Online Learning for Multi-Task Feature Selection

Haiqin Yang
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Joint work with
Irwin King and Michael R. Lyu

November 28, 2010
1 Introduction
   • Motivation
   • Problems and Contributions

2 Formulation and Algorithm
   • Problem Setup
   • Algorithm
   • Properties

3 Experiments
   • Experimental Setup
   • School Data
   • Conjoint Analysis

4 Summary
Why Multi-Task Feature Selection?

- Observations: Related tasks contain helpful information; Redundant/irrelevant features exist

  - **Gene selection** from microarray data in related diseases
    - Variables: Gene expression coefficients corresponding to the amount of mRNA in a patient’s sample (e.g., tissue biopsy)
    - Tasks: Distinguish healthy from unhealthy for different diseases
    - Problems: few samples (< 100’s), large variables (>1000’s)

  - **Text categorization** from documents in multiple related categories
    - Features represented by a vector of vocabulary on word frequency counts
    - Vocabulary: > 10000’s words
    - Tasks: 1) Detecting spam-emails from persons with same interests; 2) Automatic classifying related web page categories

- Related work
  - A generalized $L_1$-norm single-task regularization (Argyriou et. al. 2008)
  - Mixed norms of $L_1$, $L_2$, and $L_\infty$ norms (Obozinski et. al. 2009)
  - Nesterov’s method on MTFS (Liu et. al. 2009)
  - $L_{0,0}$-regularization based on MIC (Dhillon et. al. 2009)
Problems and Contributions

- Problems
  - Features among tasks are **redundant** or **irrelevant**
  - Data come in **sequence**
  - Data are **large** in volume

- Our contributions
  - The **first** online learning framework for multi-task feature selection
  - Easy implementation: **three lines of main codes**
  - **Efficiency** in both time complexity and memory cost, $O(d \times Q)$
  - Find important features and important tasks that dominating the features
  - Easily extend to nonlinear models
Multi-Task Feature Selection Models

- **Data:** i.i.d. observations: $\mathcal{D} = \bigcup_{q=1}^{Q} \mathcal{D}_q$
  
  $\mathcal{D}_q = \{z^q_i = (x^q_i, y^q_i)\}_{i=1}^{N_q}$ sampled from $\mathcal{P}_q$, $q = 1, \ldots, Q$
  $x \in \mathbb{R}^d$ – input variable, $y \in \mathbb{R}$ – response

- **Model:** $f_q(x) = w^q \top x$, $q = 1, \ldots, Q$

- **Objective:** $\min_{\mathbf{W}} \sum_{q=1}^{Q} \frac{1}{N_q} \sum_{i=1}^{N_q} \ell^q(\mathbf{W}_{\cdot q}, z^q_i) + \Omega_\lambda(\mathbf{W})$
  
  $\mathbf{W} = (w^1, w^2, \ldots, w^Q) = (\mathbf{W}_{\cdot 1}, \ldots, \mathbf{W}_{\cdot Q}) = (\mathbf{W}_{\cdot 1}^\top, \ldots, \mathbf{W}_{\cdot d}^\top)$

  - **iMTFS:** $\Omega_\lambda(\mathbf{W}) = \lambda \sum_{q=1}^{Q} \| \mathbf{W}_{\cdot q} \|_1 = \lambda \sum_{j=1}^{d} \| \mathbf{W}_{j\cdot} \|_1$

  - **aMTFS:** $\Omega_\lambda(\mathbf{W}) = \lambda \sum_{j=1}^{d} \| \mathbf{W}_{j\cdot}^\top \|_2$

  - **MTFTS:** $\Omega_\lambda,r = \lambda \sum_{j=1}^{d} \left( r_j \| \mathbf{W}_{j\cdot}^\top \|_1 + \| \mathbf{W}_{j\cdot}^\top \|_2 \right)$

\[
\begin{bmatrix}
  x & 0 & 0 & x & x \\
  0 & x & x & x & x \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  x & 0 & x & x & x \\
\end{bmatrix},
\begin{bmatrix}
  x & x & x & x & x \\
  0 & 0 & 0 & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  x & x & x & x & x \\
\end{bmatrix},
\begin{bmatrix}
  x & 0 & x & x & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & x & 0 & x & x \\
\end{bmatrix}
\]
Online Learning Algorithm Framework for MTFS

**Initialization**: $W_1 = W_0$, $\tilde{G}_0 = 0$

**for** $t = 1, 2, 3, \ldots$

1. Compute the subgradient on $W_t$, $G_t \in \partial l_t$
2. Update the average subgradient $\tilde{G}_t$:
   \[
   \tilde{G}_t = \frac{t-1}{t} \tilde{G}_{t-1} + \frac{1}{t} G_t
   \]
3. Calculate the next iteration $W_{t+1}$:
   \[
   W_{t+1} = \arg\min_W \gamma(W) \triangleq \left\{ \tilde{G}_t^T W + \Omega_\lambda(W) + \frac{\gamma}{\sqrt{t}} h(W) \right\}
   \]

**Remarks**

- **$W$**: Matrix
- Original formulation is in linear case; it can be extended to non-linear case easily
- Motivated by the success of dual averaging method (Xiao, 2009; Yang et. al. 2010)
Updating Rules for Online MTFS

Define: \( h(W) = \frac{1}{2} \|W\|_F^2 \)

- **iMTFS**: For \( i = 1, \ldots, d \) and \( q = 1, \ldots, Q \),
  \[
  (W_{i,q})_{t+1} = -\frac{\sqrt{t}}{\gamma} \left[ \|\bar{G}_{i,q}\| - \lambda \right]_+ \cdot \text{sign} \left( (\bar{G}_{i,q})_t \right).
  \]

- **aMTFS**: For \( j = 1, \ldots, d \),
  \[
  (W_{j\cdot})_{t+1} = -\frac{\sqrt{t}}{\gamma} \left[ 1 - \frac{\lambda}{\|\bar{G}_{j\cdot}\|_2} \right]_+ \cdot (\bar{G}_{j\cdot})_t.
  \]

- **MTFTS**: For \( j = 1, \ldots, d \),
  \[
  (W_{j\cdot})_{t+1} = -\frac{\sqrt{t}}{\gamma} \left[ 1 - \frac{\lambda}{\|\bar{U}_{j\cdot}\|_2} \right]_+ \cdot (\bar{U}_{j\cdot})_t,
  \]

where the \( q \)-th element of \( (\bar{U}_{j\cdot})_t \) is calculated by
\[
(\bar{U}_{j,q})_t = \left[ \|\bar{G}_{j,q}\| - \lambda r_j \right]_+ \cdot \text{sign} \left( (\bar{G}_{j,q})_t \right), \quad q = 1, \ldots, Q.
\]

**Efficiency**: \( \mathcal{O}(d \times Q) \) in memory cost and time complexity
Theoretical Results

Average regret for MTFS

\[
\bar{R}_T(w) := \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{T} \sum_{t=1}^{T} (\Omega_{\lambda}(W_t) + l_t(W_t)) - S_T(W)
\]

Theoretical bounds

\[
\bar{R}_T \sim \mathcal{O}(1/\sqrt{T})
\]
Experimental Setup

Data
- School data
- Computer survey data

Comparison algorithms
- iMTFS
- aMTFS
- DA-iMTFS
- DA-aMTFS
- DA-MTFTS

Platform
- PC with 2.13 GHz dual-core CPU
- Batch-mode algorithms: Matlab
- Online-mode algorithms: Matlab
School Data

Description

- **Objective:** Predict exam scores
- **Data:** Exam scores of 15,362 students from 139 secondary schools in London during the years 1985, 1986, and 1987, $Q = 139$
- **Features:** Year of the exam (YR), 4 school-specific and 3 student-specific features, $d = 27$

Setup

- **Evaluation:** Explained variance ($R^2$) $1 - \frac{SS_{err}}{SS_{tol}}$, the larger the better
- **Loss:** Square loss
- **Parameters setting:** Cross validation (hierarchical search and grid search)
**School Data Results**

### Accuracy

- Learning multiple tasks simultaneously can gain over 50% improvement than learning the task individually.
- Online learning algorithms attain (nearly) the same accuracies as batch-trained algorithms.
- **DA-MTFTS** attains the same accuracy as **DA-aMTFS** with fewer NNZs.

<table>
<thead>
<tr>
<th>Method</th>
<th>Explained variance</th>
<th>NNZs</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>aMTFS</td>
<td>21.0±1.7</td>
<td>815.5±100.6</td>
<td>$\lambda = 300$</td>
</tr>
<tr>
<td>iMTFS</td>
<td>13.5±1.8</td>
<td>583.0±16.6</td>
<td>$\lambda = 40$</td>
</tr>
<tr>
<td>DA-aMTFS</td>
<td>20.8±1.8</td>
<td>605.8±180.3</td>
<td>$\lambda = 20, \gamma = 1, \text{ep}=120$</td>
</tr>
<tr>
<td>DA-MTFTS</td>
<td>20.8±1.9</td>
<td>483.7±130.7</td>
<td>$\lambda = 20, \gamma = 1, \text{ep}=120$</td>
</tr>
<tr>
<td>DA-iMTFS</td>
<td>13.5±1.8</td>
<td>1037.1±21.4</td>
<td>$\lambda = 1, \gamma = 50, \text{ep}=120$</td>
</tr>
</tbody>
</table>
Effect of $\lambda$ and $\gamma$

Results

✓ NNZs decreases as $\lambda$ increases
✓ NNZs increases as $\gamma$ increases
✓ Fewer NNZs in DA-MTFTS than DA-aMTFS
Conjoint Analysis

Description

- **Objective**: Predict rating by estimating respondents’ partworths vectors
- **Data**: Ratings on personal computers of 180 students for 20 different PC, \( Q = 180 \)
- **Features**: Telephone hot line (TE), amount of memory (RAM), screen size (SC), CPU speed (CPU), hard disk (HD), CDROM/multimedia (CD), cache (CA), color (CO), availability (AV), warranty (WA), software (SW), guarantee (GU) and price (PR); \( d = 14 \)

Setup

- **Evaluation**: Root mean square errors (RMSEs)
- **Loss**: Square loss
- **Parameters setting**: Cross validation (hierarchical and grid search)
Conjoint Analysis Results

Accuracy

- Learning partworths vectors across respondents can help to improve the performance
- Online learning algorithms attain nearly the same accuracies as batch-trained algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSEs</th>
<th>NNZs</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>aMTFS</td>
<td>1.82</td>
<td>2148</td>
<td>$\lambda = 44.5$</td>
</tr>
<tr>
<td>iMTFS</td>
<td>1.91</td>
<td>789</td>
<td>$\lambda = 3$</td>
</tr>
<tr>
<td>DA-aMTFS</td>
<td>2.04</td>
<td>540</td>
<td>$\lambda = 20.0, \gamma = 0.9, \text{ep}=1$</td>
</tr>
<tr>
<td>DA-aMTFS</td>
<td>1.83</td>
<td>1800</td>
<td>$\lambda = 5, \gamma = 0.9, \text{ep}=20$</td>
</tr>
<tr>
<td>DA-iMTFS</td>
<td>2.43</td>
<td>199</td>
<td>$\lambda = 2.0, \gamma = 2.0, \text{ep}=1$</td>
</tr>
<tr>
<td>DA-iMTFS</td>
<td>1.92</td>
<td>662</td>
<td>$\lambda = 0.5, \gamma = 1.0, \text{ep}=20$</td>
</tr>
</tbody>
</table>
Effect of $\lambda$ and $\gamma$

Results

✓ NNZs decreases as $\lambda$ increases
✓ NNZs increases as $\gamma$ increases
Efficiency

Time cost

- **School Data**
  - aMTFTS: 1.30s
  - DA-MTFTS: 0.99s

- **Conjoint Analysis**
  - iMTFTS: 0.326s
  - aMTFTS: 0.162s
  - DA-iMTFS: 0.08s
  - DA-aMTFS: 0.07s
A novel online learning algorithm framework for multi-task feature selection

Apply this framework for several multi-task feature selection models

Provide closed-form solutions to update the models

Provide the convergence rate of the average regret

Experimental results demonstrate the proposed algorithms in both efficiency and effectiveness
Questions ?

Haiqin Yang
www.cse.cuhk.edu.hk/~hqyang
hqyang@cse.cuhk.edu.hk