When Dictionary Learning Meets Classification

Bufford, Teresa\textsuperscript{1}  Chen, Yuxin\textsuperscript{2}
Horning, Mitchell\textsuperscript{3}  Shee, Liberty\textsuperscript{1}

Mentor: Professor Yohann Tendero

\textsuperscript{1}UCLA  \textsuperscript{2}Dalhousie University  \textsuperscript{3}Harvey Mudd College

August 9, 2013
Outline

1. The Method: Dictionary Learning

2. Experiments and Results
   
   2.1 Supervised Dictionary Learning
   
   2.2 Unsupervised Dictionary Learning
   
   2.3 Robustness w.r.t. Noise

3. Conclusion
Dictionary Learning

**GOAL:** Classify discrete image signals \( x \in \mathbb{R}^n \).

The Dictionary, \( D \in \mathbb{R}^{n \times K} \)

\[
x \approx D\alpha = \begin{bmatrix} \text{atom}_1 & \cdots & \text{atom}_K \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_K \end{bmatrix}
\]

- Each dictionary can be represented as a matrix, where each column is an atom \( \in \mathbb{R}^n \), learned from a set of training data.
- A signal \( x \in \mathbb{R}^n \) can be approximated by a linear combination of atoms in a dictionary, represented by \( \alpha \in \mathbb{R}^K \).
- We seek a sparse signal representation:

\[
\arg \min_{\alpha \in \mathbb{R}^K} \| x - D\alpha \|_2^2 + \lambda \| \alpha \|_1. \tag{1}
\]

Supervised Dictionary Learning Algorithm

Illustrating Example

Training image signals: \( x_1, x_2, x_3, x_4 \) \( X = [x_1 \ x_2 \ x_3 \ x_4] \)

Classes: 0, 1

Group training images according to their labels:

class 0 \( x_1 \ x_4 \)

class 1 \( x_2 \ x_3 \)

Use training images with label \( i \) to train dictionary \( i \):

class 0 \( x_1 \ x_4 \) \( \rightarrow \) \( D_0 \)

class 1 \( x_2 \ x_3 \) \( \rightarrow \) \( D_1 \)
Signal Classification

Illustrating Example

Test image signals: $y_1, y_2, y_3$ \hspace{1cm} $Y = [y_1 \ y_2 \ y_3]$

Dictionaries: $D_0, D_1$

Take one test image. Compute its optimal sparse representation $\alpha$ for each dictionary. Use $\alpha$ to compute energy:

$$E_i(y_1) = \min_{\alpha} \|y_1 - D_i\alpha\|_2^2 + \lambda \|\alpha\|_1 \quad \rightarrow \quad \begin{bmatrix} E_0(y_1) \\ E_1(y_1) \end{bmatrix}$$

Do this for each test signal:

$$E(Y) = \begin{bmatrix} E_0(y_1) & E_0(y_2) & E_0(y_3) \\ E_1(y_1) & E_1(y_2) & E_1(y_3) \end{bmatrix}$$

Classify the test signal as class $i^* = \arg \min E_i(y)$. For example:

$$E(Y) = \begin{bmatrix} 5 & 12 & 8 \\ 24 & 6 & 4 \end{bmatrix} \quad \rightarrow \quad \text{class 0} \quad \begin{array}{c} y_1 \\ y_2 \ y_3 \end{array} \quad \text{class 1}$$
### Supervised Dictionary Learning Results

#### Table: Supervised Results

<table>
<thead>
<tr>
<th>Cluster Type</th>
<th>Centered &amp; Normalized</th>
<th>Digits</th>
<th>$K$</th>
<th>misclassification rate average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>False</td>
<td>{0,...,4}</td>
<td>500</td>
<td>1.3431%</td>
</tr>
<tr>
<td>Supervised</td>
<td>True</td>
<td>{0,...,4}</td>
<td>500</td>
<td>0.5449%</td>
</tr>
<tr>
<td>Supervised</td>
<td>False</td>
<td>{0,...,4}</td>
<td>800</td>
<td>0.7784%</td>
</tr>
<tr>
<td>Supervised</td>
<td>True</td>
<td>{0,...,4}</td>
<td>800</td>
<td>0.3892%</td>
</tr>
<tr>
<td>Supervised</td>
<td>False</td>
<td>{0,...,9}</td>
<td>800</td>
<td>3.1100%</td>
</tr>
<tr>
<td>Supervised</td>
<td>True</td>
<td>{0,...,9}</td>
<td>800</td>
<td>1.6800%</td>
</tr>
</tbody>
</table>

Error rate for digits \{0,...,9\} and $K = 800$ from [1] is 1.26%.

---

Supervised Results, ctd.

Here is a confusion matrix for the supervised, centered, normalized case with $k = 800$ and digits $\{0, \ldots, 4\}$.

Element $c_{ij} \in C$ is the number of times that an image of digit $i$ was classified as digit $j$.

$$C = \begin{pmatrix}
0 & 1 & 2 & 3 & 4 \\
0 & 978 & 0 & 1 & 1 & 0 \\
1 & 0 & 1132 & 2 & 1 & 0 \\
2 & 5 & 2 & 1023 & 0 & 2 \\
3 & 0 & 0 & 3 & 1006 & 1 \\
4 & 1 & 0 & 1 & 0 & 980
\end{pmatrix}$$
Unsupervised Dictionary Learning Algorithm
(Spectral Clustering)

Illustrating Example

Training image signals: \(x_1, x_2, x_3, x_4\) \(X = [x_1 \ x_2 \ x_3 \ x_4]\)
Classes: 0, 1

Train a dictionary \(D\) from all training images:

\[
\begin{bmatrix}
 x_1 & x_2 & x_3 & x_4 \\
\end{bmatrix} \rightarrow D =
\begin{bmatrix}
 atom_1 & atom_2 & atom_3 \\
\end{bmatrix}
\]

Reminder: The number of atoms in the dictionary is a parameter.

For each image, compute optimal sparse representation \(\alpha\) w.r.t. \(D\): 

\[
A =
\begin{bmatrix}
 atom_1 & \ | & \ | & \mid \\
 atom_2 & \alpha_1 & \alpha_2 & \alpha_3 \mid \\
 atom_3 & \ | & \ | & \mid \\
\end{bmatrix}
\]

Reminder: \(\alpha\) is a linear combination of atoms in \(D\).
Illustrating Example

Training image signals: $x_1, x_2, x_3, x_4 \quad X = [x_1 \ x_2 \ x_3 \ x_4]$

Classes: 0, 1

Construct a similarity matrix $S_1 = |A|^t |A|$.
Perform spectral clustering on the graph $G_1 = \{X, S_1\}$:

$G_1 = \{X, S_1\} \xrightarrow{\text{spectral\ clustering}}$ $\begin{align*}
\text{class 0} & \quad x_1 \ x_4 \\
\text{class 1} & \quad x_2 \ x_3
\end{align*}$

Use signal clusters to train initial dictionaries:

$\begin{align*}
\text{class 0} & \quad x_1 \ x_4 \quad \rightarrow \quad D_0 \\
\text{class 1} & \quad x_2 \ x_3 \quad \rightarrow \quad D_1
\end{align*}$
### Refining Dictionaries

**Repeat:**

1. **Classify the training images using current dictionaries:**
   
   \[ x_1, x_2, x_3, x_4 \] classify with \( D_0, D_1 \)

   - **class 0:** \( x_4 \)
   - **class 1:** \( x_1, x_2, x_3 \)

2. **Use these classifications to train new, refined dictionaries for each cluster:**
   
   - **class 0:** \( D_{0,\text{new}} \)
   - **class 1:** \( D_{1,\text{new}} \)
Unsupervised Dictionary Learning
Spectral Clustering Results

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Centered &amp; Normalized</th>
<th>Digits</th>
<th>$K$</th>
<th>misclassification rate average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atoms</td>
<td>True</td>
<td>{0, ..., 4}</td>
<td>500</td>
<td>24.8881%</td>
</tr>
<tr>
<td>Atoms</td>
<td>True</td>
<td>{0, ..., 4}</td>
<td>800</td>
<td>27.1843%</td>
</tr>
<tr>
<td>Signals</td>
<td>True</td>
<td>{0, ..., 4}</td>
<td>500</td>
<td>27.4372%</td>
</tr>
<tr>
<td>Signals</td>
<td>True</td>
<td>{0, ..., 4}</td>
<td>800</td>
<td>29.5777%</td>
</tr>
</tbody>
</table>

Misclassification rate for digits \{0, ..., 4\} and $K = 500$ from [1] is 1.44%.

Why do our results differ from Sprechmann, et al.’s?

Hypothesis

The problems lie in the author’s choice of similarity measures, $S = |A|^t|A|$.

Possible Problem: Normalization ($A$’s cols not of constant $l_2$ norm)

Illustration of Problem

- Assume that $A$ contains only positive entries. Then $S = |A|^t|A| = A^tA$ and the $ij^{th}$ entry of $S$ is

  $< \alpha_i, \alpha_j > = \|\alpha_i\|\|\alpha_j\| \cos(\alpha_i, \alpha_j)$.

- Nearly orthogonal vectors are more similar than co-linear ones if the products of their norms are big enough.

“more similar”
**Attempted Solution**

Normalized each column of $A$ before computing $S$. However, this caused little change in the results, due to the $l_2$ norms of $A$’s columns being almost constant.

**Figure**: The normalized histogram of the $l_2$ norm of the columns of $A$.

**Note**: The $l_2$ norms of most columns in $A$ are in $[3, 4]$. 
Possible Problem: Absolute Value (use $|A|^t|A|$ instead of $|A^tA|$)

Illustration of Problem
$a_1 = (1, -1)^t$ and $a_2 = (1, 1)^t$, which are orthogonal vectors, become co-linear when the entries are replaced by their absolute values.

Attempted Solution
In the experiments, changing $|A||A|^t$ to $|AA^t|$ does not significantly improve the results, because among all the entries of $A$ associated with MNIST data, only $\approx 13.5\%$ are negative.
In the last part of Sprechmann et al, the authors state:

“We observed that best results are obtained for all the experiments when the initial dictionaries in the learning stage are constructed by randomly selecting signals from the training set. If the size of the dictionary compared to the dimension of the data is small, [it] is better to first partition the dataset (using for example Euclidean k-means) in order to obtain a more representative sample.”

Algorithm

Use $k$-means to cluster the training signals, instead of spectral clustering.
Unsupervised Dictionary Learning (kmeans)

Illustrating Example

Training image signals: $x_1, x_2, x_3, x_4$  \[ X = [x_1 \ x_2 \ x_3 \ x_4] \]

Classes: 0, 1

Use kmeans to cluster training images:

\[ x_1, x_2, x_3, x_4 \xrightarrow{kmeans} \]

class 0 \[ \begin{array}{c} x_1 \\ x_4 \end{array} \]

class 1 \[ \begin{array}{c} x_2 \\ x_3 \end{array} \]

Note: Must center and normalize after clustering.

Use clusters to train dictionaries:

class 0 \[ \begin{array}{c} x_1 \\ x_4 \end{array} \xrightarrow{} D_0 \]

class 1 \[ \begin{array}{c} x_2 \\ x_3 \end{array} \xrightarrow{} D_1 \]

Refinement process is same as spectral clustering case.
Unsupervised Dictionary Learning K-means Results

Table: Unsupervised Results (kmeans), without refinement

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Centered &amp; Normalized</th>
<th>Digits</th>
<th>$K$</th>
<th>iter</th>
<th>misclassification rate average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signals</td>
<td>True</td>
<td>${0,...,4}$</td>
<td>500</td>
<td>2</td>
<td>7.1025%</td>
</tr>
</tbody>
</table>

Table: Unsupervised Results (kmeans), with refinement

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Centered &amp; Normalized</th>
<th>Digits</th>
<th>$K$</th>
<th>iter</th>
<th>misclassification rate average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signals</td>
<td>True</td>
<td>${0,...,4}$</td>
<td>500</td>
<td>2</td>
<td>1.1286%</td>
</tr>
<tr>
<td>Signals</td>
<td>True</td>
<td>${0,...,4}$</td>
<td>500</td>
<td>20</td>
<td>0.5449%</td>
</tr>
</tbody>
</table>

Misclassification rate for digits $\{0, ..., 4\}$ and $K = 500$ from [1] is 1.44%.

---

Unsupervised Results, ctd.

Here is a confusion matrix for the $k$-means, unsupervised, centered, normalized case with $k = 500$, iter $= 20$, and digits $\{0, \ldots, 4\}$.

Element $c_{ij} \in C$ is the number of times that an image of digit $i$ was classified as digit $j$.

$$
C = \begin{pmatrix}
0 & 979 & 0 & 1 & 0 & 0 \\
1 & 0 & 1129 & 4 & 1 & 1 \\
2 & 4 & 2 & 1023 & 2 & 1 \\
3 & 0 & 0 & 3 & 1006 & 1 \\
4 & 0 & 1 & 3 & 0 & 978
\end{pmatrix}
$$
Classification of Noisy Images

Goal: To analyze the robustness of our method with respect to noise in images

Questions to consider:

- How does adding noise affect misclassification rates?
- How does centering and normalizing the data after adding noise affect the results?
Results: Classifying Pure Gaussian Noise Classification Rates for Each Digit

Train dictionaries on centered and normalized data

Train dictionaries on not centered and not normalized data

noise variance $= 0.5$

noise variance $= 0.5$
MNIST Images with Gaussian Noise Added

Original MNIST Image

Gaussian Noise variance 0.1

Gaussian Noise variance 0.5

Gaussian Noise variance 1.0
Results: Adding Gaussian Noise to MNIST
Noise in test images only

Centered and Normalized
test noise variance = 0.5
Misclassification rate 8.21%

Not Centered and Not Normalized
test noise variance = 0.5
Misclassification rate 87.05%
Results: Adding Gaussian Noise to MNIST
Same noise variance for test & training images

<table>
<thead>
<tr>
<th>Noise variance</th>
<th>0.1</th>
<th>0.5</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centered &amp; Normalized</td>
<td>3.2%</td>
<td>17.33%</td>
<td>38.92%</td>
</tr>
<tr>
<td>Not Centered &amp; Not Normalized</td>
<td>11.08%</td>
<td>57.56%</td>
<td>75.87%</td>
</tr>
</tbody>
</table>
Conclusions:

• Unable to reproduce results reported by Sprechmann

• Method of k-means produces better results than spectral clustering

• Centering and normalizing data consistently improves results

• Method is robust to added noise
Thanks for your attention!
Unsupervised Dictionary Learning Algorithm 2 (with split initialization)

Step 1 Train a dictionary $D_0 \in \mathbb{R}^{n \times k}$ from all training images.

Step 2 Obtain matrix $A_0$ and similarity matrix $S_0 = |A_0||A_0|^t$ as before.

Step 3 Apply spectral clustering to dictionary $D_0$ to split its atoms into $D_1$ and $D_2$.

Step 4 Obtain matrices $A_1$, $S_1$, $A_2$, and $S_2$ for dictionaries $D_1$ and $D_2$.

Step 5 Apply spectral clustering to dictionaries $D_1$ and $D_2$ to split each into two clusters. Choose the segmentation that results in the lower energy and keep this one. Return the other dictionary to its original form. Now you have three dictionaries.

Step 6 Repeat this process so that after each iteration the number of dictionaries increases by one. Stop when the desired number of clusters is reached.
Semisupervised Dictionary Learning Algorithm

**Step 1** Use training images, a percentage of which have known labels. The remaining percentage are randomly labeled.

**Step 2** Train dictionaries \( \{D_0, \ldots, D_{N-1}\} \) independently.

**Step 3** Classify the training images using current dictionaries. Then use these classifications to train new, refined dictionaries for each cluster.
## Semisupervised Dictionary Learning Results

Dictionaries for digits $\{0, \ldots, 4\}$.

<table>
<thead>
<tr>
<th>Cluster Type</th>
<th>Centered &amp; Normalized</th>
<th>$K$</th>
<th>perturb. percent</th>
<th>misclassification rate average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semisupervised</td>
<td>True</td>
<td>200</td>
<td>40</td>
<td>0.9730%</td>
</tr>
<tr>
<td>Semisupervised</td>
<td>True</td>
<td>200</td>
<td>70</td>
<td>0.8951%</td>
</tr>
<tr>
<td>Semisupervised</td>
<td>True</td>
<td>200</td>
<td>90</td>
<td>1.4205%</td>
</tr>
</tbody>
</table>
Figure: Misclassification rate average per refinement iteration. Left to right: 40%, 70%, 90% perturbation.

Figure: Energy per refinement iteration. Left to right: 40%, 70%, 90% perturbation.
**Figure**: Misclassification rate average per refinement iteration. Left to right: 40%, 70%, 90% perturbation.

**Figure**: Number of training images whose classification changed per refinement iteration. Left to right: 40%, 70%, 90% perturbation.
Unsupervised - Atoms (change over refinement iterations)

Figure: Unsupervised dictionary learning ($K = 500$), spectral clustering of centered and normalized signals by atoms. Left to right: (1) energy, (2) misclassification rate average, (3) number of training images whose classification changed per refinement iteration.
Unsupervised - Signals (change over refinement iterations)

**Figure:** Unsupervised dictionary learning ($K = 500$), spectral clustering of centered and normalized signals by signals. Left to right: (1) energy, (2) misclassification rate average, (3) number of training images whose classification changed per refinement iteration.
Unsupervised - kmeans (change over refinement iterations)

Figure: Unsupervised dictionary learning ($K = 500$), kmeans clustering of centered and normalized signals. Left to right: (1) energy, (2) misclassification rate average, (3) number of training images whose classification changed per refinement iteration.
Unsupervised - kmeans (change over refinement iterations)

**Figure**: Unsupervised dictionary learning ($K = 500$), kmeans clustering of centered and normalized signals. Left: 2 dictionary learning refinements. Right: 20 dictionary learning refinements.