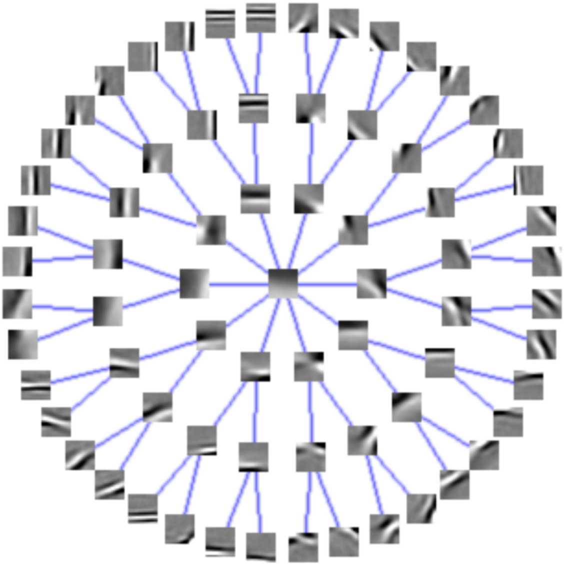
- Dictionaries are over-complete bases
- Dictate relationships between atoms

Relationships between Dictionary atoms

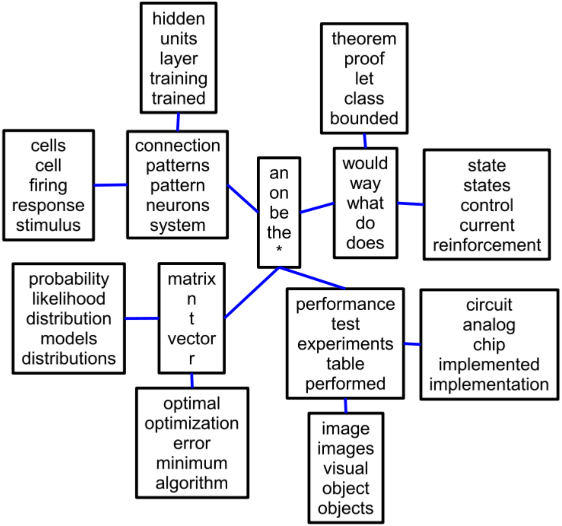


- Dictionaries are over-complete bases
- Dictate relationships between atoms
- Example: Hierarchical dictionaries

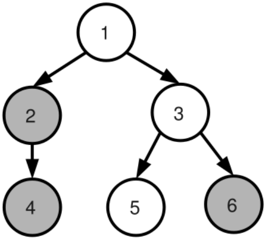
Example: Image Patches



Example: Document Topics



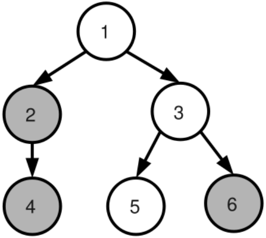
Problem Statement



Goal:

- Have sub-groups of sparse code α all be non-zero (or zero).

Problem Statement



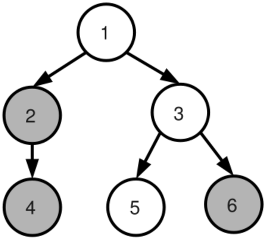
Goal:

- Have sub-groups of sparse code α all be non-zero (or zero).

Hierarchical:

- If a node is non-zero, it's parent must be non-zero
- If a node's parent is zero, the node must be zero

Problem Statement



Goal:

- Have sub-groups of sparse code α all be non-zero (or zero).

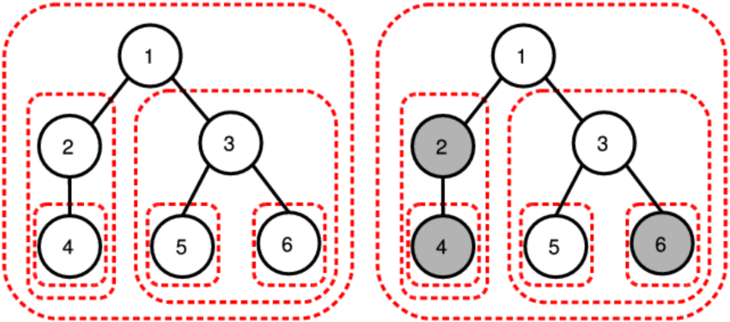
Hierarchical:

- If a node is non-zero, it's parent must be non-zero
- If a node's parent is zero, the node must be zero

Implementation:

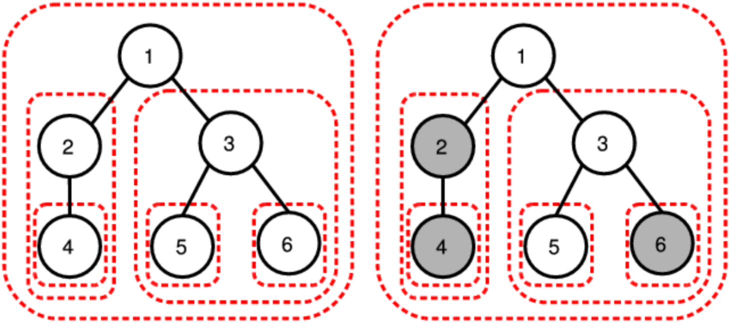
- Change the regularization
- Enforce sparsity differently...

Grouping Code Entries



- Level k included in $k + 1$ groups

Grouping Code Entries



- Level k included in $k+1$ groups
- Add $|\alpha_i|$ to objective function once for each group

Group Regularization

Updated objective function:

$$\arg\min_{\mathbf{D},\{\alpha^j\}} \sum_{j=1}^{m} \left[\|\mathbf{x}^j - \mathbf{D}\alpha^j\|^2 \qquad\qquad \right]$$

Group Regularization

Updated objective function:

$$\arg\min_{\mathbf{D},\{\alpha^j\}} \sum_{j=1}^{m} \left[\|\mathbf{x}^j - \mathbf{D}\alpha^j\|^2 + \beta\Omega\left(\alpha^j\right) \right]$$

Group Regularization

Updated objective function:

$$\arg\min_{\mathbf{D},\{\alpha^j\}} \sum_{j=1}^{m} \left[\|\mathbf{x}^j - \mathbf{D}\alpha^j\|^2 + \beta\Omega\left(\alpha^j\right) \right]$$

where

$$\Omega\left(\alpha\right) = \sum_{g \in \mathcal{P}} w_g \|\alpha_{|g}\|$$

Group Regularization

Updated objective function:

$$\arg\min_{\mathbf{D},\{\alpha^j\}} \sum_{j=1}^{m} \left[\|\mathbf{x}^j - \mathbf{D}\alpha^j\|^2 + \beta\Omega\left(\alpha^j\right) \right]$$

where

$$\Omega\left(\alpha\right) = \sum_{g \in \mathcal{P}} w_g \|\alpha_{|g}\|$$

- $\alpha_{|g}$ are the code values for group g.

Group Regularization

Updated objective function:

$$\arg\min_{\mathbf{D},\{\alpha^j\}} \sum_{j=1}^{m} \left[\|\mathbf{x}^j - \mathbf{D}\alpha^j\|^2 + \beta\Omega\left(\alpha^j\right) \right]$$

where

$$\Omega\left(\alpha\right) = \sum_{g \in \mathcal{P}} w_g \|\alpha_{|g}\|$$

- $\alpha_{|g}$ are the code values for group g.
- w_g weights the enforcement of the hierarchy

Group Regularization

Updated objective function:

$$\arg\min_{\mathbf{D},\{\alpha^j\}} \sum_{j=1}^{m} \left[\|\mathbf{x}^j - \mathbf{D}\alpha^j\|^2 + \beta\Omega\left(\alpha^j\right) \right]$$

where

$$\Omega\left(\alpha\right) = \sum_{g \in \mathcal{P}} w_g \|\alpha_{|g}\|$$

- $\alpha_{|g}$ are the code values for group g.
- w_g weights the enforcement of the hierarchy
- Solve using proximal methods.

Other Examples

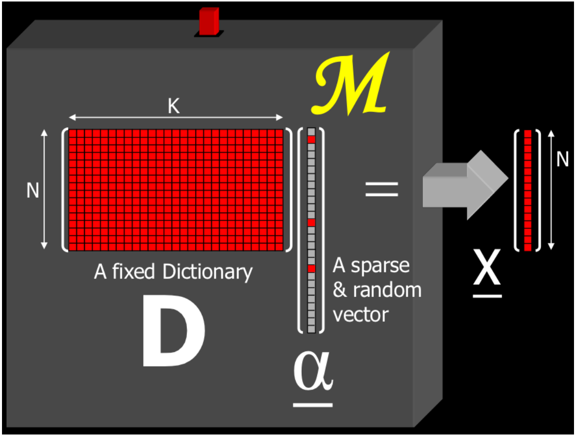
Other examples of structured sparsity:

- M. Stojnic et al., *On the Reconstruction of Block-Sparse Signals With an Optimal Number of Measurements*, `http://dx.doi.org/10.1109/TSP.2009.2020754`
- J. Mairal et al., *Convex and Network Flow Optimization for Structured Sparsity*, `http://jmlr.org/papers/volume12/mairal11a/mairal11a.pdf`

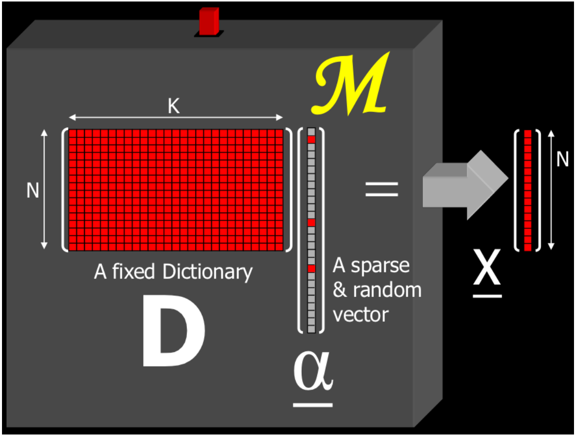
Outline

Outline

Summary

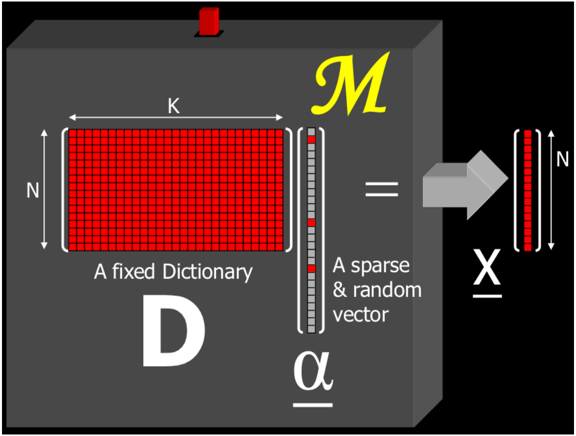


Summary



Two interesting directions:

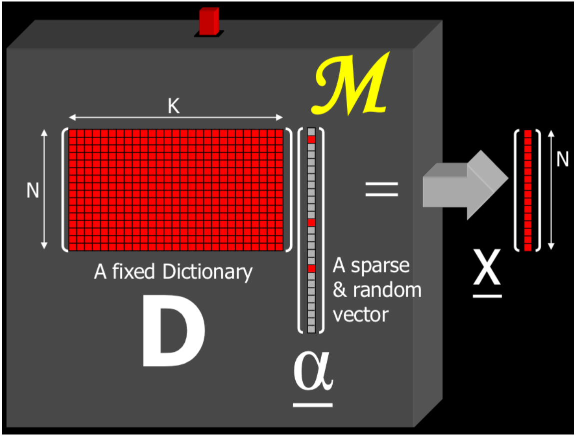
Summary



Two interesting directions:

- Increasing speed of the testing phase

Summary



Two interesting directions:

- Increasing speed of the testing phase
- Optimizing dictionary structure