Sparse Coding: An Overview

Brian Booth

SFU Machine Learning Reading Group

November 12, 2013
The aim of sparse coding

Every column of $D$ is a prototype. Similar to, but more general than, PCA.
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

Natural Images

Learned bases ($\phi_1, \phi_2, \ldots, \phi_{64}$): "Edges"

Test Example

$$\approx 0.8 \times \phi_{36} + 0.3 \times \phi_{42} + 0.5 \times \phi_{63}$$

$$[\alpha_1, \ldots, \alpha_{64}] = [0, \ldots, 0.8, \ldots, 0.3, \ldots, 0.5, \ldots, 0]$$
Sparse Coding in V1

The first stage of visual processing in the brain (V1) does “edge detection.”
Example: Image Denoising
Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold to the United States. Through all these years, and even into the 1900s, others arrived from everywhere: Acadians (Cajuns), Africans, indige-
Sparse Coding and Acoustics

The inner ear (cochlea) also does sparse coding of frequencies.
Inner ear (cochlea) also does sparse coding of frequencies.
Sparse Coding and Natural Language Processing

\[ \phi_{36} \approx 0.7 \times \text{Topic36} + 0.4 \times \text{Topic42} + 0.1 \times \text{Topic63} \]
Introduction: Why Sparse Coding?
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

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

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

\[
x \approx \sum_{i=1}^{n} \alpha_i d_i = \alpha D
\]

Learning:
- Given training data \( x^j, j \in \{1, \cdots, m\} \)
- Learn dictionary \( D \) and sparse code \( \alpha \)
The aim of sparse coding, revisited

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

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

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

Encoding:
- Given test data $\mathbf{x}$, dictionary $\mathbf{D}$
- Learn sparse code $\alpha$
Dictionary learning involves optimizing:

$$\arg \min_{\{d_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|x^j - \sum_{i=1}^{n} \alpha^j_i d_i \|^2$$
Dictionary learning involves optimizing:

$$\arg \min_{\{d_i\}, \{\alpha^j\}} \left( \sum_{j=1}^{m} \left\| x^j - \sum_{i=1}^{n} \alpha^j_i d_i \right\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha^j_i| \right)$$

subject to $\sum_{i=1}^{n} \|d_i\|^2 \leq c$, for all $i = 1, \ldots, n$. In matrix notation:

$$\arg \min_{D, A} \| X - AD \|_F^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha^j_i|$$

Split the optimization over $D$ and $A$ in two.
Learning: The Objective Function

Dictionary learning involves optimizing:

$$\arg \min_{\{d_i\}, \{\alpha^j\}} \sum_{j=1}^{m} \|x^j - \sum_{i=1}^{n} \alpha^j_i d_i \|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha^j_i|$$

subject to $\|d_i\|^2 \leq c$, $\forall i = 1, \cdots, n$. 
Dictionary learning involves optimizing:

$$\arg\min_{\{d_i\},\{\alpha^j\}} \sum_{j=1}^{m} \|x^j - \sum_{i=1}^{n} \alpha^j_i d_i\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha^j_i|$$

subject to $\|d_i\|^2 \leq c, \ \forall i = 1, \ldots, n$.

In matrix notation:

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

subject to $\sum_i D_{i,j}^2 \leq c, \ \forall i = 1, \ldots, n$. 

Split the optimization over $D$ and $A$ in two.
Dictionary learning involves optimizing:

$$\arg \min \{d_i\}, \{\alpha^j\} \sum_{j=1}^{m} \|x^j - \sum_{i=1}^{n} \alpha^j_i d_i\|^2 + \beta \sum_{j=1}^{m} \sum_{i=1}^{n} |\alpha^j_i|$$

subject to $\|d_i\|^2 \leq c, \quad \forall i = 1, \ldots, n.$

In matrix notation:

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

subject to $\sum_i D_{i,j}^2 \leq c, \quad \forall i = 1, \ldots, n.$

Split the optimization over $D$ and $A$ in two.
Step 1: Learning the Dictionary

Reduced optimization problem:

$$\arg \min_{D} \|X - AD\|_{F}^{2}$$

subject to $\sum_{i} D_{i,j}^{2} \leq c$, $\forall i = 1, \cdots, n$. 
Step 1: Learning the Dictionary

Reduced optimization problem:

$$\arg\min_{D} \|X - AD\|_{F}^{2}$$

subject to $\sum_{i,j} D_{i,j}^{2} \leq c, \quad \forall i = 1, \cdots, n.$

Introduce Lagrange multipliers:

$$\mathcal{L} (D, \lambda) = \text{tr} \left( (X - AD)^{T} (X - AD) \right) + \sum_{j=1}^{n} \lambda_{j} \left( \sum_{i} D_{i,j} - c \right)$$
Step 1: Learning the Dictionary

Reduced optimization problem:

$$\arg\min_D \|X - AD\|^2_F$$

subject to $\sum_i D_{i,j}^2 \leq c, \quad \forall i = 1, \ldots, n.$

Introduce Lagrange multipliers:

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

where each $\lambda_j \geq 0$ is a dual variable...
Step 1: Moving to the dual

From the Lagrangian

\[ \mathcal{L}(D, \lambda) = \text{tr} \left( (X - AD)^T (X - AD) \right) + \sum_{j=1}^{n} \lambda_j \left( \sum_i D_{i,j}^2 - c \right) \]
Step 1: Moving to the dual

From the Lagrangian

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

minimize over $D$ to obtain Lagrange dual

$$D(\lambda) = \min_D \mathcal{L}(D, \lambda) =$$
Step 1: Moving to the dual

From the Lagrangian

\[ \mathcal{L}(D, \lambda) = \text{tr} \left( (X - AD)^T (X - AD) \right) + \sum_{j=1}^{n} \lambda_j \left( \sum_i D_{i,j}^2 - c \right) \]

minimize over \( D \) to obtain Lagrange dual

\[ D(\lambda) = \min_D \mathcal{L}(D, \lambda) = \text{tr} \left( X^T X - XA^T \left( AA^T + \Lambda \right)^{-1} \left( XA^T \right)^T - c\Lambda \right) \]
Step 1: Moving to the dual

From the Lagrangian

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

minimize over $D$ to obtain Lagrange dual

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

- The dual can be optimized using conjugate gradient
Step 1: Moving to the dual

From the Lagrangian

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

minimize over $D$ to obtain Lagrange dual

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

- The dual can be optimized using conjugate gradient
- Only $n, \lambda$ values compared to $D$ being $n \times k$
Step 1: Dual to the Dictionary

With the optimal $\Lambda$, our dictionary is

$$D^T = (AA^T + \Lambda)^{-1} (XA^T)^T$$
Step 1: Dual to the Dictionary

With the optimal $\Lambda$, our dictionary is

$$D^T = \left( AA^T + \Lambda \right)^{-1} \left( XA^T \right)^T$$

**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 $A$

$$\arg \min_{A} \|X - AD\|_F^2 + \beta \sum_{i,j} |\alpha_{i,j}|$$
Step 2: Learning the Sparse Code

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

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

- Unconstrained, convex quadratic optimization
Step 2: Learning the Sparse Code

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

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

- 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 $\mathbf{D}$ now fixed, optimize for $\mathbf{A}$

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

- 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*

- K. Gregor and Y. LeCun, *Learning Fast Approximations of Sparse Coding*

- S. Hawe et al., *Separable Dictionary Learning*
Outline

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

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

- hidden units
- layer
- training
- trained
- theorem
- proof
- let
- class
- bounded
- cells
- cell
- firing
- response
- stimulus
- connection patterns
- pattern
- neurons system
- an on be the
- *
- matrix
- n
- t vector
- optimal optimization
- error
- minimum
- algorithm
- performance
- test
- experiments table
- performed
- state
- states
- control
- current
- reinforcement
- circuit
- analog
- chip
- implemented
- implementation
- image
- images
- visual
- object
- objects
Problem Statement

Goal:
- Have sub-groups of sparse code $\alpha$ all be non-zero (or zero).
Goal:
- Have sub-groups of sparse code $\alpha$ all be non-zero (or zero).

Hierarchical:
- If a node is non-zero, its 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 $\alpha$ 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
Updated objective function:

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

Updated objective function:

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

Updated objective function:

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

where

$$\Omega (\alpha) = \sum_{g \in \mathcal{P}} w_g \|\alpha|_g\|$$
Group Regularization

Updated objective function:

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

where

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

- $\alpha_{|g}$ 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:

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

where

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

- $\alpha|_g$ are the code values for group $g$.
- $w_g$ weights the enforcement of the hierarchy
- Solve using proximal methods.
Other examples 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

Outline

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

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

- Increasing speed of the testing phase
- Optimizing dictionary structure
Two interesting directions:

- Increasing speed of the testing phase
- Optimizing dictionary structure
Two interesting directions:

- Increasing speed of the testing phase
Summary

Two interesting directions:

- Increasing speed of the testing phase
- Optimizing dictionary structure