A Randomized Approach for Crowdsourcing in the Presence of Multiple Views

Presenter: Yao Zhou  joint work with: Jingrui He

Arizona State University  Arizona State University
Roadmap

- Motivation
- Proposed framework: M2VW
- Experimental results
- Conclusion
Feature Heterogeneity (Multi-view)

- Example: Document classification

Documents → First Dictionary → First View
Documents → Second Dictionary → Second View

Different dictionaries
Bag of words under different views
Feature Heterogeneity (Multi-view)

- Example: Image classification

- Image set
- Different feature extractions

View 1: SIFT
View 2: HOG
View 3: Deep Features
View 4: Contour Features
View Consistency

Initial Data $D_L$

View 1
Classifier 1 $F^1(\bullet)$

View 2
Classifier 2 $F^2(\bullet)$

New Data $x'$

Make Predictions on

$F^1(x') = F^2(x')$
Crowdsourcing

What is crowdsourcing?
- Crowdfunding (Kickstarter)
- Collective Knowledge (Data labeling, Foreign language translation)
- Collective Creativity (Analogy mining)
- Implicit Crowdsourcing (CAPTCHA)

Crowdsourcing in Machine Learning
- In ML, training a (semi)supervised model needs training labels. Many crowdsourcing platforms provide services to collect labels information.
Worker Consensus

<table>
<thead>
<tr>
<th>Items</th>
<th>Workers</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="fox" alt="" /></td>
<td>W</td>
</tr>
<tr>
<td><img src="cat" alt="" /></td>
<td>D</td>
</tr>
<tr>
<td><img src="dog" alt="" /></td>
<td>D</td>
</tr>
<tr>
<td><img src="wolf" alt="" /></td>
<td>W</td>
</tr>
<tr>
<td>?</td>
<td>W</td>
</tr>
<tr>
<td>?</td>
<td>D</td>
</tr>
<tr>
<td>?</td>
<td>D</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>W</td>
<td>D</td>
</tr>
<tr>
<td>?</td>
<td>W</td>
</tr>
</tbody>
</table>

Example of crowdsourcing labels: Wild animals (denoted by "W"), domestic animals (denoted by "D"), and missing labels (denoted by "?").

Predictions of the workers regarding the same item should be similar.

Low-cost and efficient:
Collecting a large number of labels in a short period of time.
Research Questions

- Q1: How to model the multi-view learning problem using crowdsourcing labels?

- Q2: What is the appropriate tool to solve it?

- Q3: How to speed up?
Roadmap

- Motivation
- Proposed framework: M2VW
- Experimental results
- Conclusion
M2VW: Formulation

- Weight matrix of the workers: $W \in \mathbb{R}^{P \times N_w}$

$$W = \begin{bmatrix} w^1_1 & \ldots & w^1_{N_w} \\ \vdots & \ddots & \vdots \\ w^m_1 & \ldots & w^m_{N_w} \\ \vdots & \ddots & \vdots \\ w^V_1 & \ldots & w^V_{N_w} \end{bmatrix}$$

where, $w^m_k \in \mathbb{R}^{d_m}$ indicates the weights learned for the $k^{th}$ worker in the $m^{th}$ view.
M2VW: Formulation

- **Prediction tensor:** \( \mathcal{A} \in \mathbb{R}^{N \times N_w \times V} \)

The \( i \)th slice is defined as:

\[
\mathcal{A}_{i::} = W^T (x_i e_v^T) \odot B \in \mathbb{R}^{N_w \times V}
\]

- \( N \): # of examples
- \( N_w \): # of workers
- \( V \): # of views

Vector of all 1’s (of size \( V \))

Feature vector of \( x_i \)

Block diagonal matrix

\( B = \text{diag}(e_{d_1}, \ldots, e_{d_V}) \)

Prediction of the \( i \)th example made by the \( k \)th worker under the \( m \)th view
M2VW: Formulation

- Optimization formulation:

\[
\min_W \sum_{k=1}^{N_w} \sum_{i=1}^{N} \mathcal{L}(Y_{ik}, w_k^T x_i) + \text{Rank}(\mathcal{A}) + \mathcal{R}(W)
\]

- Remarks:
  - Problem of rank minimization is NP-hard\(^1\).
  - Rank of a tensor is not uniquely defined\(^2,3\).

\[
\text{Rank}(\mathcal{A}) \approx \|\mathcal{A}\|_* := \sum_{l=1}^{3} \alpha_l \|\mathcal{A}(l)\|_*
\]

\[
s.t. \sum_{l=1}^{3} \alpha_l = 1, \alpha_l \geq 0, l = 1, 2, 3
\]

M2VW: Formulation

- Interpretations of the terms:
  - Loss function:
    \[ \mathcal{L}(Y_{ik}, w_k^T x_i) = \log(1 + \exp(-Y_{ik} w_k^T x_i)) \]
  - First matricization: \( \mathcal{A}_{(1)} \in \mathbb{R}^{N_w \times NV} \)

Minimizing \( \|\mathcal{A}_{(1)}\|_* \) requires the predictions of different workers over the same item on the same view to be correlated.
M2VW: Formulation

- Interprets the terms:
  - Second matricization: $A_{(2)} \in \mathbb{R}^{V \times N \times N}$

- Remarks:
  - Third matricization requires the predictions of different worker-view combinations on the same item should also be correlated, which is the repetition of *Worker Consensus* and *View Consistency*.

Minimizing $\|A_{(2)}\|_*$ requires the predictions of different views over the same worker on the same item to be consistent.
M2VW: Formulation

- Interpretations of the terms:
  - Regularization term: \( \mathcal{R}(W) = \|W\|_G + \|W\|_{2,1} \)

\[
\|W\|_G = \sum_{k=1}^{N_W} \sum_{m=1}^{V} \|w^m_k\|_2
\]

**Group sparsity**: group-wise weights for the features that corresponding to a specific worker under a specific view.

\[
\|W\|_{2,1} = \sum_{p=1}^{P} \|W_p:\|_2
\]

**Feature sparsity**: general sparsity weights cross multiple workers.
M2VW: Formulation

- Optimization formulation (relaxed)

\[
\min_{\{W,M_1,M_2\}} \sum_{k=1}^{N_w} \sum_{i=1}^{N} \log \left(1 + \exp(-Y_{ik} w_k^T x_i)\right) + \lambda \left(\|W\|_G + \|W\|_{2,1}\right) \\
+ \sum_{l=1}^{2} \alpha_l \|M_l\|_* + \frac{\beta_l}{2} \|A(l) - M_l\|^2_F
\]

- Key advantages of this relaxation:
  - The interdependent trace norm terms are split, so they can be solved independently.
  - Relaxation penalty term $\|A(l) - M_l\|_F^2$ can be transformed into a smooth differentiable function.
  - The (transformed) terms in objective function is also separable and parallelizable with respect to the workers.
M2VW: Algorithm

Solution: Gradient based method (BCD)

Subproblem of updating $W$: 

$$
\min_W \sum_{k=1}^{N_w} \sum_{i=1}^{N} \log \left( 1 + \exp \left(-Y_{ik} w_k^T x_i \right) \right) + \sum_{l=1}^{2} \frac{\beta_l}{2} \|A(l) - M_l\|_F^2 + \lambda \left( \|W\|_G + \|W\|_{2,1} \right)
$$

- Logistic loss $\mathcal{L}(W)$
- Relaxation penalty $\mathcal{R}(W)$
- Feature sparsity $\mathcal{F}(W)$

Gradient of the loss function $\mathcal{L}(W)$

$$
\frac{\partial \mathcal{L}(W)}{\partial W} = -X \cdot \left[ \frac{e^{-Y_o(X^T W)}}{1 + e^{-Y_o(X^T W)}} \circ Y \right]
$$
M2VW: Algorithm

- Solution: Gradient based method (BCD)
  - Subproblem of updating $W$:

\[
\min_{W} \sum_{k=1}^{N_w} \sum_{i=1}^{N} \log \left( 1 + \exp \left( -Y_{ik}w_{k}^{T}X_{i} \right) \right) + \sum_{l=1}^{2} \frac{\beta_{l}}{2} \left\| A_{l} - M_{l} \right\|_{F}^{2} + \lambda \left( \left\| W \right\|_{G} + \left\| W \right\|_{2,1} \right)
\]

- Gradient of the relaxation penalty $\mathcal{RP}(W)$

\[
\frac{\partial \mathcal{RP}(W)}{\partial \text{vec}(W^{T})} = \text{vec} \left( (\beta_{1} + \beta_{2})W^{T}Q^{T}Q^{T} - \beta_{1}M_{1}Q^{T} - \beta_{2}T(M_{2})Q^{T} \right)
\]

Conclusion from Lemma 1 & Lemma 2
M2VW: Algorithm

- Solution: Gradient based method (BCD)
  - Subproblem of updating $W$:

  $$
  \min_W \sum_{k=1}^{N_w} \sum_{i=1}^{N} \log \left( 1 + \exp(-Y_{ik} w_k^T x_i) \right) + \sum_{l=1}^{2} \frac{\beta_l}{2} \| A_{(l)} - M_l \|_F^2 + \lambda \left( \| W \|_G + \| W \|_{2,1} \right)
  $$

  - Logistic loss $\mathcal{L}(W)$
  - Relaxation penalty $\mathcal{R}(W)$
  - Feature sparsity $\mathcal{F}(W)$

  - Gradient of the feature sparsity $\mathcal{F}(W)$

  $$
  \frac{\partial \| W \|_G}{\partial W_k} = D^k_{g} W_{k:}, \quad k = 1, \ldots, N_w
  $$

  update columns in worker-wise

  $$
  \frac{\partial \| W \|_{2,1}}{\partial W} = D_s W
  $$

  update all entries together
M2VW: Algorithm

- Solution: Gradient based method (BCD)
  - Subproblem of updating $M_l$:
    
    $$
    \min_{M_l} \frac{\alpha_l}{\beta_l} \|M_l\|_* + \frac{1}{2} \|A(l) - M_l\|_F^2
    $$
  
  - Closed form solution\textsuperscript{[4]}:
    
    $$
    D_\tau(A(l)) = U\Sigma_\tau V^T
    $$

    where, $\Sigma_\tau = \text{diag}(\{\sigma_i - \tau\}_+)$, $\tau = \frac{\alpha_l}{\beta_l}$ and $\sigma_i$ is the $i^{th}$ singular value of $A(l)$

M2VW: Algorithm

- Solution: Gradient based method (BCD)
  - **Computational complexity:**

\[ \mathcal{O}\left( N_w N V^2 + n' N_w P (P + N V) \right) \]

- **Observation:**
The complexity is linear w.r.t. the number of workers \( N_w \).

- **Question:**
  **How to speed up?**
M2VVW: Randomized Algorithm

**Theorem 5.1.** [Separability] The gradient of the problem objective is block separable with respective to each worker.

**Remarks:**

\[
\frac{\partial \mathcal{R}(W)}{\partial w_k} = (\beta_1 + \beta_2)QQ^T w_k - Q\left(\beta_1 \text{vec}(M_1^{(k)}) + \beta_2 \text{vec}(\mathbb{T}(M_2)^{(k)})\right)
\]

\[
\frac{\partial \mathcal{L}(W)}{\partial w_k} = -X \cdot \left[ \frac{e^{-Y_{:k} \circ (X^T \cdot w_k)}}{1 + e^{-Y_{:k} \circ (X^T \cdot w_k)}} \circ Y_{:k} \right]
\]

\[
\frac{\partial \mathcal{FS}(W)}{\partial w_k} = D_g^k w_k + D_s w_k
\]
M2VW: Randomized Algorithm

- Decomposition of $W$ into $N_w$ blocks:
  - Any $W$ can be written uniquely as:

$$W = \sum_{k=1}^{N_w} w_k u_k^T$$

- Block (worker) update:
  - For the gradient of the $k^{th}$ worker:

$$W \leftarrow W - \frac{\partial f(W)}{\partial w_k} u_k^T$$

Worker blocks of $W$
M2VW: Randomized Algorithm

- **Batch workers update:**
  - For the round of the BCD iterations, and assume that we select a subset of the block coordinate directions: \( N_b = \{ u_k | k \in [N_w] \} \)

\[
W \leftarrow W - \sum_{k}^{N_b} \left( \frac{\partial \mathcal{L}(W)}{\partial w_k} + \frac{\partial \mathcal{R}(W)}{\partial w_k} + \lambda \frac{\partial \mathcal{F}(W)}{\partial w_k} \right) u_k^T
\]

- **Remarks:**
  - Optimization objective \( f(W) \) is smooth and block separable.
  - The subset of the block coordinate directions are uniformly selected with probability of \( \frac{1}{N_w} \).
Roadmap

- Motivation
- Proposed framework: M2VW
- Experimental results
- Conclusion
Experiment

 Dataset:

- **20 Newsgroups**\(^{[5]}\): two of its largest subsets, 50 synthetic workers.
- **Animal Breed**\(^{[6]}\): subset of ImageNet, 31 real crowdsourcing workers.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Positive Class</th>
<th>Negative Class</th>
<th># Examples (+/-)</th>
<th># Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp. vs. Sci.</td>
<td>comp.os.ms-windows.misc</td>
<td>sci.crypt</td>
<td>1875 (967/908)</td>
<td>150 (80/70)</td>
</tr>
<tr>
<td></td>
<td>comp.sys.mac.hardware</td>
<td>sci.space</td>
<td>1827 (871/956)</td>
<td>150 (80/70)</td>
</tr>
<tr>
<td>Rec. vs. Talk</td>
<td>rec.autos</td>
<td>talk.politics.guns</td>
<td>1844 (975/869)</td>
<td>150 (80/70)</td>
</tr>
<tr>
<td></td>
<td>rec.sport.baseball</td>
<td>talk.politics.mideast</td>
<td>1545 (860/685)</td>
<td>150 (80/70)</td>
</tr>
<tr>
<td>Animal Breed</td>
<td>domestic cat</td>
<td>wild cat</td>
<td>439 (245/194)</td>
<td>120 (110/10)</td>
</tr>
<tr>
<td></td>
<td>domestic canidae</td>
<td>wild canidae</td>
<td>514 (235/279)</td>
<td>120 (110/10)</td>
</tr>
<tr>
<td></td>
<td>domestic horse</td>
<td>wild horse</td>
<td>485 (266/219)</td>
<td>120 (110/10)</td>
</tr>
</tbody>
</table>

---


Experiment

Effectiveness results:

- Evaluation metric: Average F1-score.
- Comparison methods:
  - ConLR: Logistic Regression using concatenated features.
  - PMC\textsuperscript{[7]}: Pseudo Multi-view Co-training.
  - VRKHS\textsuperscript{[8]}: Vector-valued RKHS multi-view learning.
  - MultiC\textsuperscript{2}[6]: Heterogeneous classification using crowdsourcing labels.

![Graphs showing F1-score comparison between different methods](image)

Figure 1: Left: 20 Newsgroups data set (10 random runs of train/test splits). Right: Animal data set.

Experiment

- Terms necessities and parameter sensitivity:

- Baseline I: Loss term only
- Baseline II: Loss term + sparsity term
- Baseline III: Loss term + low-rank term

Figure 2: Left: necessity of feature sparsity term. Right: necessity of low-rank prediction tensor term.
Experiment

Efficiency:

(a) Performance (F1-score) on Comp. vs. Sci.

(b) Performance (F1-score) on Rec. vs. Talk

(c) Running time (in seconds) of Comp. vs. Sci.

(d) Running time (in seconds) of Rec. vs. Talk

Figure 4: Efficiency of Batch-RBCD

Improved performance with proper worker batch size

Run time scales linearly w.r.t. the number of workers
Conclusion

- **Dual heterogeneity learning framework:**
  - Feature heterogeneity (multi-view learning)
  - Worker heterogeneity (crowdsourcing)

- **Algorithms:**
  - Relaxation leads to independent and differentiable objective.
  - Separability of the objective leads to RBCD solution.

- **Experiment results:**
  - Consistently better results on synthetic and real dataset.
  - Linear scalability w.r.t. the number of workers.
Thank you!
&
Questions?