Condensed Filter Tree for Cost-Sensitive Multi-Label Classification

Chun-Liang Li and Hsuan-Tien Lin

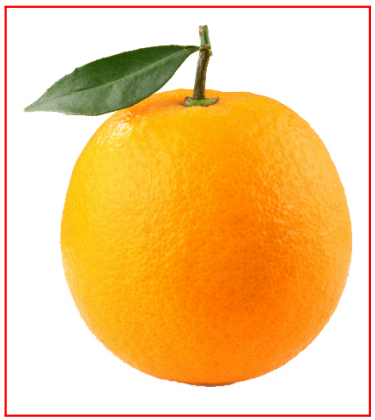
Department of Computer Science and Information Engineering
National Taiwan University

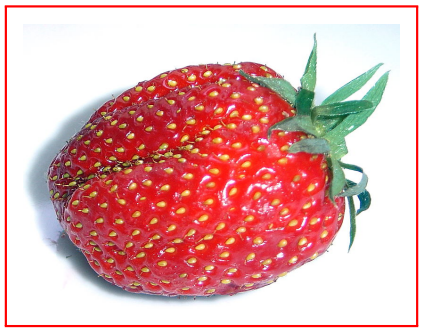
June 22, 2014 (ICML)

What is it

Multi-label classification

One instance is associated multiple labels





Multi-Label Classification

Given

N examples $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$

- Feature vector $\mathbf{x} \in \mathbb{R}^d$
- Label vector $\mathbf{y} \in \{0, 1\}^K$

$\mathbf{y} =$	0	1	1	1
$\mathcal{Y} =$	{Apple	Kiwi	Orange	Strawberry}

Goal

Find a classifier $h\colon \mathbb{R}^d \to \{0, 1\}^K$ which **closely** predicts the label vector $\mathbf{y}$

Performance Evaluations

Number of Error Labels

- **Hamming Loss** $\frac{1}{K}\sum_{k=1}^{K}[\![\mathbf{y}[k] = \hat{\mathbf{y}}[k]]\!]$
- May be meaningless, why? Sparsity!

- **Rank Loss** $\sum_{\mathbf{y}[i]\rhd\mathbf{y}[j]}\left([\![\hat{\mathbf{y}}[i] < \hat{\mathbf{y}}[j]]\!] + \frac{1}{2}[\![\hat{\mathbf{y}}[i] = \hat{\mathbf{y}}[j]]\!]\right)$
- **F1 score** $\frac{1}{K}\frac{2\|\mathbf{y}\cap\hat{\mathbf{y}}\|_1}{\|\mathbf{y}\|_1 + \|\hat{\mathbf{y}}\|_1}$
- and so on

How to optimize? Can we have a **general description**?

Cost-Sensitive Multi-Label Classification (CSMLC)

Given

- $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$
- A cost matrix (function) $C : \{0,1\}^K \times \{0,1\}^K \to \mathbb{R}$

Hamming Loss with 2 labels

Answer \ Prediction	00	01	10	11
00				
01	$\frac{1}{2}$	0	$\frac{2}{2}$	$\frac{1}{2}$
10				
11				

Goal

Minimize average cost given any cost matrix (evaluation criterion).

The Main Focus of this work

Our Contributions

Existing General Framework

Framework	Additional Efforts
Probabilistic Classifier Chain	Efficient Inference rule
Structure SVM	Efficient Maximizer

The Proposed Algorithm

- **without** additional design effort
- **superior** empirical performance

Preliminary Approach

Cost-Sensitive Multi-Class

- Well-Studied (Beygelzimer et al., 2005, 2007)
- Define cost between classes

+

Label Powerset

- **Multi-Label** to **Multi-Class**
- Transform each $\{0,1\}^K$ to a unique class
- Ex: {00, 01, 10, 11} $\Rightarrow$ {1,2,3,4}

Simple Idea: Cost-Sensitive Multi-Class + Label Powerset

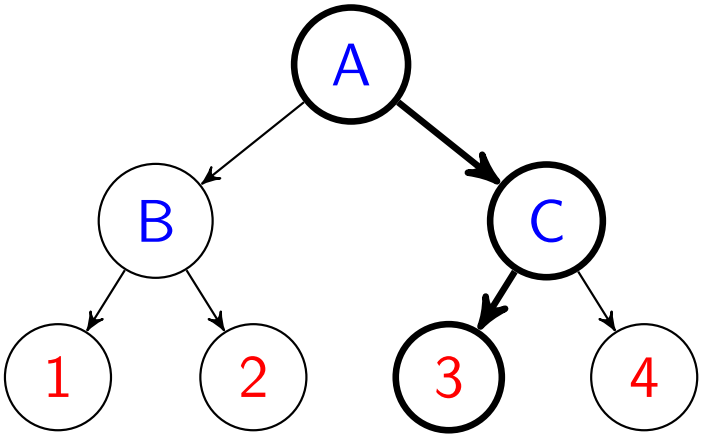
1. Reduce multi-label to multi-class via label powerset
2. Describe the evaluation criterion by cost matrix
3. Apply any cost-sensitive multi-class algorithm

Advantage: Optimize any loss without specific design effort

Challenge: **#Classifiers, Prediction, Training** with $O(2^K)$ classes

Utilize the Tree Structure (Beygelzimer et al., 2007)

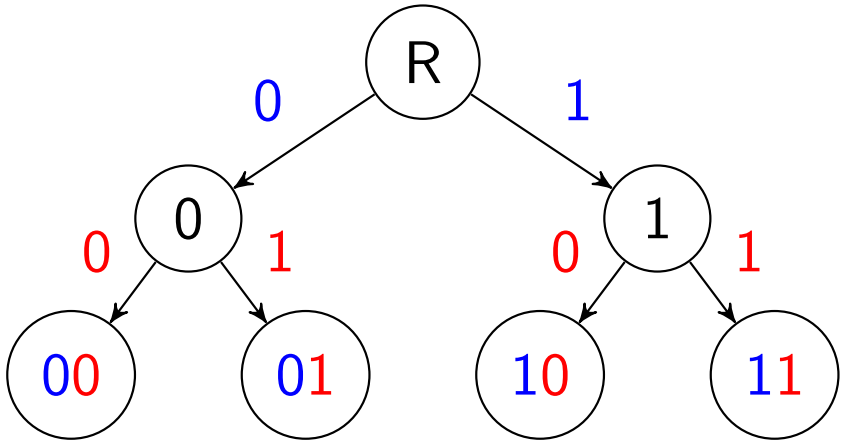
Weighted binary classifier
for cost-sensitive multiclass



extended classes

- Predict in $\log(2^K) = O(K)$
- **Challenge:** $O(2^K)$ classifiers for all internal nodes

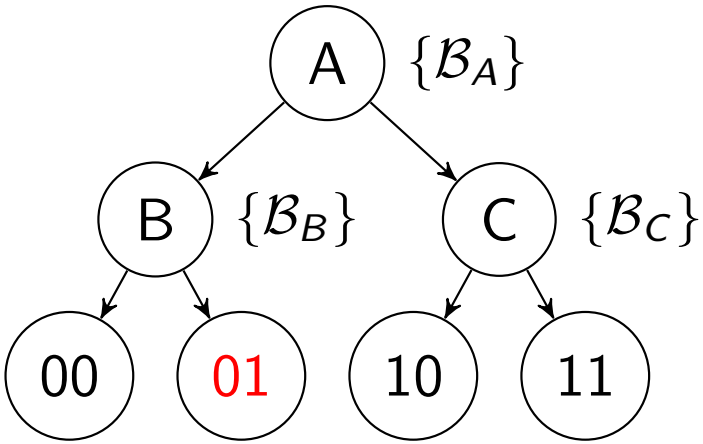
Properly order classes (vectors) and **relabel** the internal nodes



K-Classifier Trick

- k-th layer predicts k-th label $\mathbf{y}[k]$
- Let each layer **share** 1 classifier $h_k(node, \mathbf{x})$
- Reduce $O(2^K)$ classifiers to K classifiers

Train K classifiers - Filter Tree (Beygelzimer et al., 2007)



Filter Tree

- A cost-sensitive multi-class algorithm
- A bottom-up training (regret is bounded)
- Use all training data to generate binary classification data $\mathcal{B}$ for each nodes

K-Classifier trick

- $\mathcal{B}_A$ for layer 1
- $\mathcal{B}_B$ and $\mathcal{B}_C$ for layer 2
- Layer k should consider 2^{k-1} sets
- **Not solve** training difficulty

Naive Approach

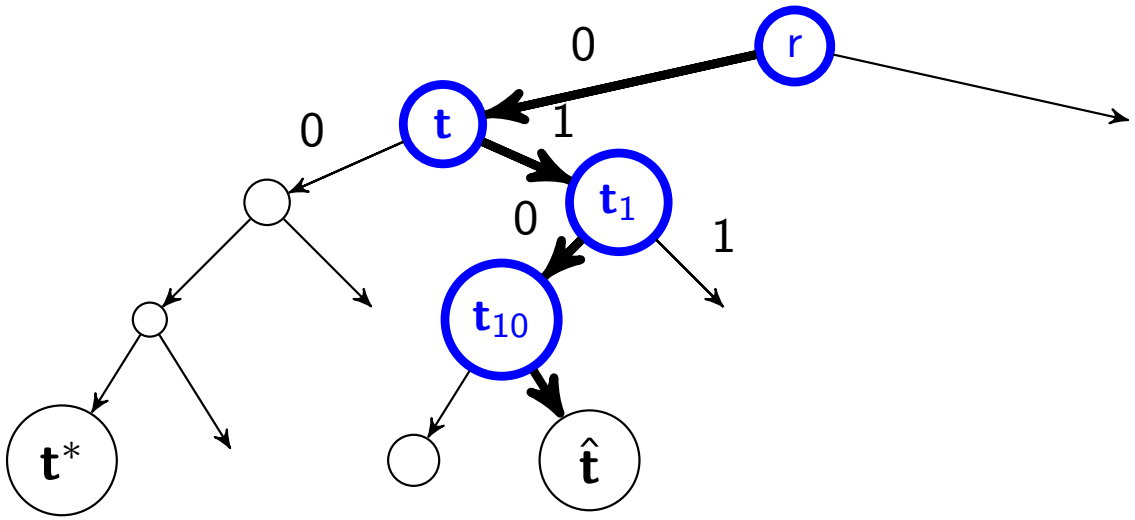
- Infeasible to consider all nodes
- **Uniformly sample** M nodes for training

Condensed Filter Tree

Not every node is important

Intuition

- Ignore the nodes you **won't visit**
- Focus on what you **really meet**



A Greedy Manner

1. Add the nodes on the **prediction path**
2. Retrain on them to get better classifiers

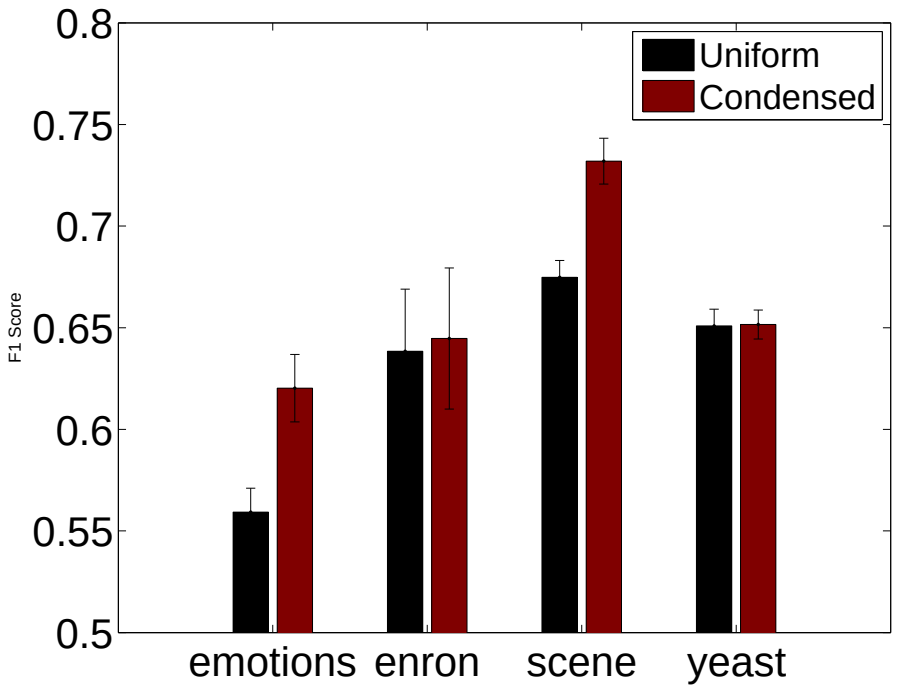
- Can repeat this process several times
- Theoretical guarantee

Experiment I (the higher the better)

Settings

- 80% data for training, 20% for testing, repeat 40 times
- Tune parameters by 5-fold CV, apply to 9 datasets (partially report)

F1 Score $\frac{1}{K}\frac{2\|\mathbf{y}\cap\hat{\mathbf{y}}\|_1}{\|\mathbf{y}\|_1+\|\hat{\mathbf{y}}\|_1}$ (the **higher** the better)

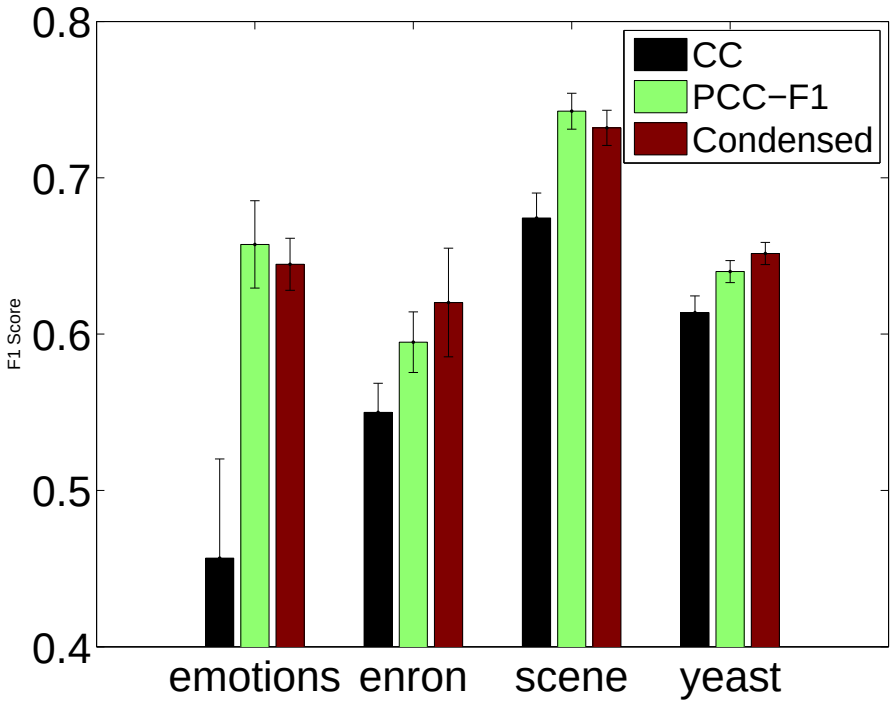


Consider prediction path is useful and **CFT has superior performance**

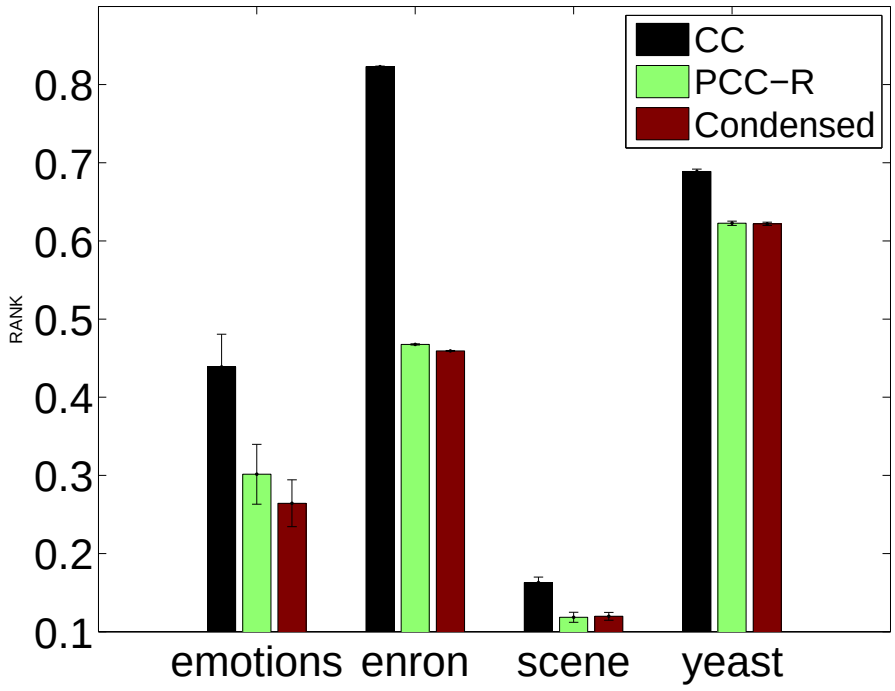
Experiment II

Criteria with Efficient Designs of PCC

F1 Score $= \frac{1}{K} \frac{2\|\mathbf{y} \cap \hat{\mathbf{y}}\|_1}{\|\mathbf{y}\|_1 + \|\hat{\mathbf{y}}\|_1}$ (the **higher** the better)



Rank Loss $= \frac{1}{K} \sum_{\mathbf{y}[i] \rhd \mathbf{y}[j]} \left([\![\hat{\mathbf{y}}[i] < \hat{\mathbf{y}}[j]]\!] + \frac{1}{2} [\![\hat{\mathbf{y}}[i] = \hat{\mathbf{y}}[j]]\!] \right)$ (the **lower** the better)

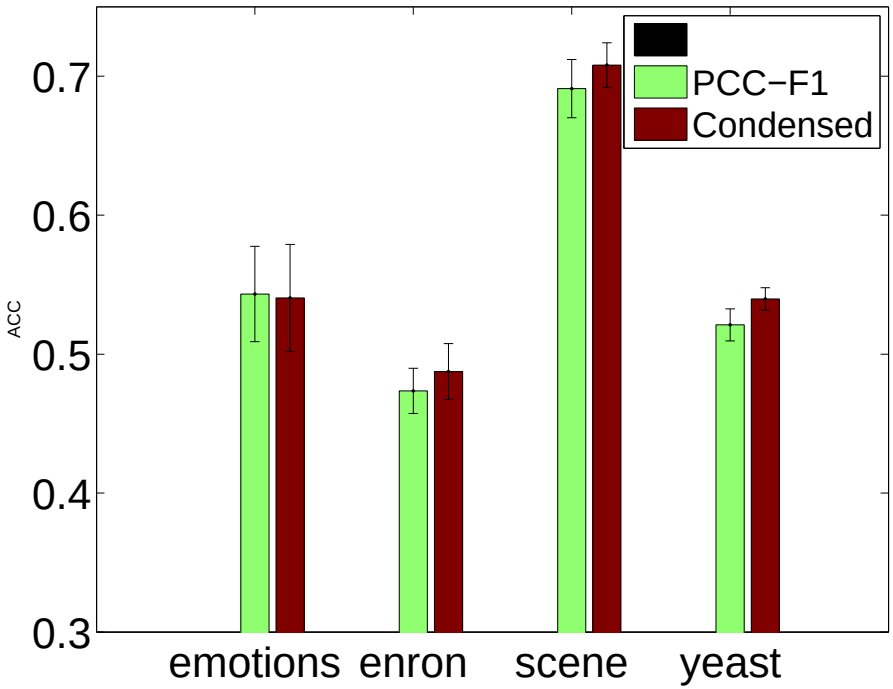


Competitive Performance

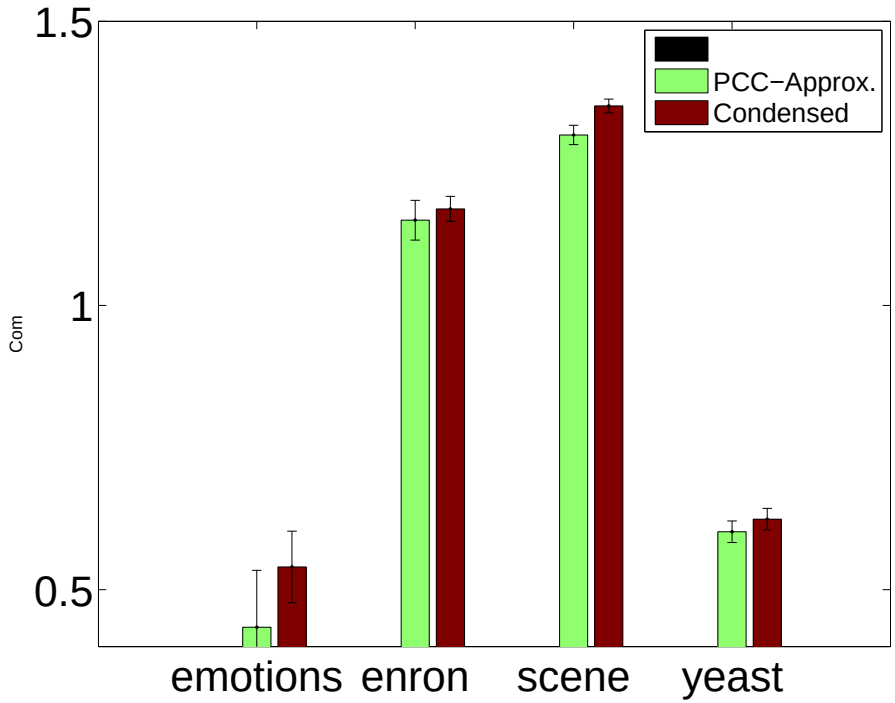
Experiment III

Criteria without Efficient Designs of PCC

Accuracy$=\frac{1}{K}\frac{\|\mathbf{y}\cap\hat{\mathbf{y}}\|_1}{\|\mathbf{y}\cup\hat{\mathbf{y}}\|_1}$
(the **higher** the better)



Composited Score
$=$ F1 Score $-5\times$ HAM Loss $+1$.
(the **higher** the better)



Better Performance

Conclusion

- Propose cost-sensitive multi-label classification to **unify different evaluation criteria** for designing a general algorithm
- Utilize the **tree structure** with K-classifier trick to **solve difficulty** of **prediction** and **representation**
- **Select key nodes** for **training efficiently** and **effectively** with theoretical guarantee
- Better or Competitive performance with specific designed algorithms

Thank you! Questions?