Analyzing Task Driven Learning Algorithms
Final Presentation

Mike Pekala

May 1, 2012

Advisor: Prof. Doron Levy (dlevy at math.umd.edu)
UMD Dept. of Mathematics & Center for Scientific Computation and Mathematical Modeling (CSCAMM)
Overview

Project Overview

Existing Algorithm Implementation/Validation

- Sparse Reconstruction
  - Least Angle Regression (LARS) [Efron et al., 2004]
  - Feature-Sign [Lee et al., 2007]
  - Non-negative and incremental Cholesky variants

- Dictionary Learning
  - Task-Driven Dictionary Learning (TDDL) [Mairal et al., 2010]

Application/Analysis to New (Publicly Available) Datasets

- Hyperspectral Imagery
  - USGS Hyperspectral Library [Clark et al., 2007]
x = Φα
Penalized Least Squares

Recall the Lasso: given $\Phi = [\phi_1, \ldots, \phi_p] \in \mathbb{R}^{m \times p}, t \in \mathbb{R}_+$, solve:

$$\min_{\alpha} \| x - \Phi \alpha \|^2_2 \quad s.t. \| \alpha \|_1 \leq t$$

which has an equivalent unconstrained formulation:

$$\min_{\alpha} \| x - \Phi \alpha \|^2_2 + \lambda \| \alpha \|_1$$

for some scalar $\lambda \geq 0$. The $L_1$ penalty improves upon OLS by introducing parsimony (feature selection) and regularization (improved generality).

Many ways to solve this problem, e.g.

1. Directly, via convex optimization (can be expensive)
2. Iterative techniques
   - Forward selection (“matching pursuit”), forward stagewise, others.
   - Least Angle Regression (LARS) [Efron et al., 2004]
   - Feature-Sign [Lee et al., 2007]
LARS Properties

Full details in [Efron et al., 2004]

Why is it good?

- Less aggressive than some greedy techniques; less likely to eliminate useful predictors when predictors are correlated.
- More efficient than Forward Selection, which can take thousands of tiny steps towards a final model.

Some Properties

- (Theorem 1) Assuming covariates added/removed one at a time from active set, complete LARS solution path yields all Lasso solutions.
- (Sec. 3.1) With a change to the covariate selection rule, LARS can be modified to solve the Positive Lasso problem.
- (Sec. 7) The cost of LARS is comparable to that of a least squares fit on $m$ variables. The LARS sequence incrementally generates a Cholesky factorization of $\Phi^T \Phi$ in a very specific order.
(2.22) Successive LARS estimates $\hat{\mu}_k$ always approach but never reach the OLS estimate $\bar{x}_k$ (except maybe on the final iteration).

$\hat{\mu}_2$ approaches OLS solution $\bar{x}_2$
LARS Implementation/Validation

Diabetes Validation Test: Coefficients

\[ n = 10, m = 442; \text{ Matches Figure 1 in [Efron et al., 2004]} \]
Also validated by comparing orthogonal designs with theoretical result.
Feature-Sign Properties

Full details in [Lee et al., 2007]

Why is it good?

- Very efficient; reported performance gains over LARS.
- Can be initialized with arbitrary starting coefficients.
- Simple to implement.
- One half of a two-part algorithm for matrix factorization.

Some Properties

- Tries to search for, or “guess”, signs of coefficients. Knowing signs reduces LASSO to an unconstrained quadratic program (QP) with closed form solution.
- Iteratively refines these sign guesses; involves an intermediate line search.
- Objective function strictly decreases.
Implemented nonnegative extension. Performance hit (at least with my implementation) as the unconstrained QP becomes a constrained QP. Solved using Matlab’s quadprog().

Validated by comparing results with LARS.
$x = \Phi \alpha$
Dictionary Learning for Sparse Reconstruction

Following the notation/development of [Mairal et al., 2010].

- Given: training data set of signals $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ in $\mathbb{R}^{m \times n}$
- Goal: design a dictionary $\mathbf{\Phi}$ in $\mathbb{R}^{m \times p}$ (possible for $p > m$, i.e. an overcomplete dictionary) by minimizing the empirical cost function

$$g_n(D) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell_u(x_i, D)$$

where $\ell_u$, the unsupervised loss function, is small when $\Phi$ is “good” at representing $x_i$ sparsely. In [Mairal et al., 2010], the authors use the elastic-net formulation:

$$\ell_u(x, D) \triangleq \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} ||x - D\alpha||_2^2 + \lambda_1 ||\alpha||_1 + \frac{\lambda_2}{2} ||\alpha||_2^2$$

(1)
Dictionary Learning for Sparse Reconstruction

- To prevent artificially improving $\ell_u$ by arbitrarily scaling $D$, one typically constrains the set of permissible dictionaries:

$$\mathcal{D} \triangleq \{ D \in \mathbb{R}^{m \times p} \text{ s.t. } \forall j \in \{1, \ldots, p\}, \|d_j\|_2 \leq 1 \}$$

- Optimizing the empirical cost $g_n$ can be very expensive when the training set is large (as is often the case in dictionary learning problems). However, in reality, one usually wants to minimize the expected loss:

$$g(D) \triangleq \mathbb{E}_x [\ell_u(x, D)] = \lim_{n \to \infty} g_n(D) \text{ a.s.}$$

(where expectation is taken with respect to the unknown distribution of data objects $p(x)$) In these cases, online stochastic techniques have been shown to work well [Mairal et al., 2009].
Consider the classification task:

- Given: a fixed dictionary $\mathbf{D}$, an observation $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^m$ and a sparse representation of the observation $\mathbf{x} \approx \mathbf{\alpha}^*(\mathbf{x}, \mathbf{D})$
- Goal: identify the associated label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is a finite set of labels (would be a subset of $\mathbb{R}^q$ for regression)

Assume $\mathbf{D}$ is fixed and $\mathbf{\alpha}^*(\mathbf{x}, \mathbf{D})$ will be used as the features for predicting $y$. The classification problem is to learn the model parameters $\mathbf{W}$ by solving:

$$
\min_{\mathbf{W} \in \mathcal{W}} f(\mathbf{W}) + \frac{\nu}{2} \| \mathbf{W} \|_F^2
$$

where

$$
f(\mathbf{W}) \triangleq \mathbb{E}_{y, \mathbf{x}} [\ell_s(y, \mathbf{W}, \mathbf{\alpha}^*(\mathbf{x}, \mathbf{D}))]
$$

and $\ell_s$ is a convex loss function (e.g. logistic) adapted to the supervised learning problem.
Task Driven Dictionary Learning for Classification

Now, we wish to jointly learn $D, W$:

$$\min_{D \in \mathcal{D}, W \in \mathcal{W}} f(D, W) + \frac{\nu}{2} \|W\|_F^2$$

where

$$f(D, W) \triangleq \mathbb{E}_{y,x} [\ell_s(y, W, \alpha^*(x, D))]$$

Example:

- Binary classification: $\mathcal{Y} = \{-1, +1\}$
- Linear model: $w \in \mathbb{R}^p$
- Prediction: $\text{sign}(w^T \alpha^*(x, D))$
- Logistic loss: $\ell_s = \log \left(1 + e^{-yw^T \alpha^*}\right)$

$$\min_{D \in \mathcal{D}, w \in \mathbb{R}^p} \mathbb{E}_{y,x} \left[\log \left(1 + e^{-yw^T \alpha^*(x, D)}\right)\right] + \frac{\nu}{2} \|w\|_2^2$$
Solving the Problem

Stochastic gradient descent is often used to minimize functions whose gradients are expectations. The authors of [Mairal et al., 2010] show that, under suitable conditions, equation (2) is differentiable on $\mathcal{D} \times \mathcal{W}$, and that,

\[
\nabla_W f(D, W) = \mathbb{E}_{y,x} [\nabla_W \ell_s(y, w, \alpha^*)]
\]

\[
\nabla_D f(D, W) = \mathbb{E}_{y,x} \left[ -D \beta^* \alpha^{*T} + (x - D \alpha^*) \beta^{*T} \right]
\]

where $\beta^* \in \mathbb{R}^p$ is defined by the properties:

\[
\beta^* \Lambda^C = 0 \text{ and } \beta^* \Lambda = (D^T_D \Lambda + \lambda_2 I)^{-1} \nabla_{\alpha_{\Lambda}} \ell_s(y, W, \alpha^*)
\]

and $\Lambda$ are the indices of the nonzero coefficients of $\alpha^*(x, D)$. 
Algorithm: SGD for task-driven dictionary learning

[112x266]Dictionary Learning

Algorithm: SGD for task-driven dictionary learning

[112x266]Dictionary Learning

[Mairal et al., 2010]

**Input:** $p(y, x)$ (a way to draw samples i.i.d. from $p$), $\lambda_1, \lambda_2, \nu \in \mathbb{R}$ (regularization parameters), $D \in D_0$ (initial dictionary), $W_0 \in \mathcal{W}$ (initial model), $T$ (num. iterations), $t_0, \rho \in \mathbb{R}$ (learning rate parameters)

1. for $t = 1$ to $T$ do
2. Draw $(y_t, x_t)$ from $p(y, x)$ (mini-batch of size 200)
3. Compute $\alpha^*$ via sparse coding (LARS, Feature-Sign)
4. Determine active set $\Lambda$ and $\beta^*$
5. Update learning rate $\rho_t$
6. Take projected gradient descent step
7. end
**TDDL Implementation/Validation**

Matched experimental results on the USPS [Hastie et al., 2009] data set with those reported in [Mairal et al., 2010]

<table>
<thead>
<tr>
<th>Digit</th>
<th>$\rho$</th>
<th>$\lambda$</th>
<th># in $D_0$</th>
<th>Runtime (h)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>.150</td>
<td>5</td>
<td>8.2</td>
<td>.926</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>.225</td>
<td>7</td>
<td>7.1</td>
<td>.990</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>.225</td>
<td>7</td>
<td>6.8</td>
<td>.972</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>.225</td>
<td>7</td>
<td>7.4</td>
<td>.968</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>.225</td>
<td>4</td>
<td>7.6</td>
<td>.971</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>.225</td>
<td>4</td>
<td>7.2</td>
<td>.972</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>.225</td>
<td>2</td>
<td>7.5</td>
<td>.969</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>.175</td>
<td>5</td>
<td>7.9</td>
<td>.983</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>.200</td>
<td>3</td>
<td>8.5</td>
<td>.951</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>.200</td>
<td>3</td>
<td>8.1</td>
<td>.969</td>
</tr>
</tbody>
</table>

mean reported

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>.967</td>
</tr>
<tr>
<td>reported</td>
<td>.964</td>
</tr>
</tbody>
</table>
Topic: Hyperspectral Imaging

\[(x, y, \lambda)\]

Smith Island, Near IR

\[
\begin{array}{cccccccc}
0.6 & 0.8 & 1 & 1.2 & 1.4 & 1.6 & 1.8 & 2 \\
-0.15 & -0.1 & -0.05 & 0 & 0.05 & 0.1 \\
\end{array}
\]

wavelength (\(\mu m\))

\[
\begin{array}{cccccccc}
0.6 & 0.8 & 1 & 1.4 & 1.6 & 1.8 & 2.2 & 2.4 \\
-0.15 & -0.1 & -0.05 & 0.05 & 0.1 \\
\end{array}
\]
Material heterogeneity and environmental interference mean that one never measures “pure” pixels/spectra. Instead, “spectral unmixing” is often used to determine the material present at some pixel $\mathbf{x} \in \mathbb{R}^m$, 

$$
\mathbf{x} = \sum_{k=1}^{n} \phi_k \alpha_k + \epsilon
$$

where $\{\phi_k\}$ is a spectral library, $\{\alpha_k\}$ are scalar mixture coefficients and $\epsilon$ is noise. Recent results suggest sparse coding may apply to the spectral unmixing problem; also to infer HSI-resolution data from lower resolution measurements [Charles et al., 2011].
Mixture Element Detection

- Original plan: analysis on single pixel classification problems for objects in scene comprised of $\geq 1$ pixel
  - Problem very easy in some cases (baseline algorithms have no difficulty)
  - In the opinion of one HSI expert, a more relevant problem today is sub-pixel detection

- Modified plan: “mixture element detection” problem
  - Select a single spectral signature as the target
  - Generate mixtures of $s$ spectral “ingredients”; some containing target signature, some without
  - Binary classification problem: identify mixtures containing target signatures

- Used TDDL + nonnegative Feature-Sign solver; various baselines for comparison
Urban scene in Texas
- 307× 307 pixels
- 210 spectral bands (162 valid)
- wavelengths: 412-2390 nm
- radiance data
- no “standard” ground truth
- freely available
Manual Ground Truth

[Image of an aerial view with labeled areas and a corresponding graph of raw spectra (averages)].
Mixture Element Detection, 1-vs-all Classification

### Classification Accuracy

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>kNN1</th>
<th>kNN3</th>
<th>LR-SC</th>
<th>kNN1-SC</th>
<th>kNN3-SC</th>
<th>TD-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>84.0</td>
<td>83.2</td>
<td>82.0</td>
<td><strong>86.8</strong></td>
<td>77.0</td>
<td>78.4</td>
<td>82.4</td>
</tr>
<tr>
<td>M2</td>
<td>82.6</td>
<td>79.2</td>
<td>76.6</td>
<td>75.8</td>
<td>72.8</td>
<td>77.4</td>
<td>81.2</td>
</tr>
<tr>
<td>M3</td>
<td>76.8</td>
<td>74.6</td>
<td>75.0</td>
<td>74.0</td>
<td>75.2</td>
<td>69.0</td>
<td><strong>81.4</strong></td>
</tr>
<tr>
<td>M4</td>
<td>86.4</td>
<td>87.8</td>
<td>82.6</td>
<td>84.0</td>
<td>77.6</td>
<td>78.6</td>
<td><strong>88.2</strong></td>
</tr>
</tbody>
</table>

### Parameter Value

- # Target Mixtures: 500
- # Clutter Mixtures: 500
- % Training: 50
- # Ingredients: 3
- Min. % Target: 5
- Max. % Target: 25
- Noise Variance: 0
- TDDL Iterations: 10000

- Not clear any one approach significantly better
- Only 4 total ingredients in library, signatures fairly distinct
USGS Spectral Library
[Clark et al., 2007]

- Freely available library of 1365 different spectra (minerals, mixtures, coatings, volatiles, man-made, vegetation)
- Focus on a subset of 44 spectra from the vegetation category ($\sim 0.3 - 2.5 \mu m$, $\sim 1200$ valid wavelengths)
### Mixture Element Detection, 1-vs-all Classification

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>kNN1</th>
<th>kNN3</th>
<th>LR-SC</th>
<th>kNN1-SC</th>
<th>kNN3-SC</th>
<th>TD-50</th>
<th>TD-300</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>60.2</td>
<td>63.4</td>
<td>65.8</td>
<td>67.7</td>
<td>57.4</td>
<td>61.6</td>
<td><strong>70.2</strong></td>
<td><strong>70.2</strong></td>
</tr>
<tr>
<td>M2</td>
<td>49.2</td>
<td>61.6</td>
<td>63.2</td>
<td>56.4</td>
<td>52.8</td>
<td>60.0</td>
<td>59.4</td>
<td>60.8</td>
</tr>
<tr>
<td>M3</td>
<td>52.8</td>
<td>56.2</td>
<td>54.0</td>
<td>56.4</td>
<td>52.2</td>
<td>50.4</td>
<td>54.6</td>
<td>55.0</td>
</tr>
<tr>
<td>M4</td>
<td>57.8</td>
<td>62.4</td>
<td>61.8</td>
<td>60.6</td>
<td>54.0</td>
<td>56.6</td>
<td>63.6</td>
<td><strong>70.8</strong></td>
</tr>
<tr>
<td>M5</td>
<td>55.0</td>
<td>67.6</td>
<td>68.6</td>
<td>66.6</td>
<td>59.6</td>
<td>63.2</td>
<td>64.2</td>
<td><strong>73.34</strong></td>
</tr>
<tr>
<td>M6</td>
<td>52.2</td>
<td>59.0</td>
<td>62.6</td>
<td>61.0</td>
<td>54.2</td>
<td>57.6</td>
<td>62.8</td>
<td><strong>65.2</strong></td>
</tr>
<tr>
<td>M7</td>
<td>44.8</td>
<td>56.4</td>
<td>59.6</td>
<td>60.2</td>
<td>56.8</td>
<td>58.4</td>
<td><strong>63.4</strong></td>
<td><strong>64.2</strong></td>
</tr>
<tr>
<td>M8</td>
<td>64.8</td>
<td>80.8</td>
<td>81.4</td>
<td>66.6</td>
<td>64.0</td>
<td>65.2</td>
<td>82.2</td>
<td>81.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td># Target Mixtures</td>
<td>500</td>
<td>More challenging mixture model</td>
</tr>
<tr>
<td># Clutter Mixtures</td>
<td>500</td>
<td>LR suffers from noise; SC helps</td>
</tr>
<tr>
<td>% Training</td>
<td>50</td>
<td>TDDL relatively strong performer</td>
</tr>
<tr>
<td># Ingredients</td>
<td>5</td>
<td>kNN3 pretty good, especially when given enough data</td>
</tr>
<tr>
<td>Min. % Target</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Max. % Target</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Noise Variance</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>TDDL Iterations</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

(* := TD-200)
Software

Platform Load Sharing Facility (LSF) scheduler on 20 compute nodes (Intel Xeon X5650, 12 threads)

Software includes scripts for various tasks (kfold CV, train/test)

$ lsload

<table>
<thead>
<tr>
<th>HOST_NAME</th>
<th>status</th>
<th>r15s</th>
<th>r1m</th>
<th>r15m</th>
<th>ut</th>
<th>pg</th>
<th>ls</th>
<th>it</th>
<th>tmp</th>
<th>swp</th>
<th>mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn17</td>
<td>ok</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0%</td>
<td>0.0</td>
<td>0</td>
<td>40576</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>maul</td>
<td>ok</td>
<td>0.0</td>
<td>0.2</td>
<td>0.1</td>
<td>0%</td>
<td>0.0</td>
<td>7</td>
<td>15</td>
<td>19G</td>
<td>26G</td>
<td>20G</td>
</tr>
<tr>
<td>cn00</td>
<td>ok</td>
<td>12.0</td>
<td>12.2</td>
<td>11.8</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>10528</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn08</td>
<td>ok</td>
<td>12.0</td>
<td>12.5</td>
<td>12.0</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn12</td>
<td>ok</td>
<td>12.0</td>
<td>12.2</td>
<td>11.7</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn07</td>
<td>ok</td>
<td>12.0</td>
<td>12.3</td>
<td>11.8</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn19</td>
<td>ok</td>
<td>12.0</td>
<td>12.2</td>
<td>11.7</td>
<td>98%</td>
<td>0.0</td>
<td>0</td>
<td>9040</td>
<td>8832M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn15</td>
<td>ok</td>
<td>12.0</td>
<td>12.0</td>
<td>11.8</td>
<td>98%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn13</td>
<td>ok</td>
<td>12.0</td>
<td>12.4</td>
<td>11.7</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>7264</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn04</td>
<td>ok</td>
<td>12.0</td>
<td>11.6</td>
<td>11.6</td>
<td>98%</td>
<td>0.0</td>
<td>0</td>
<td>9016</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn02</td>
<td>ok</td>
<td>12.0</td>
<td>12.3</td>
<td>11.9</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>47712</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn03</td>
<td>ok</td>
<td>12.1</td>
<td>12.4</td>
<td>12.1</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>29312</td>
<td>8832M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn05</td>
<td>ok</td>
<td>12.1</td>
<td>12.4</td>
<td>11.7</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>7504</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn09</td>
<td>ok</td>
<td>12.2</td>
<td>12.1</td>
<td>11.8</td>
<td>98%</td>
<td>0.0</td>
<td>0</td>
<td>20240</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn11</td>
<td>ok</td>
<td>12.2</td>
<td>12.1</td>
<td>11.9</td>
<td>98%</td>
<td>0.0</td>
<td>0</td>
<td>9032</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn14</td>
<td>ok</td>
<td>12.2</td>
<td>11.9</td>
<td>11.6</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn16</td>
<td>ok</td>
<td>12.3</td>
<td>11.6</td>
<td>11.7</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn01</td>
<td>ok</td>
<td>12.3</td>
<td>11.9</td>
<td>11.8</td>
<td>98%</td>
<td>0.0</td>
<td>1</td>
<td>71</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn10</td>
<td>ok</td>
<td>12.3</td>
<td>12.1</td>
<td>11.8</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>30672</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn18</td>
<td>ok</td>
<td>12.3</td>
<td>12.8</td>
<td>11.9</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>78</td>
<td>8832M</td>
<td>2000M</td>
<td>22G</td>
</tr>
<tr>
<td>cn06</td>
<td>ok</td>
<td>12.6</td>
<td>11.5</td>
<td>11.5</td>
<td>99%</td>
<td>0.0</td>
<td>0</td>
<td>3e+05</td>
<td>8824M</td>
<td>2000M</td>
<td>22G</td>
</tr>
</tbody>
</table>
Deliverables

Software/Data Sets

- Solvers (LARS, F-S, TDDDL): ~2000 lines of Matlab
  - Diabetes data set downloaded from LARS author’s website; removed header (provided)
  - Test matrices constructed on-the-fly by unit tests (provided)
  - Limited doxygen documentation (requires doxygen and “Using Doxygen with Matlab” from Matlab Central to regenerate)
- Analysis experiments: ~1500 lines of Matlab, ~140 lines bash
  - URLs to HSI data sets provided in references
- USGS Viewer: ~500 lines of Matlab


Final report and software tarball to be delivered by May 11
**Solvers**

Here is a list of all files with brief descriptions:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>FeatureSign/fs_11.m</code></td>
<td>Implements the sparse coding algorithm 1 from [1]</td>
</tr>
<tr>
<td><code>FeatureSign/Unitests/compare_to_lars.m</code></td>
<td>Compares LARS and FeatureSign solutions for consistency</td>
</tr>
<tr>
<td><code>FeatureSign/Unitests/fs1_test_correlated.m</code></td>
<td>Make sure the algorithm runs with correlated atoms</td>
</tr>
<tr>
<td><code>LARS/lars.m</code></td>
<td>An implementation of the Least Angle Regression algorithm [1]</td>
</tr>
<tr>
<td><code>LARS/Unitests/fs1_math_fonts.m</code></td>
<td>Improves fonts of the current plot</td>
</tr>
<tr>
<td><code>LARS/Unitests/plot_k_vs_cost.m</code></td>
<td>Plots objective function cost vs iteration</td>
</tr>
<tr>
<td><code>LARS/Unitests/pplot_vs_value.m</code></td>
<td>Plots LARS results for diabetes data set</td>
</tr>
<tr>
<td><code>LARS/Unitests/test_correlated.m</code></td>
<td>Make sure the algorithm runs with correlated atoms</td>
</tr>
<tr>
<td><code>LARS/Unitests/test_diabetes.m</code></td>
<td>Runs LARS on the diabetes data set provided by [1]</td>
</tr>
<tr>
<td><code>LARS/Unitests/test_orthogonal.m</code></td>
<td>Test the LARS algorithm using an orthogonal dictionary</td>
</tr>
<tr>
<td><code>LARS/Unitests/test_problemA.m</code></td>
<td>Runs a specific case that was problematic in the past</td>
</tr>
<tr>
<td><code>LARS/Unitests/test_problemB.m</code></td>
<td>Runs a specific case that was problematic in the past</td>
</tr>
<tr>
<td><code>TDDL/tddl.m</code></td>
<td>Task driven dictionary learning solver</td>
</tr>
<tr>
<td><code>TDDL/tddl_calc_gradients.m</code></td>
<td>Gradient calculations for the TDDL algorithm</td>
</tr>
<tr>
<td><code>TDDL/tddl_mpi.m</code></td>
<td>Task driven dictionary learning, MPI version (deprecated)</td>
</tr>
<tr>
<td><code>TDDL/Unsupervised_dm.m</code></td>
<td>Dictionary learning w/o a task</td>
</tr>
<tr>
<td><code>TDDL/Unitests/test_grad_w.m</code></td>
<td>Some experiments with TDDL gradients of w (deprecated)</td>
</tr>
</tbody>
</table>
Summary

Project Goals Met

- Implemented algorithms from three papers
  - (LARS, Feature-Sign, TDDL)
- Validated using data sets with existing/known results
  - (diabetes, orthogonal designs, USPS)
- Conducted new experiments with hyperspectral data sets
  - (Urban, USGS)

Thanks!!

- Dr.’s Levy, Balan, Ide, Wang, Banerjee for guidance and help throughout the course!
- AMSC663/4 for great questions and enduring four presentations on this topic 😊


Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Datasets for ”the elements of statistical learning”, 2009.  

