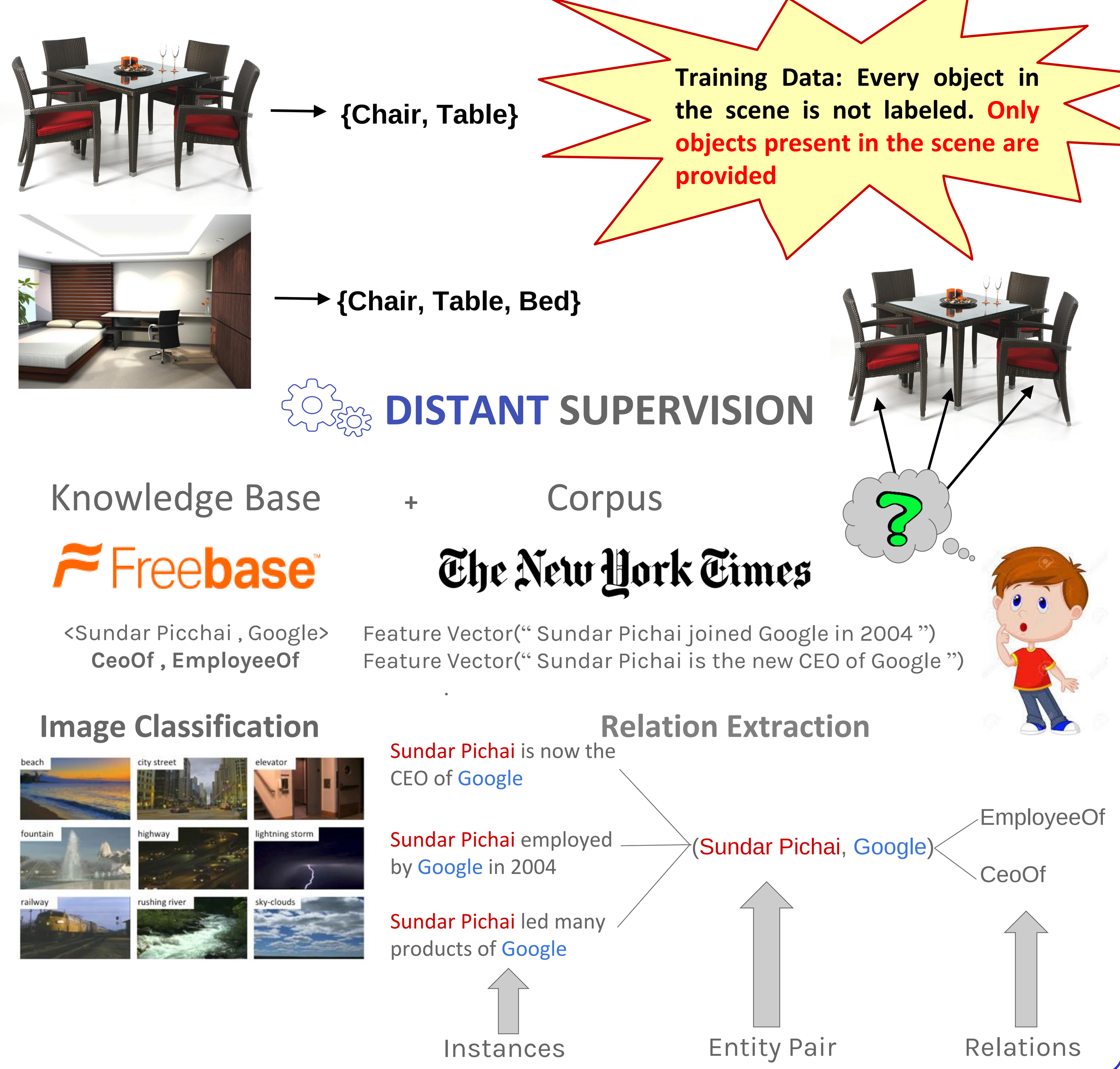
Scalable Optimization of Multivariate Performance Measures in Multi-instance Multi-label Learning

Apoorv Aggarwal*, Sandip Ghoshal*, Ankith M S*, Suhit Sinha*, Ganesh Ramakrishnan*, Purushottam Kar#, Prateek Jain†

*Indian Institute of Technology Bombay, INDIA
#Indian Institute of Technology Kanpur, INDIA
†Microsoft Research, INDIA

Goal: Scalable algorithm for optimizing multivariate performance measures for multi-instance multi-label (MIML) learning problems

Multi Instance Multi Label Learning



Performance Measures

Univariate or Decomposable Measures:

- Ill suited in the presence of label imbalance or a heavy tailed label distribution. Tend to neglect performance on rare labels

Multivariate Performance Measures:

- Typically non-decomposable as their evaluation does not decompose over individual points
- Some performance measures such as F-macro force predictor to do well on rare labels as well.

Non Decomposable Loss Functions

Dataset: $\{(x_i, y_i)\}_{i=1}^N \left\{ \begin{array}{l} x_i = \{x_i^{(1)}, \ldots, x_i^{(n_i)}\} \in X \\ y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,L}] \in Y = \{0,1\}^L \end{array} \right.$

Macro F-measure:

- **Class-wise F-measure**

$$\boxed{F_\beta^j(\mathrm{f}; X, Y)} := \left(\frac{\beta}{\mathrm{Prec}^j(\mathrm{f}; X, Y)} + \frac{1-\beta}{\mathrm{Rec}^j(\mathrm{f}; X, Y)}\right)^{-1}$$

- Averaged over all classes

$$F_\beta^{macro}(\mathrm{f}; X, Y) := \frac{1}{L}\sum_{j=1}^{L} \boxed{F_\beta^j(\mathrm{f}; X, Y)}$$

Do well on rare labels

Micro F-measure:

$$F_\beta^{micro}(\mathrm{f}; X, Y) := \left(\frac{\beta}{\mathrm{Prec}(\mathrm{f}; X, Y)} + \frac{1-\beta}{\mathrm{Rec}(\mathrm{f}; X, Y)}\right)^{-1}$$

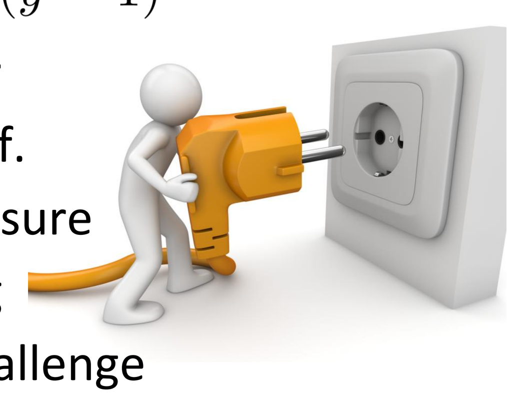
MIML_perf: A Novel Plugin Classifier Learning Framework

Drawbacks of Existing Approaches

- Training Objective ≠ Evaluation Measure
- Not scalable to large, web scale datasets
- Ill suited for label imbalance problems

Plug-in Classifiers

- Learn a CPE model to predict $g(x) \approx P(y = 1)$
- Tune a threshold η to obtain a classifier $f(x) = \mathrm{sign}(g(x) - \eta)$ to maximize perf. measure Δ, e.g. classfn accuracy, F-measure
- Challenge: lack of instance level training in MIML. Learning CPE model itself a challenge



MIML_perf

- Solution: EM-style alternating approach
- Model instance level labels using latent variables
- $z_k^{(i,j)}$ models if instance k in bag i expresses label j or not
- Alternately, improve latent variable assign and model

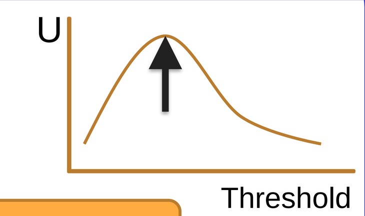
- Fix hidden variables, update CPE models
  - CPE-train ($D^j \leftarrow \{\{(x_i^{(k)}, z_k^{(i,j)})\}_{k=1}^{n_i}\}_{i=1}^{n}$)
- Tune threshold(s) s.t. Δ is maximized (EUM)
- Fix plug-in models, update hidden variables
  - Re-estimate hidden labels probabilistically according to CPE-train scores

CPE-Train

- Treat hidden variables as instance level labels $z_k^{(i,j)} \in \{0,1\}$
- Train a CPE classifier g^j for each label j
- Use $g^j(x_i^{(k)})$ to model probability that the k^{th} instance in bag i expresses label j

Tuning the Thresholds [EUM Approach]

- Optimizing F-macro → threshold η_j for label j
- Optimizing F-micro → Single threshold η



I_1, I_2, I_3 → Aggregation → Find Threshold

$$\hat{y}_j \leftarrow \bigvee_{k=1}^{n_t} I\{j : g^j(x^{(k)}) \geq \eta_j\}$$

(Instance Level) (Bag Level, label j) (Maximize Performance Measure)

Speed up while tuning the thresholds

- No. of CPE Scores = No. of instances → Millions
- Tune the threshold over the largest CPE Scores in the bags only — Monotonicity property

(no. of instances) $\sum_{i=1}^{N} n_i \rightarrow N$ (no. of bags)

- F-scores can be written as $f(\mathrm{TP}, \mathrm{TN})$
- Sort the CPEs and do a linear scan to find the best threshold

$$O(N^2 \cdot L) \rightarrow O(N \log N \cdot L)$$

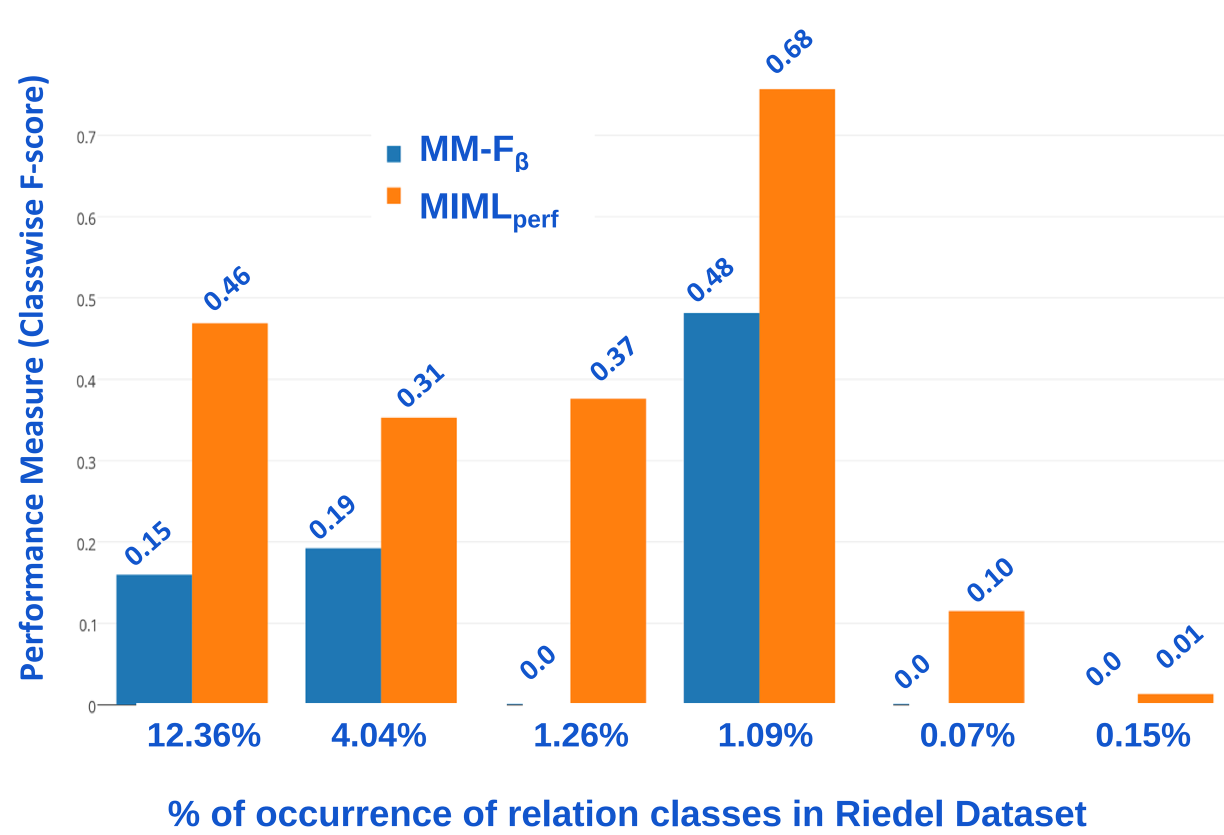
Estimating the Hidden Variables

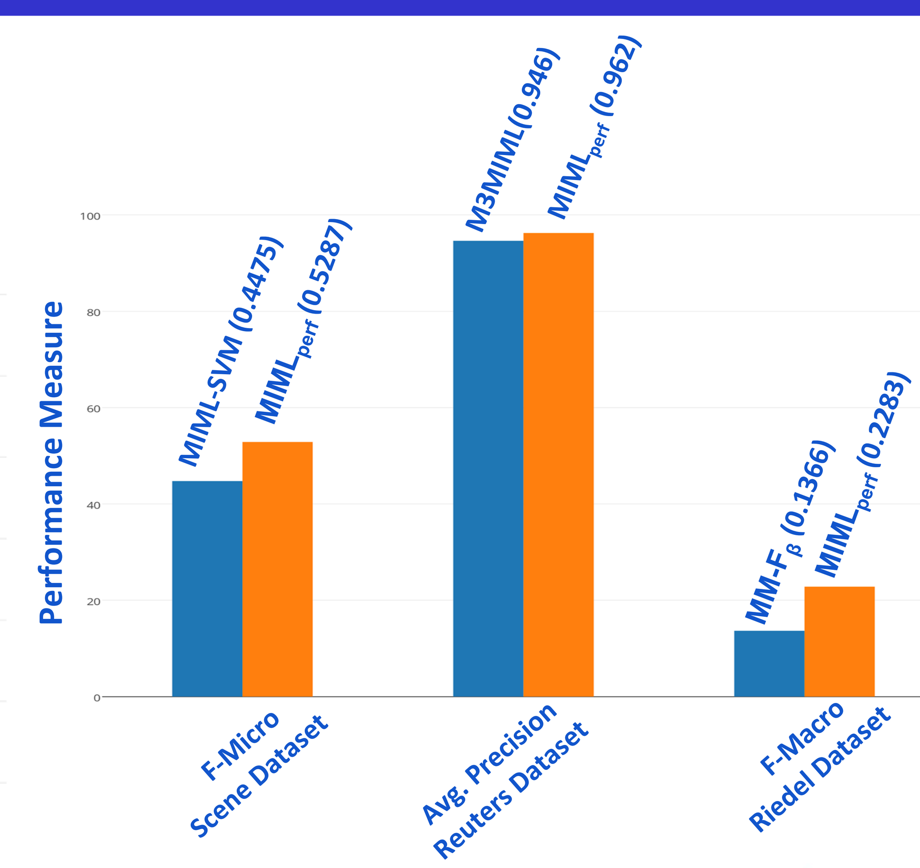
- If bag i does not have label j → $z^{(i,j)} = 0$ (extreme sparsity)
- If bag i has label j, choose $c^{(i,j)}$ instances probabilistically according to CPE scores and make them 1
- Choice of $c^{(i,j)}$

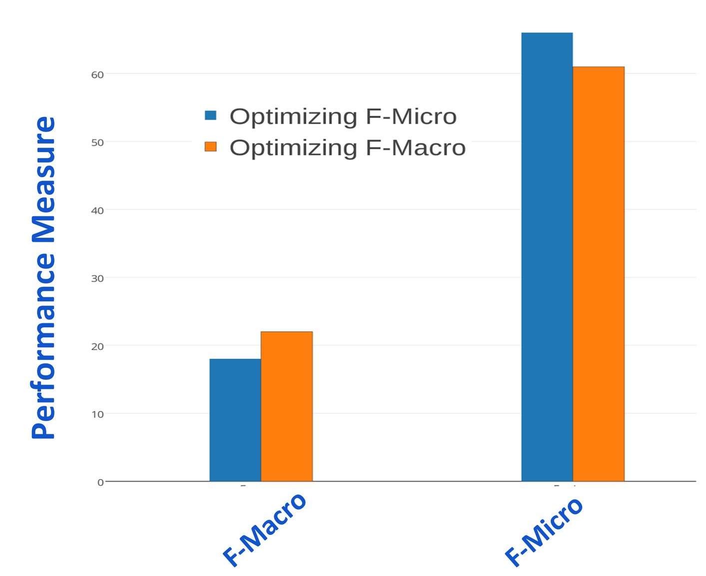
  Initialization – $c^{(i,j)} \leftarrow k.n_i$

  Subsequently – $c^{(i,j)} \leftarrow \sum_{k=1}^{n_i} \mathbb{I}[g^j(x_i^{(k)}) \geq \eta_j]$

Experiments







Effect of the training objective on test performance on the Riedel dataset

MM-Fβ → MIML^perf