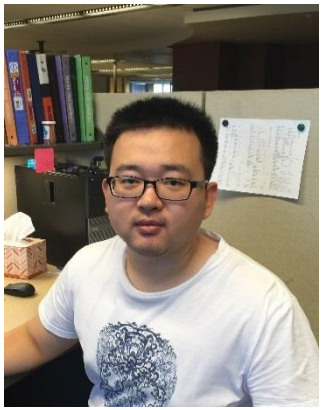
A Randomized Approach for Crowdsourcing in the Presence of Multiple Views

Presenter: Yao Zhou joint work with: Jingrui He



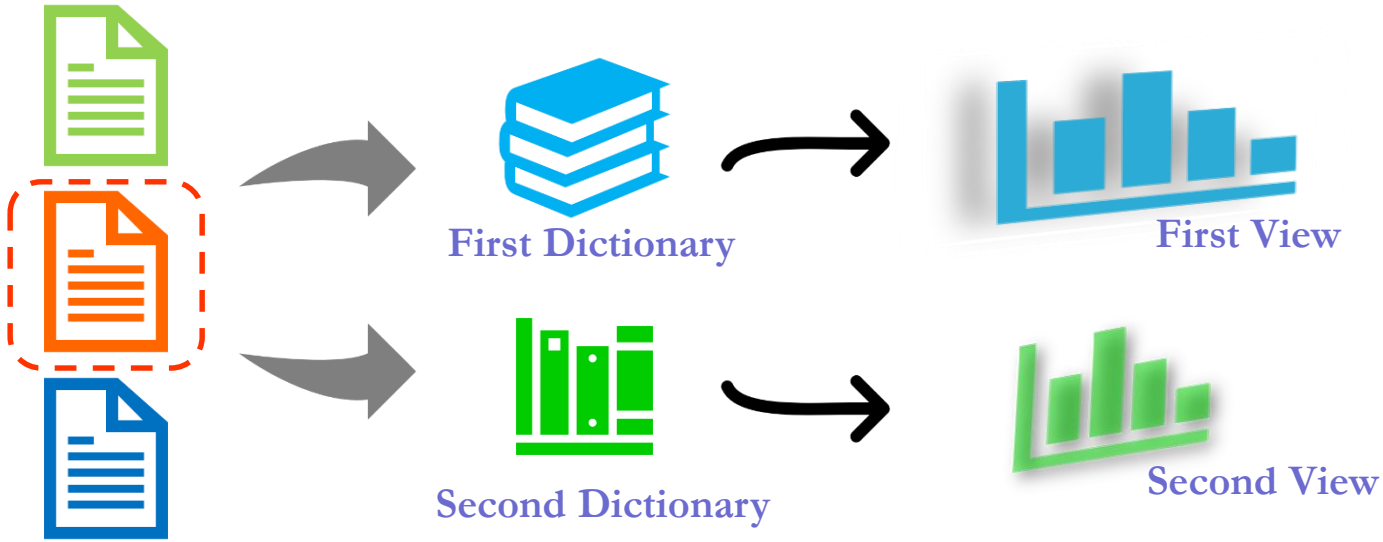
Arizona State University Arizona State University

Roadmap

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

Feature Heterogeneity (Multi-view)

- Example: Document classification

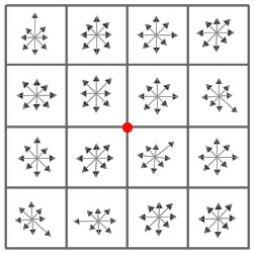


Feature Heterogeneity (Multi-view)

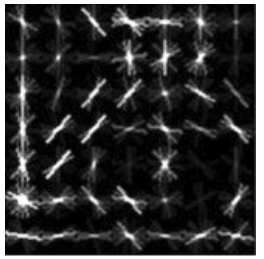
- Example: Image classification

Image set

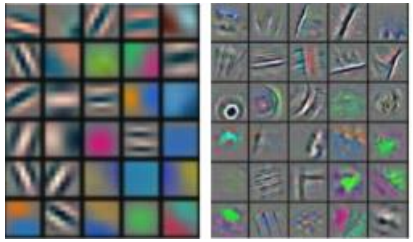
Different feature extractions



View 1: SIFT



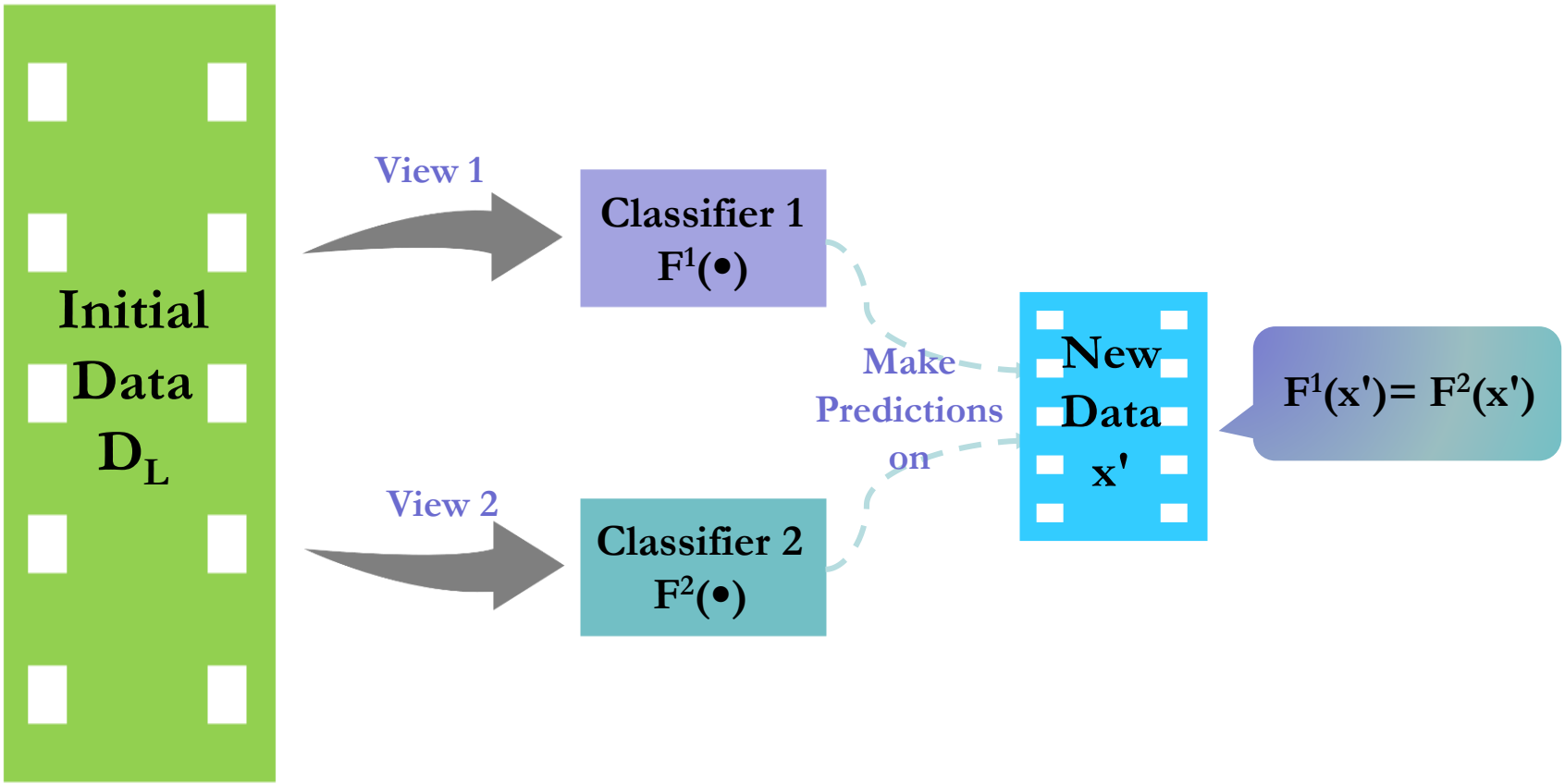
View 2: HOG



View 3: Deep Features

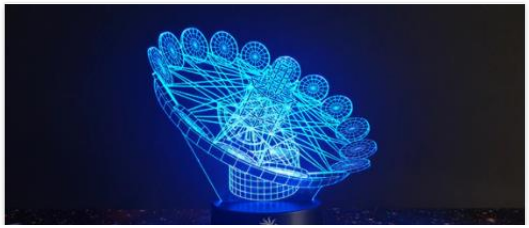
View 4: Contour Features

View Consistency



Crowdsourcing

- **What is crowdsourcing?**
  - Crowdfunding (Kickstarter)
  - Collective Knowledge (Data labeling, F)
  - 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

Workers

Items

	W	D	?	W	?	W
	D	D	D	D	?	D
	D	D	D	D	D	D
	W	W	D	?	W	?

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:** $W \in \mathbb{R}^{P \times N_w}$

$$W = \underbrace{\begin{bmatrix} \mathbf{w}_1^1 & \dots & \mathbf{w}_{N_w}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{w}_1^m & \dots & \mathbf{w}_{N_w}^m \\ \vdots & \ddots & \vdots \\ \mathbf{w}_1^V & \dots & \mathbf{w}_{N_w}^V \end{bmatrix}}_{N_w \text{ workers}} \begin{matrix} \} \ 1^{st}\text{-view} : \mathbb{R}^{d_1} \\ \\ \} \ m^{th}\text{-view} : \mathbb{R}^{d_m} \\ \\ \} \ V^{th}\text{-view} : \mathbb{R}^{d_V} \end{matrix}$$

where, $\mathbf{w}_k^m \in \mathbb{R}^{d_m}$ indicates the weights learned for the k[th] worker in the m[th] view.

M2VW: Formulation

❑ **Prediction tensor:** $\mathcal{A} \in \mathbb{R}^{N \times N_w \times V}$

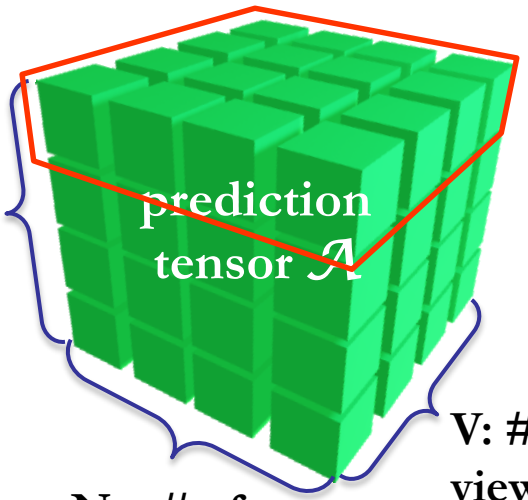
Vector of all 1's (of size V)

The i^{th} slice is defined as: $\mathcal{A}_{i::} = W^T[(\mathbf{x}_i \mathbf{e}_V^T) \circ B] \in \mathbb{R}^{N_w \times V}$

Feature vector of x_i

Block diagonal matrix

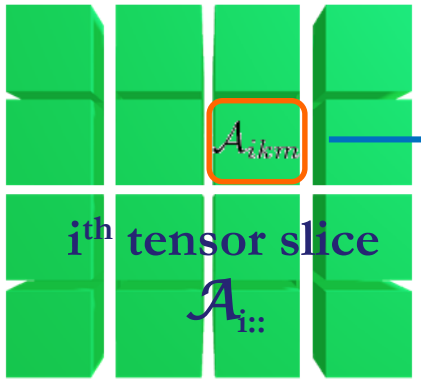
$B = \text{diag}(\mathbf{e}_{d_1}, \cdots, \mathbf{e}_{d_V})$



N: # of examples

N_w: # of workers

V: # of views



Prediction of the i^{th} example made by the k^{th} worker under the m^{th} view

M2VW: Formulation

❑ Optimization formulation:

$$\min_{W} \sum_{k=1}^{N_w} \sum_{i=1}^{N} \mathcal{L}(Y_{ik}, \mathbf{w}_k^T \mathbf{x}_i) + \text{Rank}(\mathcal{A}) + \mathcal{R}(W)$$

Loss function

Low-rank prediction tensor

Regularization term

❑ Remarks:

- Problem of rank minimization is NP-hard[1].
- Rank of a tensor is not uniquely defined[2,3].

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

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

Tightest convex envelope

Non-negative combination of matrices trace norms

[1]. E. J. Candes, et al, "Exact matrix completion via convex optimization," Foundations of Computational Mathematics, 2009.
[2]. T. G. Kolda, et al, "Tensor decompositions and applications," SIAM Review, 2009.
[3]. J. Liu, et al, "Tensor completion for estimating missing values in visual data," IEEE TPAMI, 2012.

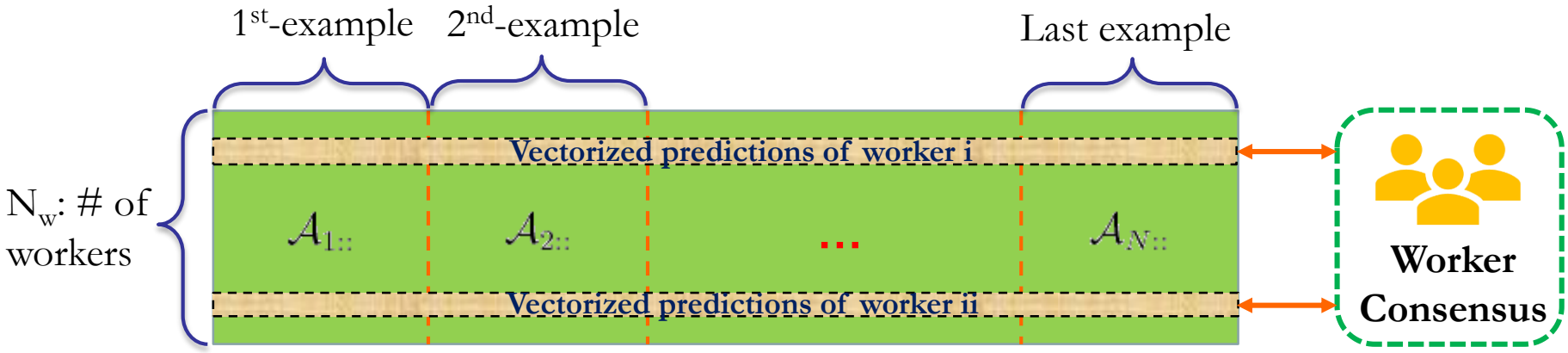
M2VW: Formulation

- **Interpretations of the terms:**
  - Loss function:

$$\mathcal{L}(Y_{ik}, \mathbf{w}_k^T \mathbf{x}_i) = \log\left(1 + \exp(-Y_{ik} \mathbf{w}_k^T \mathbf{x}_i)\right)$$

Convex and monotonically decreasing

  - First matricization: $\mathcal{A}_{(1)} \in \mathbb{R}^{N_w \times NV}$

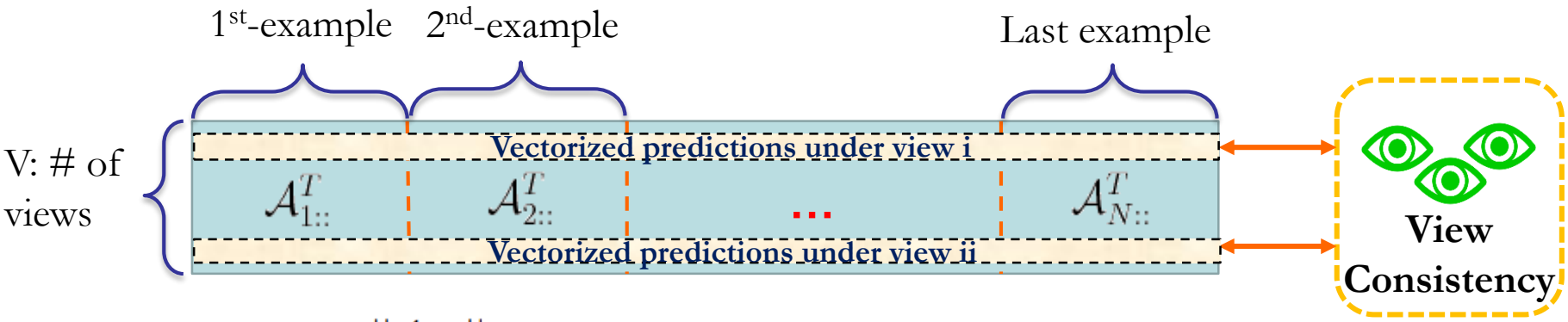


Minimizing $||\mathcal{A}_{(1)}||_*$ requires the predictions of **different workers** over the **same item** on the **same view** to be correlated.

M2VW: Formulation

❑ Interpretations of the terms:

- ➢ Second matricization: $\mathcal{A}_{(2)} \in \mathbb{R}^{V \times N_w N}$



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

❑ 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**.

M2VW: Formulation

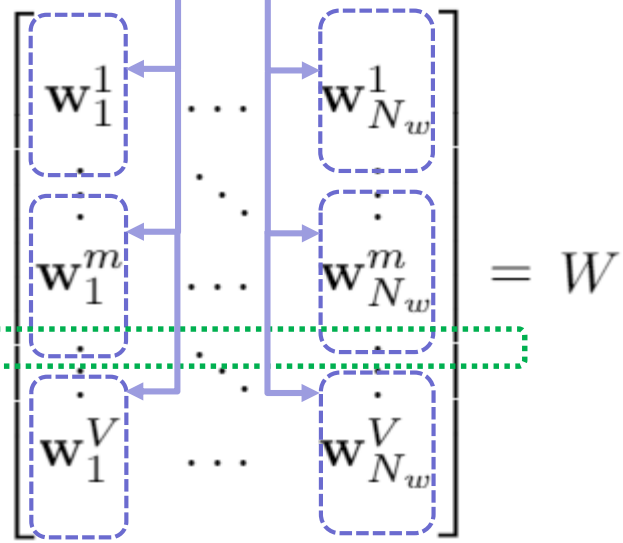
- **Interpretations of the terms:**
  - Regularization term: $\mathcal{R}(W) = ||W||_G + ||W||_{2,1}$

$$||W||_G = \sum_{k=1}^{N_W} \sum_{m=1}^{V} ||\mathbf{w}_k^m||_2$$

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)**

$$\min_{\{W,M_1,M_2\}} \sum_{k=1}^{N_w}\sum_{i=1}^{N} \log\left(1+\exp(-Y_{ik}\mathbf{w}_k^T\mathbf{x}_i)\right) + \lambda\left(||W||_G + ||W||_{2,1}\right) + \sum_{l=1}^{2} \alpha_l ||M_l||_* + \frac{\beta_l}{2}||\mathcal{A}_{(l)} - M_l||_F^2$$

- **Key advantages of this relaxation:**
  - The interdependent trace norm terms are split, so they can be solved **independently.**
  - Relaxation penalty term $||\mathcal{A}_{(l)} - M_l||_F^2$ 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:**

$$\min_{W} \underbrace{\sum_{k=1}^{N_w}\sum_{i=1}^{N}\log\left(1+\exp(-Y_{ik}\mathbf{w}_k^T\mathbf{x}_i)\right)}_{\text{logistic loss } \mathcal{L}(W)} + \underbrace{\sum_{l=1}^{2}\frac{\beta_l}{2}||\mathcal{A}_{(l)}-M_l||_F^2}_{\text{relaxation penalty } \mathcal{RP}(W)} + \lambda\underbrace{\left(||W||_G+||W||_{2,1}\right)}_{\text{feature sparsity } \mathcal{FS}(W)}$$

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

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

M2VW: Algorithm

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

$$\min_{W} \underbrace{\sum_{k=1}^{N_w}\sum_{i=1}^{N}\log\left(1+\exp(-Y_{ik}\mathbf{w}_k^T\mathbf{x}_i)\right)}_{\text{logistic loss } \mathcal{L}(W)} + \underbrace{\sum_{l=1}^{2}\frac{\beta_l}{2}||\mathcal{A}_{(l)}-M_l||_F^2}_{\text{relaxation penalty } \mathcal{RP}(W)} + \lambda\underbrace{\left(||W||_G+||W||_{2,1}\right)}_{\text{feature sparsity } \mathcal{FS}(W)}$$

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

$$\frac{\partial \mathcal{RP}(W)}{\partial \text{vec}(W^T)} = \text{vec}\left((\beta_1+\beta_2)W^TQQ^T-\beta_1M_1Q^T-\beta_2\mathbb{T}(M_2)Q^T\right)$$

Conclusion from Lemma 1 & Lemma 2

M2VW: Algorithm

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

$$\min_{W} \underbrace{\sum_{k=1}^{N_w}\sum_{i=1}^{N}\log\left(1+\exp(-Y_{ik}\mathbf{w}_k^T\mathbf{x}_i)\right)}_{\text{logistic loss } \mathcal{L}(W)} + \underbrace{\sum_{l=1}^{2}\frac{\beta_l}{2}||\mathcal{A}_{(l)}-M_l||_F^2}_{\text{relaxation penalty } \mathcal{RP}(W)} + \lambda\underbrace{\left(||W||_G+||W||_{2,1}\right)}_{\text{feature sparsity } \mathcal{FS}(W)}$$

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

$$\frac{\partial||W||_G}{\partial W_{k:}} = D_g^k W_{:k}, k=1,\cdots,N_w$$

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:**

$$\min_{M_l} : \frac{\alpha_l}{\beta_l}||M_l||_* + \frac{1}{2}||\mathcal{A}_{(l)} - M_l||_F^2$$

  - **Closed form solution[4]:**

$$D_\tau(\mathcal{A}_{(l)}) = U\Sigma_\tau V^T$$

where, $\Sigma_\tau = diag(\{\sigma_i - \tau\}_+)$, $\tau = \frac{\alpha_l}{\beta_l}$ and σ_i is the ith singular value of $\mathcal{A}_{(l)}$

[4]. J.-F. Cai, et al, "A singular value thresholding algorithm for matrix completion," SIAM Journal on Optimization, 2010.

M2VW: Algorithm

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

$$\mathcal{O}\Big(N_w NV^2 + n' N_w P(P + NV)\Big)$$

  - **Observation:**

The complexity is linear w.r.t. the number of workers N_w.

  - **Question:**

How to speed up?

M2VW: Randomized Algorithm

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

- **Remarks:**

$$\frac{\partial \mathcal{RP}(W)}{\partial \mathbf{w}_k} = (\beta_1 + \beta_2) Q Q^T \mathbf{w}_k - Q\left(\beta_1 \text{vec}(M_1^{(k)}) + \beta_2 \text{vec}(\mathbb{T}(M_2)^{(k)})\right)$$

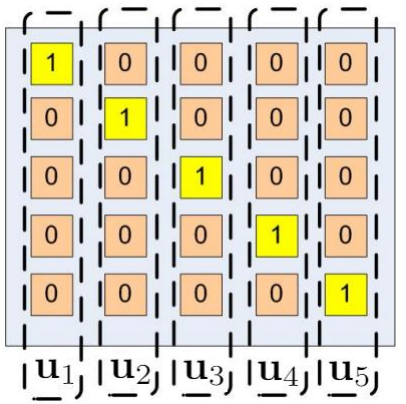
$$\frac{\partial \mathcal{L}(W)}{\partial \mathbf{w}_k} = -X \cdot \left[\frac{\mathrm{e}^{-Y_{:k} \circ (X^T \cdot \mathbf{w}_k)}}{1 + \mathrm{e}^{-Y_{:k} \circ (X^T \cdot \mathbf{w}_k)}} \circ Y_{:k}\right]$$

$$\frac{\partial \mathcal{FS}(W)}{\partial \mathbf{w}_k} = D_g^k \mathbf{w}_k + D_s \mathbf{w}_k$$

M2VW: Randomized Algorithm

- **Decomposition of $\mathbf{W}$ into $\mathbf{N_w}$ blocks:**
  - Any $\mathbf{W}$ can be written uniquely as:

$$W = \sum_{k=1}^{N_w} \mathbf{w}_k \mathbf{u}_k^T$$



Block coordinate directions

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

$$W \leftarrow W - \frac{\partial f(W)}{\partial \mathbf{w}_k} \mathbf{u}_k^T$$

$$W = \begin{bmatrix} | & \vdots & | & \vdots & | \\ \mathbf{w}_1 & & \mathbf{w}_k & & \mathbf{w}_{N_w} \\ | & \vdots & | & \vdots & | \end{bmatrix}$$

Worker blocks of $\mathbf{W}$

M2VW: Randomized Algorithm

- **Batch workers update:**
  - For n[th] round of the BCD iterations, and assume that we select a subset of the block coordinate directions: $N_b = \{\mathbf{u}_k | \, k \in [N_w]\}$

$$W \leftarrow W - \sum_{k}^{N_b} \Big(\frac{\partial \mathcal{L}(W)}{\partial \mathbf{w}_k} + \frac{\partial \mathcal{RP}(W)}{\partial \mathbf{w}_k} + \lambda \frac{\partial \mathcal{FS}(W)}{\partial \mathbf{w}_k} \Big) \mathbf{u}_k^T$$

- **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.

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

[5]. T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, Computer Science Technical Report CMU-CS-96-118. Carnegie Mellon University.
[6]. Y. Zhou, et al, "MultiC^2 : an optimization framework for learning from task and worker dual heterogeneity" in SDM, 2017.

Experiment

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

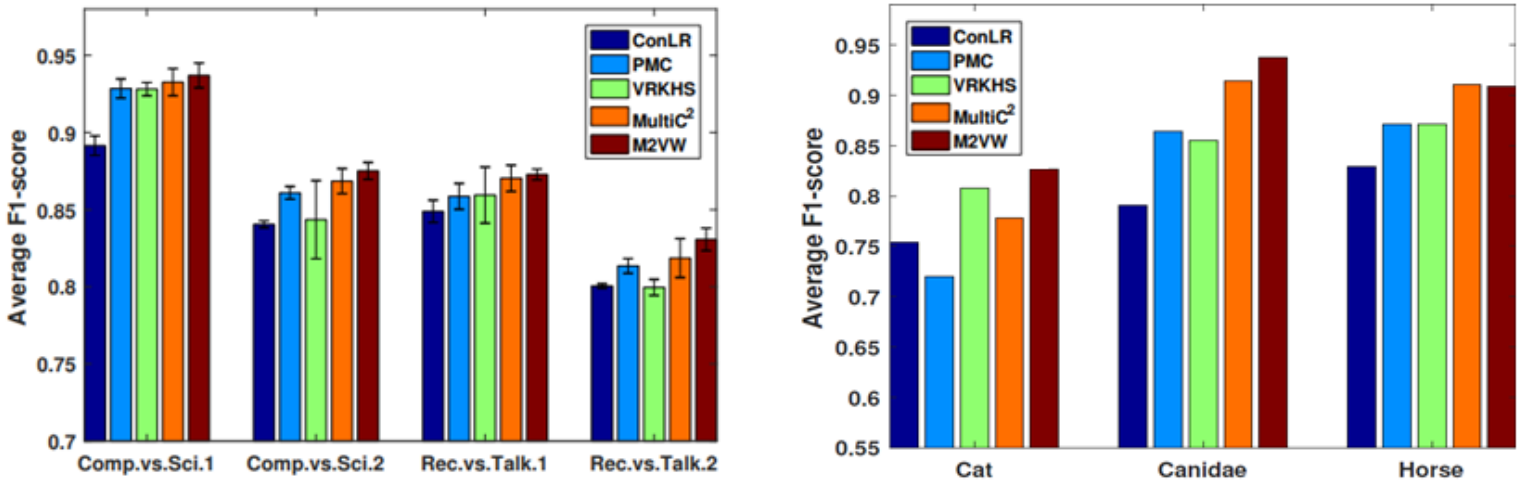


Figure 1: Left: 20 Newsgroups data set (10 random runs of train/test splits). Right: Animal data set.

[7]. M. Chen, et al, "Automatic feature decomposition for single view co-training," in ICML, 2011.

[8]. H. Q. Minh, et al, "A unifying framework for vector-valued manifold regularization and multi-view learning," in ICML, 2013.

Experiment

- Terms necessities and parameter sensitivity:

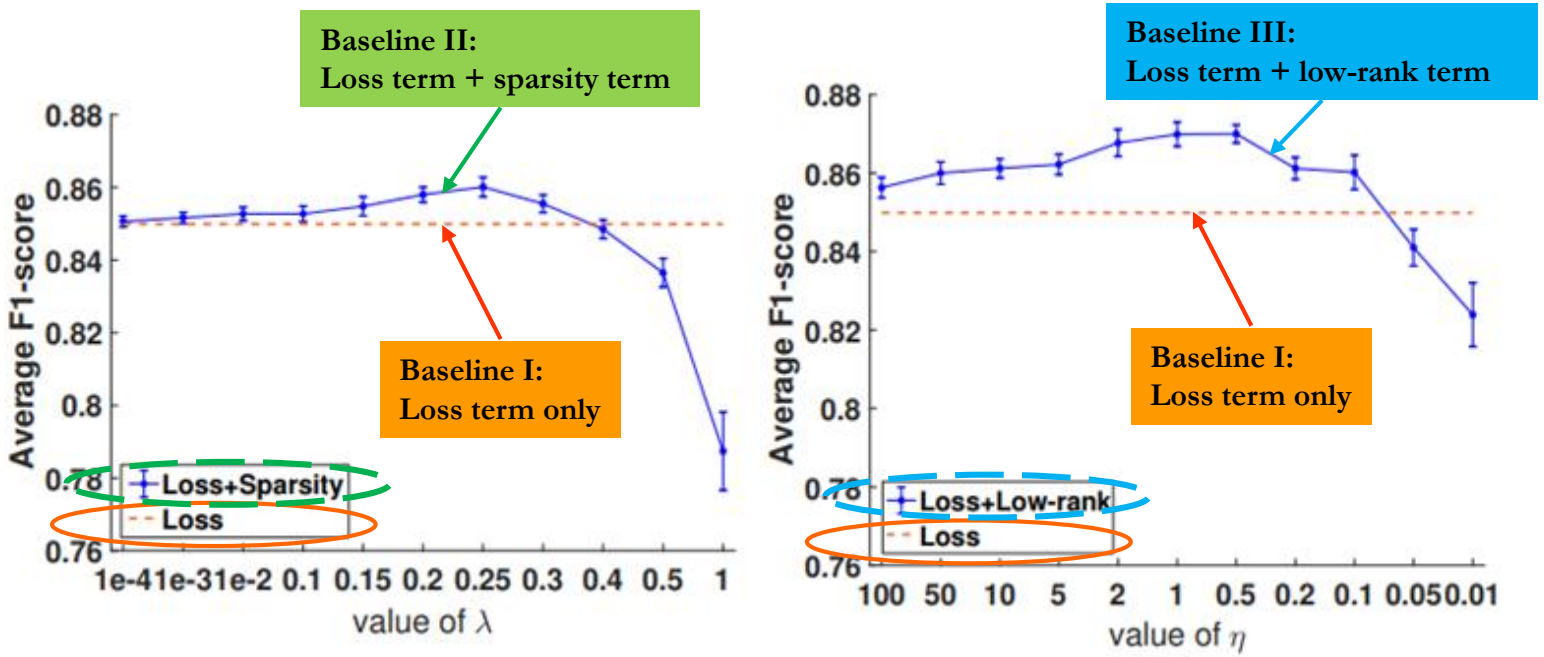
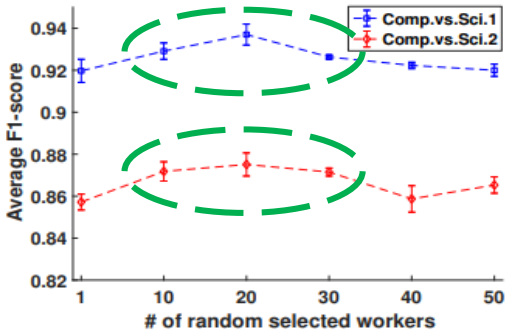


Figure 2: Left: necessity of feature sparsity term. Right: necessity of low-rank prediction tensor term.

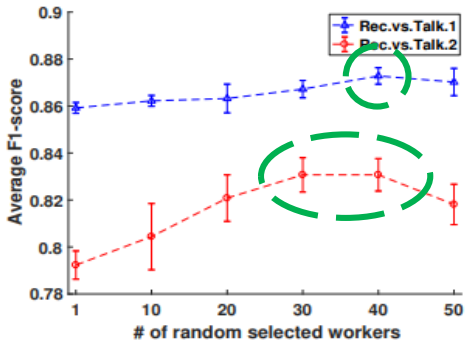
Experiment

❑ Efficiency:

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

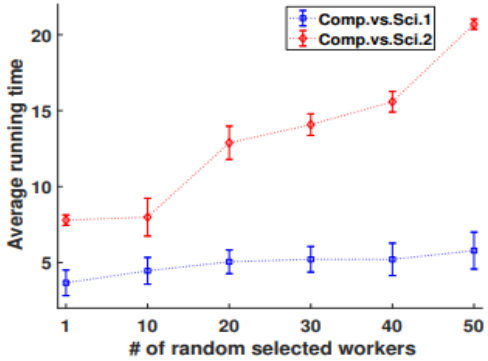


(b) Performance (F1-score) on *Rec. vs. Talk*

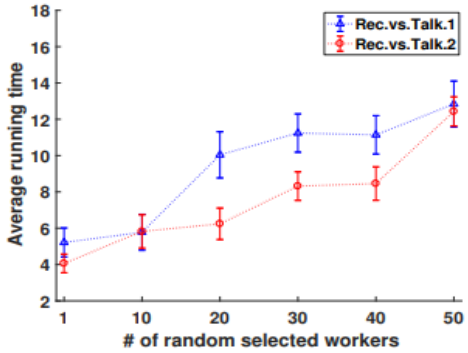


Improved performance with proper worker batch size

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



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



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

Figure 4: Efficiency of Batch-RBCD

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?