Online Multi-Task Learning based on K-SVD

Paul Ruvolo, Eric Eaton

Bryn Mawr College, Computer Science Department

June 13, 2014
Presented by Zhengming Xing
Outline

• Introduction
• Multi-Task learning using K-SVD
• Lifelong learning
• Experiment Results
Introduction

Objective and novelty of the paper
Learning and sharing knowledge between multiple tasks to improve the performance.

- K-SVD based algorithm to reduce computation.
- Online Multi-Task learning

Review of K-SVD
Given the input signal \( \{x_n\}_{n=1}^{N} \in \mathbb{R}^d \), obtain an overcomplete dictionary and sparse representation of the signal.

\[
\text{argmin}_{L} \sum_{i=1}^{n} \min_{s^{(i)}} \{\|Ls^{(i)} - x_i\|_2^2 + \mu \|s^{(i)}\|_0\} \tag{1}
\]
Solving equation 1 includes two optimization steps. Let us define $\mathbf{S} = [\mathbf{s}^{(1)}...\mathbf{s}^{(n)}]$ and $\mathbf{X} = [\mathbf{x}^{(1)}...\mathbf{x}^{(n)}]$

- **Sparse coding stage**
  Given a fixed value of $\mathbf{L}$, solve the following problem
  $$\mathbf{s}^{(i)} \leftarrow \text{argmin}_\mathbf{s}\{\|\mathbf{Ls} - \mathbf{x}_i\|_2^2 + \mu\|\mathbf{s}\|_0\}$$

- **Codebook update stage.**
  $$\|\mathbf{X} - \mathbf{LS}\|_2^2 = \|\mathbf{X} - \sum_{k \neq j} \mathbf{l}_k \mathbf{s}_k^T - \mathbf{l}_j \mathbf{s}_j^T\|_2^2 = \|\mathbf{E}_j - \mathbf{l}_j \mathbf{s}_j^T\|_2^2$$
  Using SVD to solve the above problem
  $$(\mathbf{U}, \Sigma, \mathbf{V}) = \text{SVD}(\mathbf{E}_j^A)$$
  $$\mathbf{l}_j \leftarrow \mathbf{u}_1, \mathbf{s}^{(A)}_j \leftarrow \sigma_{1,1} \mathbf{v}_1$$

where $\mathbf{E}_j = \mathbf{X} - \sum_{k \neq j} \mathbf{l}_k \mathbf{s}_k^T$, $\mathcal{A}$ indicates the columns corresponds to nonzero $s_{ji}$. 
Multi-task learning using K-SVD

Let \( \{ Z^{(1)}, Z^{(2)}, \ldots, Z^{(T)} \} \) denotes a set of supervised learning task.

- Each task \( Z^{(t)} = (\hat{f}(t), X^{(t)}, y^{(t)}) \) is defined by a mapping \( \hat{f}(t) \) from instance space \( X^{(t)} \) to a set of label \( y^{(t)} \).

- Each task model \( f^{(t)}(x) = f(x, \theta^{(t)}) \) is specified by a task-specific parameter vector \( \theta^{(t)} \in \mathbb{R}^d \).

- Each task parameter vector \( \theta^{(t)} = Ls^{(t)}, L \in \mathbb{R}^{d \times k} \) is shared for all task and \( s^{(t)} \) is sparse.

The object function is as following

\[
e_T(L) = \sum_{t=1}^{T} \min_{s^{(t)}} \left\{ \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}(f(x_i^{(t)}; Ls^{(t)}), y_i^{(t)}) + \lambda \|Ls^{(t)}\|_2^2 + \mu \|s^{(t)}\|_0 \right\}
\]

where \( \mathcal{L} \) is some loss function for fitting the task model.
Second order Taylor expansion of the objective function around $\theta^{(t)}$. where $\theta^{(t)}$ is an optimal predictor for task $t$ only.

$$g_T(L) = \sum_{t=1}^{T} \min_{s(t)} \left\{ \frac{1}{n_t} ||\theta^{(t)} - Ls^{(t)}||_D^2 + \mu ||s^{(t)}||_0 \right\}$$

$||v||_A = v^TAv$ and

$$D^{(t)} = \lambda I + \nabla_{\theta, \theta}^2 \frac{1}{2n_t} \sum_{i=1}^{n_t} \mathcal{L}(f(x_i^{(t)}; \theta), y_i^{(t)}) \mid_{\theta = \theta^{(t)}}$$

$$\theta^{(t)} = \arg\min_{\theta} \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}(f(x_i^{(t)}, \theta), y_i^{(t)}) + \lambda ||\theta||_2^2$$
Multi-task learning using K-SVD

- Sparse coding step
  \[ s^{(i)} \leftarrow \arg\min_s \{ \| Ls - x_i \|_2^2 + \mu \| s \|_0 \} \]

- Codebook update step
  Replace the original SVD with generalized SVD
  \[ (U, \Sigma, V) = \text{GSVD}(E^{A_j}, M, W) \]
  where \( M = \frac{1}{|A|} \sum_{t \in A} D^{(t)} \), \( W \) is diagonal with
  \[ w_{t,t} = \frac{1^{T} D^{(A_t)} 1}{\sum_{t' \in A} 1^{T} D^{(A_{t'})} 1}. \]
  In this way, the update values of \( l_j \) and \( s_j^A \) are obtained via minimizing
  \[ \sum_{t1=1}^{|A|} \sum_{t2=1}^{|A|} w_{t1,t2}(e_j^{A_{t1}} - l_j s_j^{A_{t1}})^T M (e_j^{A_{t2}} - l_j s_j^{A_{t2}}) \]
  Generalized SVD, the matrix \( A \) decompose to
  \[ A = U \Sigma V^T \] with: \( U^T M U = I \) and \( V^T W V = I \)
Algorithm

Algorithm 1 MTL-SVD

Input training data \((X^{(1)}, y^{(1)}), \ldots, (X^{(T)}, y^{(T)})\); dictionary size \(k\)

for \(i = 1 \) to \( T \) do
  \((\theta^{(i)}, D^{(i)}) \leftarrow \text{singleTaskLeaner}(X^{(i)}, y^{(i)})\)
end for

while convergence criteria not meet do
  for \( t = 1 \) to \( T \) do
    update \( s^{(t)} \)
  end for
  for \( j = 1 \) to \( k \) do
    update \( l_j \) and \( s_j^A \)
  end for
end while
Lifelong learning problem

1.) Tasks are received sequentially
2.) Knowledge is transferred from previously learned tasks
3.) New knowledge is stored for future use
4.) Existing knowledge is refined
Lifelong learning problem

Adaptions to allow efficient lifelong learning

• update the value of $s^{(t)}$ only based on the training data for task $t$.

• update subset of columns of $L$ corresponding to non-zero entries of $s^{(t)}$.

• the above two steps only performed once per patch.
Experiment Results

Synthetic Regression Tasks: 100 tasks with $d = 13$ and $n_t = 100$. Land Mine Detection: 14820 data instances divided into 29 different geographical regions.
Facial Expression Recognition: 3 facial action units. 21 tasks and each with 450 – 999 images.

London School Data: Exam score from 15362 students from 139 schools with $d = 27$. 