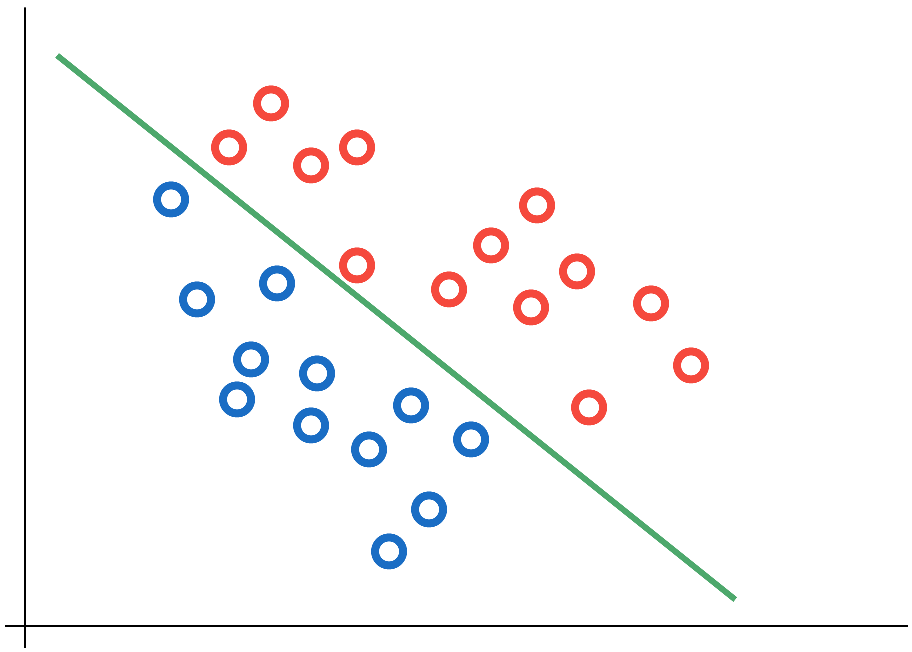
# An Efficient Probabilistic Framework for Multi-Dimensional Classification

Iyad Batal Charmgil Hong Milos Hauskrecht

Department of Computer Science
University of Pittsburgh

# Motivation

- Traditional classification
  - Each data instance is associated with a single class variable

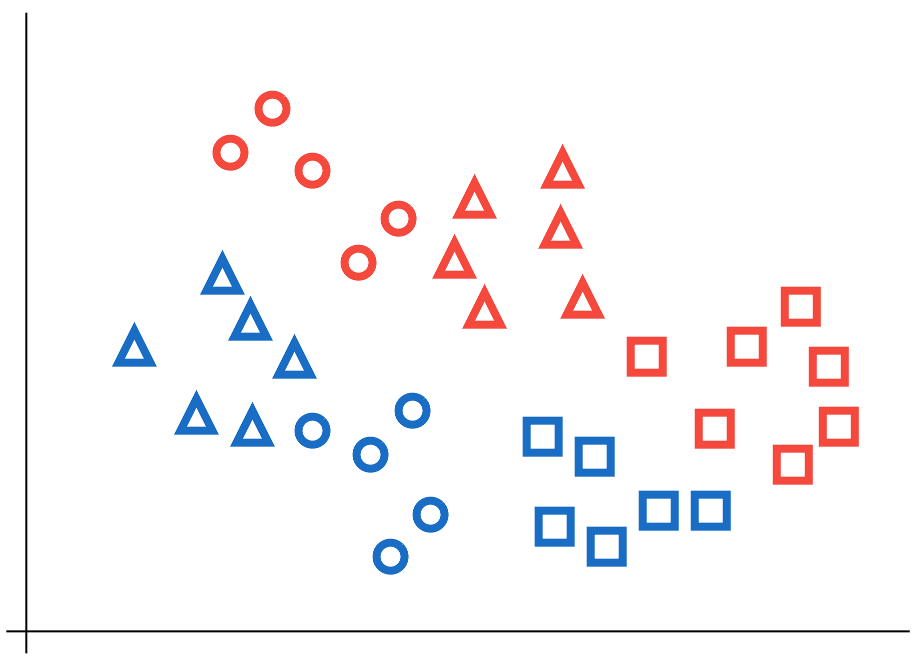


# Motivation

- Multi-dimensional classification
  - In many real-world applications, each data instance can be associated with multiple class variables
  - Examples
    - A news article may cover multiple topics such as *politics* and *economy*
    - An image may include multiple objects as *building*, *road* and *car*
    - A gene may be associated with several biological functions

# Motivation

- Multi-dimensional classification
  - Each data instance is associated with multiple class variables
  - Objective: assign to each instance the most probable assignment of the class variables



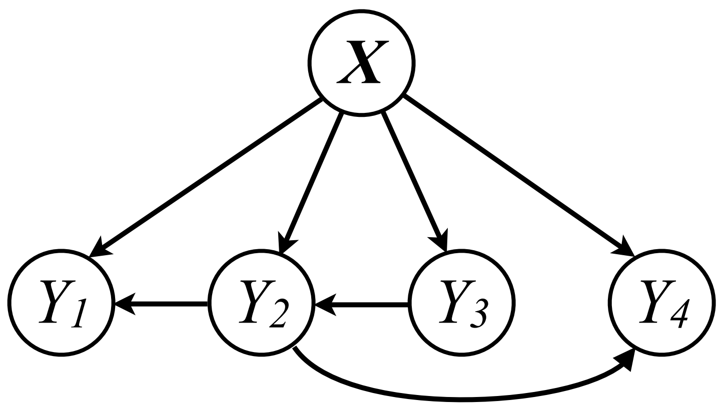
Class 1 $\in$ { red, blue }
Class 2 $\in$ { ○, △, □ }

# Motivation

- Simplest solution
  - Learning $d$ independent classifiers for $d$ class labels
  - It does not capture the dependency relations between the classes

# CTBN

- Conditional Tree-structured Bayesian Network (CTBN) for modeling $\mathrm{P}(Y_1, \ldots, Y_d | \boldsymbol{X})$
  - Each class variable can have at most one other class variable as a parent (the classes form a directed tree)
  - The feature vector $\boldsymbol{X}$ is the common parent for all class variables




*An example CTBN*

# CTBN

- Conditional Tree-structured Bayesian Network (CTBN) for modeling $\mathrm{P}(Y_1, \ldots, Y_d | \boldsymbol{X})$
  - Each class variable can have at most one other class variable as a parent (the classes form a directed tree)
  - The feature vector $\boldsymbol{X}$ is the common parent for all class variables
- We restrict the dependency structure to a tree because:
  1. The optimal structure can be learned efficiently (coming up)
  2. Exact inference can be done in $\mathrm{O}(d)$ time (please refer the paper)

# Representation

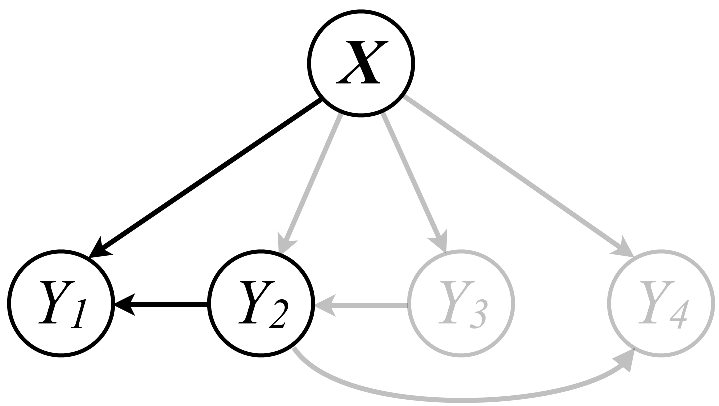
- The conditional class distribution is:

$$P(y_1, \ldots, y_d|\mathbf{x}) = \prod_{i=1}^{d} P(y_i|\mathbf{x}, y_{\pi(i,T)})$$

the parent of $y_i$ in CTBN $T$

  - It is the product of the dependencies in the network

$P(Y_1|\boldsymbol{X}, Y_2)$




*An example CTBN*

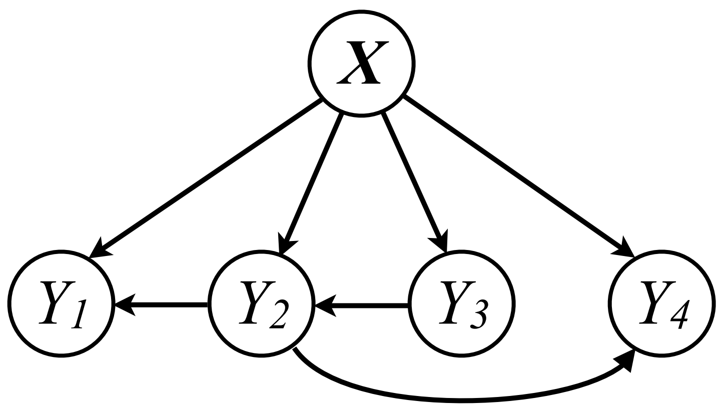
# Representation

- The conditional class distribution is:

$$P(y_1, ..., y_d|\mathbf{x}) = \prod_{i=1}^{d} P(y_i|\mathbf{x}, y_{\pi(i,T)})$$

the parent of $y_i$ in CTBN $T$

  - It is the product of the dependencies in the network




This network represents
$$P(y_1, y_2, y_3, y_4|\mathbf{x}) = P(y_3|\mathbf{x}) \cdot P(y_2|\mathbf{x}, y_3) \cdot P(y_1|\mathbf{x}, y_2) \cdot P(y_4|\mathbf{x}, y_2)$$

# Representation

- The conditional class distribution is:

$$P(y_1, \ldots, y_d | \mathbf{x}) = \prod_{i=1}^{d} P(y_i | \mathbf{x}, y_{\pi(i,T)})$$

  - It is the product of the dependencies in the network

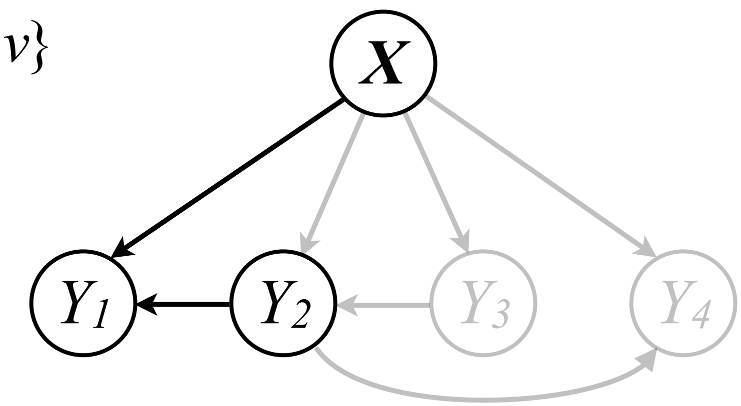
- Each $P(y_i \,|\, \boldsymbol{x}, y_{\pi(i,T)})$ is represented by classifier functions.

$Y_2 \in \{1, \ldots, v\}$

$f_{1,0}(X) = P(Y_1 | \mathbf{X}, Y_2 = 1)$

$\ldots$

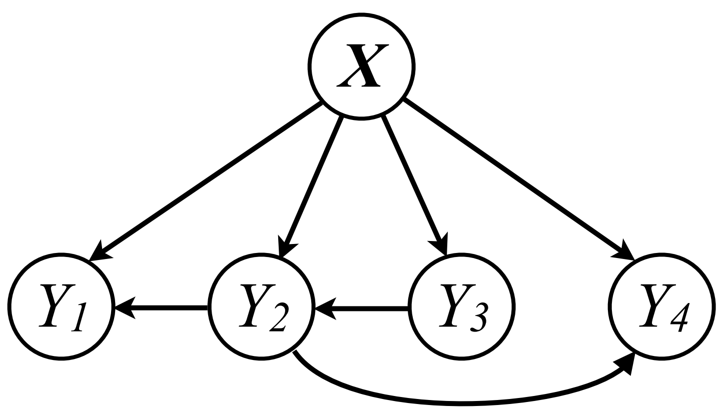
$f_{1,v}(X) = P(Y_1 | \mathbf{X}, Y_2 = v)$




For each class $Y_i$, we learn a different probabilistic classifier for each possible value $v$ of the parent class

# Structure learning

- Objective: Find the tree structure that best approximates $P(\boldsymbol{Y}|\boldsymbol{X})$, i.e., that maximizes the conditional log-likelihood of data
- Idea: Cast the learning into the maximum branching tree problem
- Next: illustration through the example CTBN

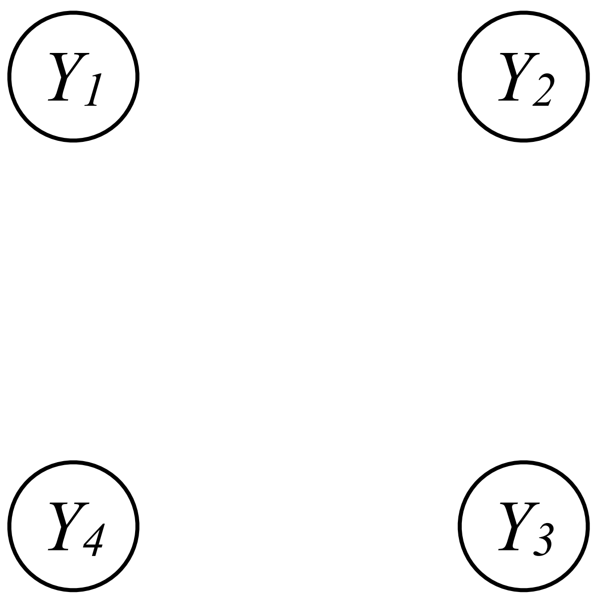


# Structure learning

1. Define a complete weighted directed graph $G$

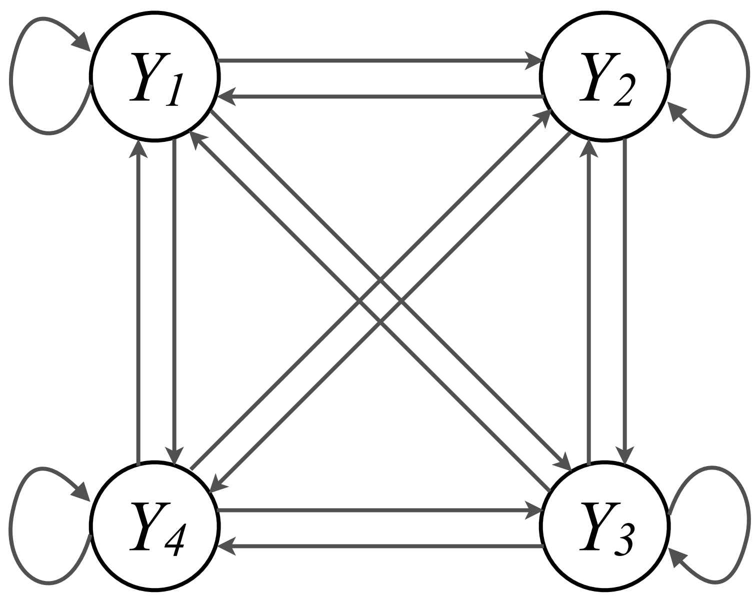
# Structure learning

1. Define a complete weighted directed graph $G$
   - Draw $d$ nodes for all class variables $Y_i : i \in \{1, ..., d\}$



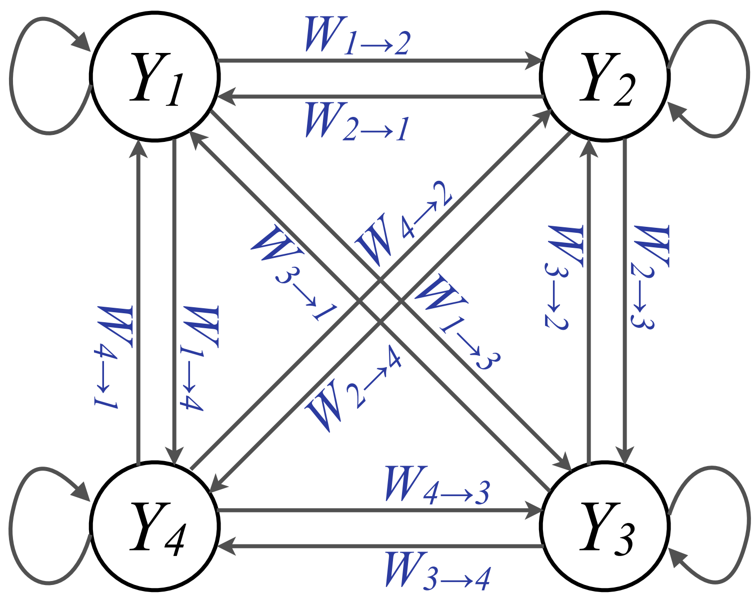
# Structure learning

1. Define a complete weighted directed graph $G$
   - Draw $d$ nodes for all class variables $Y_i : i \in \{1, ..., d\}$
   - Connect all the node pairs and add self-loops with directed edges



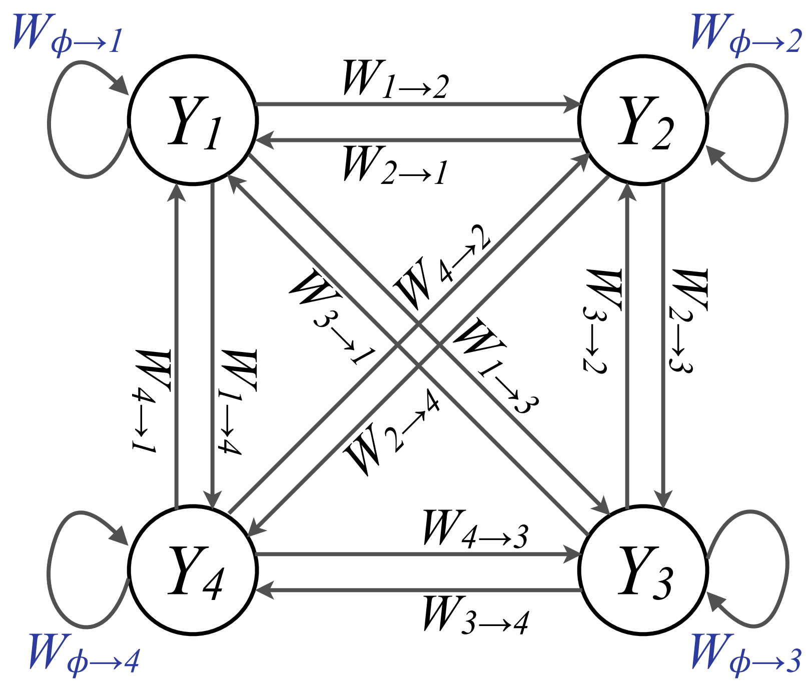
# Structure learning

2. Compute the edge weights using conditional log-likelihood of the data: $W_{j \to i} = \sum_{(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \in D} log P(y_i^{(k)} | \mathbf{x}^{(k)}, y_j^{(k)})$



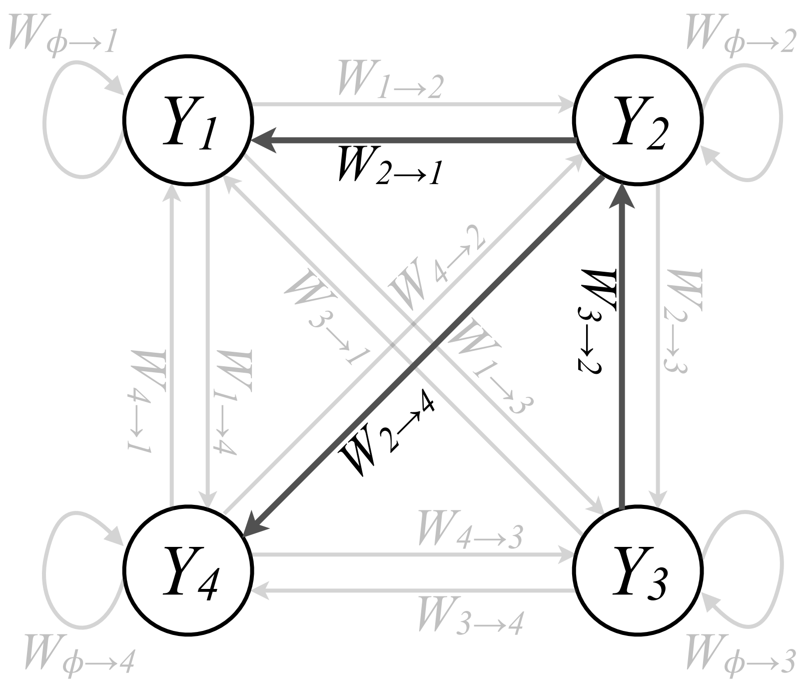
# Structure learning

3. Compute the edge weights using conditional log-likelihood of the data: $W_{\phi \to i} = \sum_{(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \in D} log P(y_i^{(k)} | \mathbf{x}^{(k)})$



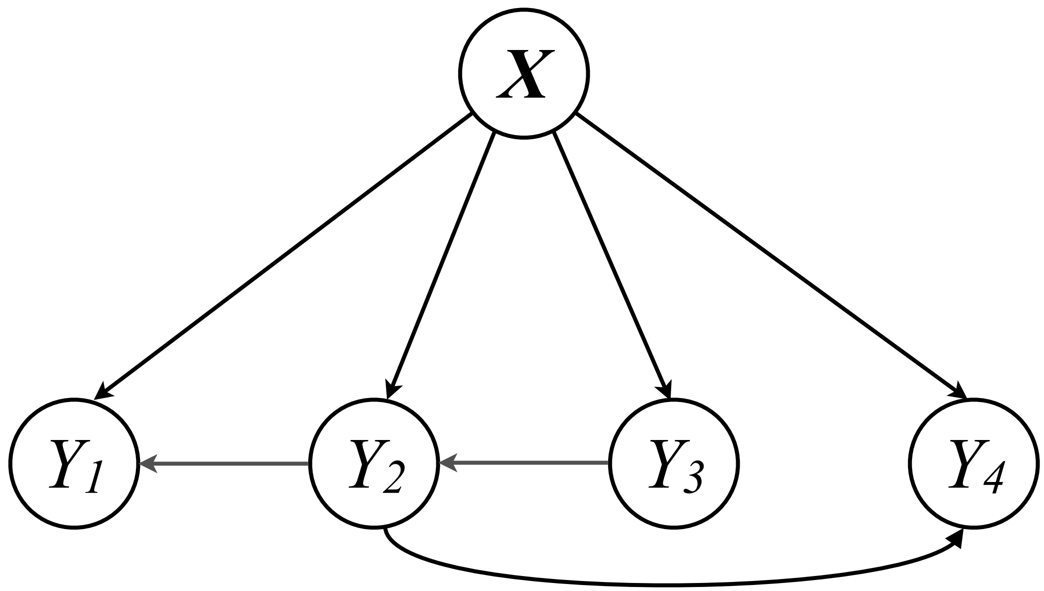
# Structure learning

4. Find the tree that maximizes the sum of the edge weights by solving the maximum branching problem



# Structure learning

5. Add a node for $\boldsymbol{X}$ as the common parent for all classes



# Experiments

- Compared methods
  - *Binary Relevance* (BR) [Boutell et al., ’04, Clare et al., ’01]
  - *Classification with heterogeneous features* (CHF) [Godbole and Sarawagi, ’04]
  - *Multi-label k-nearest neighbor* (MLKNN) [Zhang and Zhou, ’07]
  - *Instance-based learning by logistic regression* (IBLR) [Cheng and Hüllermeier, ’09]
  - *Classifier chains* (CC) [Read et al., ’09]
  - *Maximum margin output coding* (MMOC) [Zhang and Schneider, ’12]

# Experiments

- Data
  - 10 publicly available datasets from different domains

| Dataset | # Instances | # Features | # Classes | Domain |
|---|---|---|---|---|
| Emotions | 593 | 72 | 6 | Music |
| Yeast | 2,417 | 103 | 14 | Biology |
| Scene | 2,407 | 294 | 6 | Image |
| Enron | 1,702 | 1,001 | 53 | Text |
| TMC 2007 | 28,596 | 30,438 | 22 | Text |
| RCV1_subset1 | 6,000 | 8,394 | 10 | Text |
| RCV1_subset2 | 6,000 | 8,304 | 10 | Text |
| RCV1_subset3 | 6,000 | 8,328 | 10 | Text |
| RCV1_subset4 | 6,000 | 8,332 | 10 | Text |
| RCV1_subset5 | 6,000 | 8,367 | 10 | Text |

# Experiment Results

- *Exact Match Accuracy*

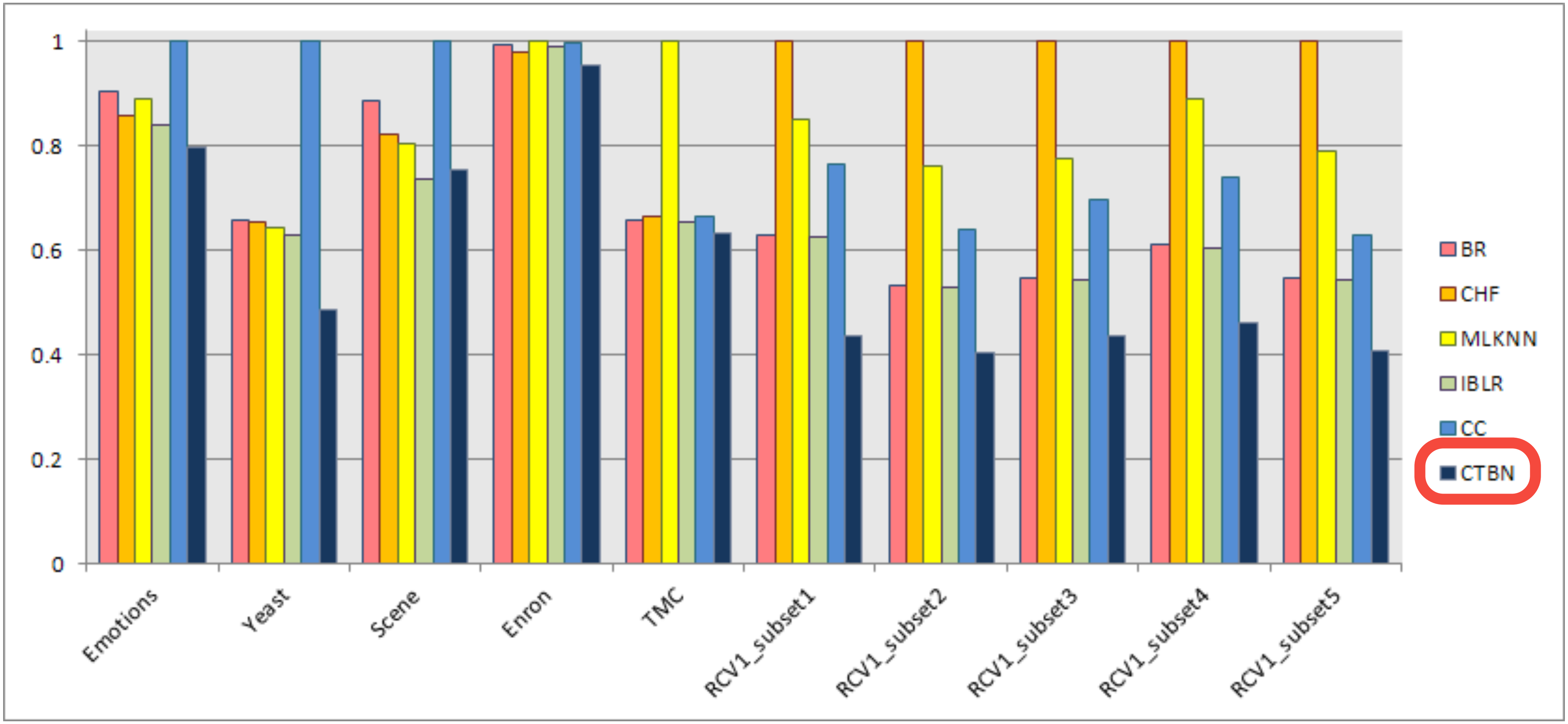
  *The probability of all classes being predicted correctly (higher is better)*

| Dataset | BR | CHF | MLKNN | IBLR | CC | MMOC | **CTBN** |
|---|---|---|---|---|---|---|---|
| Emotions | 0.266 | 0.315 | 0.283 | 0.332 | 0.272 | 0.336 | **0.335** |
| Yeast | 0.147 | 0.162 | 0.179 | 0.204 | 0.194 | 0.214 | **0.195** |
| Scene | 0.521 | 0.160 | 0.629 | 0.644 | 0.633 | 0.684 | **0.626** |
| Enron | 0.162 | 0.169 | 0.078 | 0.163 | 0.173 | | **0.168** |
| TMC | 0.315 | 0.322 | 0.165 | 0.316 | 0.323 | | **0.329** |
| RCV1_subset1 | 0.278 | 0.357 | 0.205 | 0.279 | 0.429 | | **0.448** |
| RCV1_subset2 | 0.42 | 0.466 | 0.288 | 0.417 | 0.517 | | **0.531** |
| RCV1_subset3 | 0.442 | 0.485 | 0.327 | 0.446 | 0.54 | | **0.561** |
| RCV1_subset4 | 0.494 | 0.532 | 0.354 | 0.491 | 0.579 | | **0.59** |
| RCV1_subset5 | 0.412 | 0.457 | 0.276 | 0.411 | 0.497 | | **0.538** |
| **#win-tie-loss** | **9-1-0** | **8-2-0** | **7-3-0** | **9-1-0** | **6-4-0** | **0-1-2** | |

# Experiment Results

- *Normalized conditional log-likelihood loss*

  *Negative log-likelihood normalized on each dataset (lower is better)*



# Conclusion

- We proposed a novel probabilistic approach to multi-dimensional classification
  - CTBN encodes the conditional dependence relations between classes
  - Efficient structure learning and exact inference algorithms are presented
  - Our approach outperforms several state-of-the-art methods