Multi-Label Classification with Cutset Networks

N. Di Mauro, A. Vergari, and F. Esposito

Department of Computer Science, LACAM Laboratory
University of Bari "Aldo Moro", Italy

International Conference on Probabilistic Graphical Models
September 6-9, 2016 - Lugano, Switzerland

PGM 2016 Lugano

UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

Motivation

many real world classification problems involve multiple label classes

- the problem of Multi-Label Classification (MLC) concerns learning a mapping from an example to a set of relevant labels

common approach to MLC is to adopt a problem transformation technique, where a multi-label problem is transformed into one or more single-label problems

- binary relevance[1]
  - decomposes the MLC problem into a set of single label classification problems, one for each different label
- classifier chain[2]
  - transforms a MLC problem into a chain of binary classification problems

[1] **Tsoumaskas2010 Tsoumaskas2010 Tsoumaskas2010**

[2] **Read et al., "Classifier Chains for Multi-label Classification", 2009**

Motivation (II)

Probabilistic Graphical Models provide a powerful formalism to model and reason about MLC problems

- they are able to capture the conditional independence assumption among RVs into a graph based representation
- exploiting MPE inference is one of the approaches to solve MLC[3]

$$\hat{\mathbf{y}} = \underset{\mathbf{y}\in\{0,1\}^L}{\operatorname{argmax}} P(\mathbf{y}|\mathbf{x}) = \underset{\mathbf{y}\in\{0,1\}^L}{\operatorname{argmax}} P(\mathbf{y},\mathbf{x})$$

- learning and inference with PGMs can be challenging

[3] **Corani et al., "Trading off Speed and Accuracy in Multilabel Classification", 2014**

Motivation (III)

the need for exact and efficient inference procedures has lead to the introduction of *Tractable Probabilistic Models* (TPMs)

Cutset Networks (CNets) have been recently proposed as easy-to-learn TPMs[4][5][6]

- **weighted probabilistic model trees** in the form of OR-trees having tree-structured probabilistic models as leaves, and positive weights on inner edges

we show how to employ CNets for MLC problems

- CNets offer exact and tractable MPE inference, thus alleviating a major issue for MLC
- we propose a CNet structure learning algorithmic variant for MLC by focusing on label dependencies

[4] **Rahman et al., "Cutset Networks: A Simple, Tractable, and Scalable Approach for Improving the Accuracy of Chow-Liu Trees", 2014**

[5] **Di Mauro et al., "Learning Accurate Cutset Networks by Exploiting Decomposability", 2015**

[6] **Di Mauro et al., "Learning Bayesian Random Cutset Forests", 2015**

Cutset Networks

Tree-structured model

A *directed tree-structured model* is a Bayesian Network in which each variable has at most one parent

- joint probability distribution over $\mathbf{X}$

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|\mathsf{Pa}_i)$$

- tractability: inference for complete or marginal queries has complexity linear in the number of variables
- learning CLtrees: maximizing the Mutual Information (MI) among random variables in $\mathbf{X}$ leads to the best tree approximating the underlying probability distribution of $\mathcal{D}$ in terms of the Kullback-Leibler divergence
- mixture of CLtrees (MT)
  - $Q(\mathbf{x}) = \sum_{i=1}^{k} \lambda_i \mathcal{T}_i(\mathbf{x})$, $\lambda_i \geq 0$ s.t. $\sum_{i=1}^{k} \lambda_i = 1$
  - weights λ_i learned by EM

Cutset Networks (II)

Cutset Networks (CNets) are a hybrid of rooted OR trees and CLtrees, with OR nodes as internal nodes and CLtrees as leaves[7]

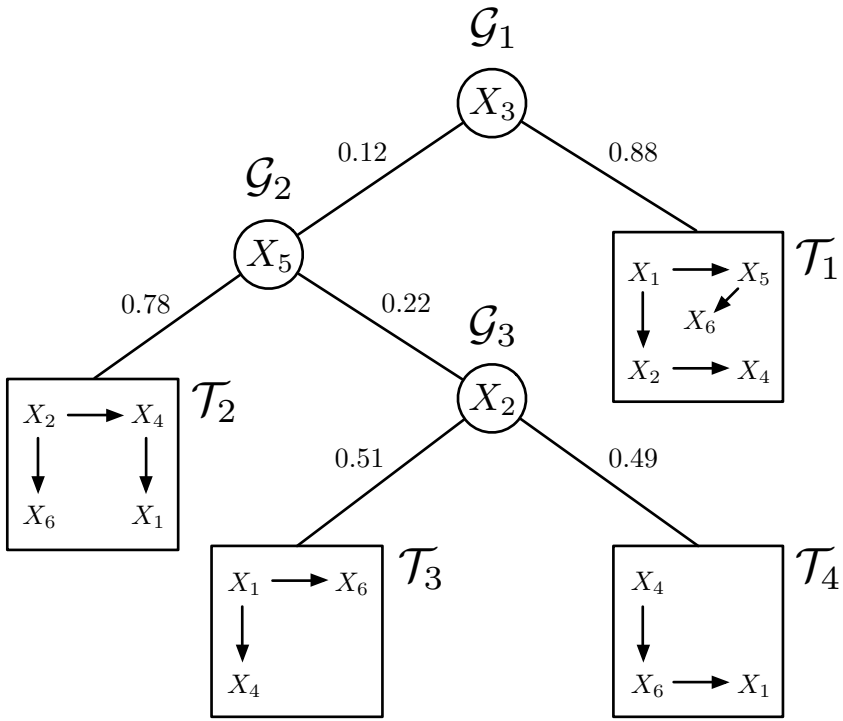
- each OR node is labeled by a variable X_i, and each edge emanating from it represents the conditioning of X_i by a value $x_i^j \in Val(X_i)$, weighted by the probability $w_{i,j}$ of conditioning the variable X_i to the value x_i^j

A cutset network is a pair $\langle \mathcal{G}, \boldsymbol{\gamma} \rangle$

- $\mathcal{G} = \mathcal{O} \cup \{\mathcal{T}_1, \ldots, \mathcal{T}_L\}$ is composed by the rooted OR tree, $\mathcal{O}$, plus the leaf CLtrees $\mathcal{T}_l$
- $\boldsymbol{\gamma} = \boldsymbol{w} \cup \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$ corresponds to the parameters $\boldsymbol{w}$ of the OR tree and $\boldsymbol{\theta}_l$ of the CLTrees
- The *scope* of a CNet $\mathcal{G}$ (resp. a CLtree $\mathcal{T}_l$), denoted as scope($\mathcal{G}$) (resp. scope($\mathcal{T}_l$)), is the set of random variables that appear in it

[7] **Rahman et al., "Cutset Networks: A Simple, Tractable, and Scalable Approach for Improving the Accuracy of Chow-Liu Trees", 2014**

Cutset Networks (III)



distribution represented by a CNet

$$P(x) = \left(\prod_{(v_i,v_j) \in \mathsf{path}_O(x))} w_{ij} \right) \left(T_{l(x)}(x_{V(T_{l(x)})}) \right)$$

CNets Structure Learning

dCSN

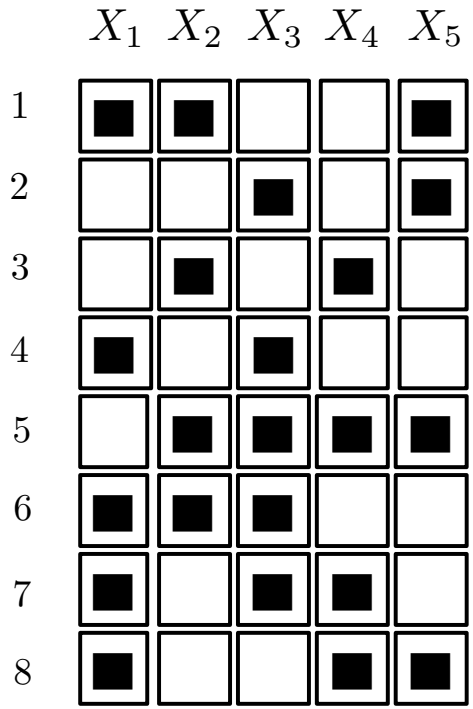
The dCSN algorithm

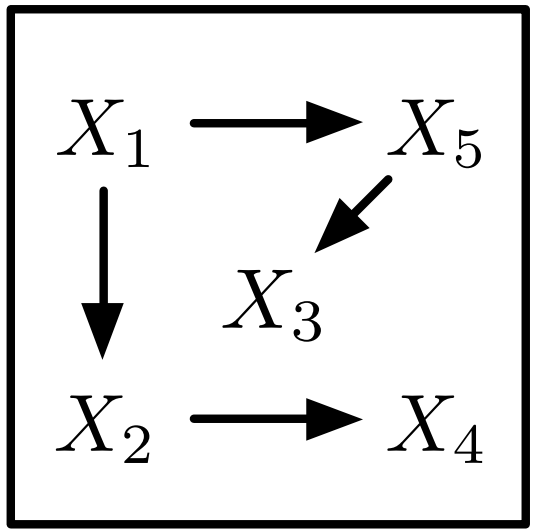
- avoiding decision tree heuristics
- choosing the best variable directly maximizing the log-likelihood
- complex structures penalized adopting the Bayesian Information Criterion BIC

$$\text{score}_{\text{BIC}}(\langle \mathcal{G}, \boldsymbol{\gamma} \rangle) = \log P_{\mathcal{D}}(\langle \mathcal{G}, \boldsymbol{\gamma} \rangle) - \frac{\log M}{2} \text{Dim}(\mathcal{G})$$

1. start with a single CLtree for all variables $\mathbf{X}$
2. check whether there is a decomposition
   - OR node applied on many CLtrees providing a better log-likelihood
3. the decomposition process is recursively applied testing each leaf for a possible substitution

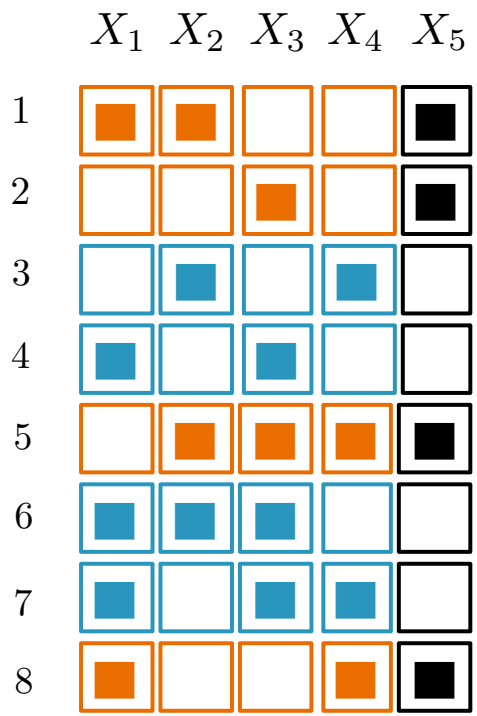
dCSN example I

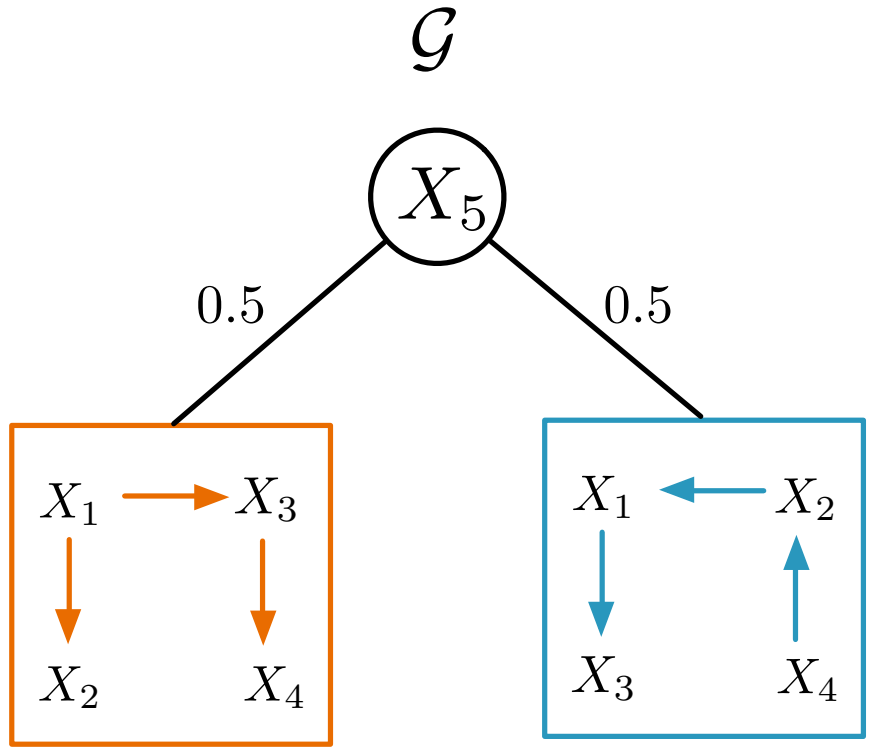




- starting with a single CLTree for all variables X_1, X_2, X_3, X_4, X_5

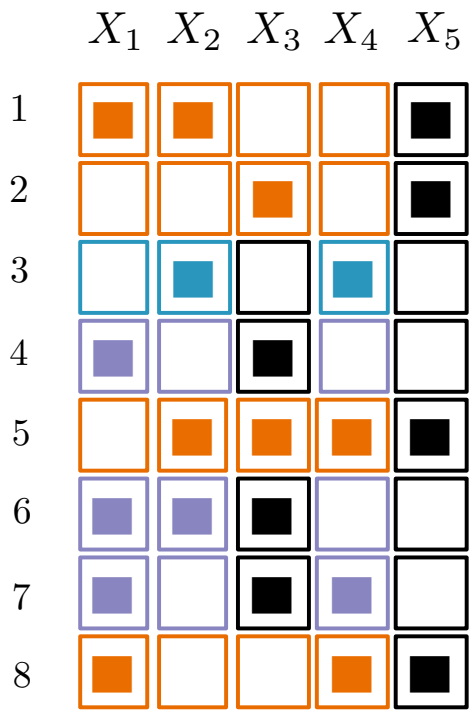
dCSN example II

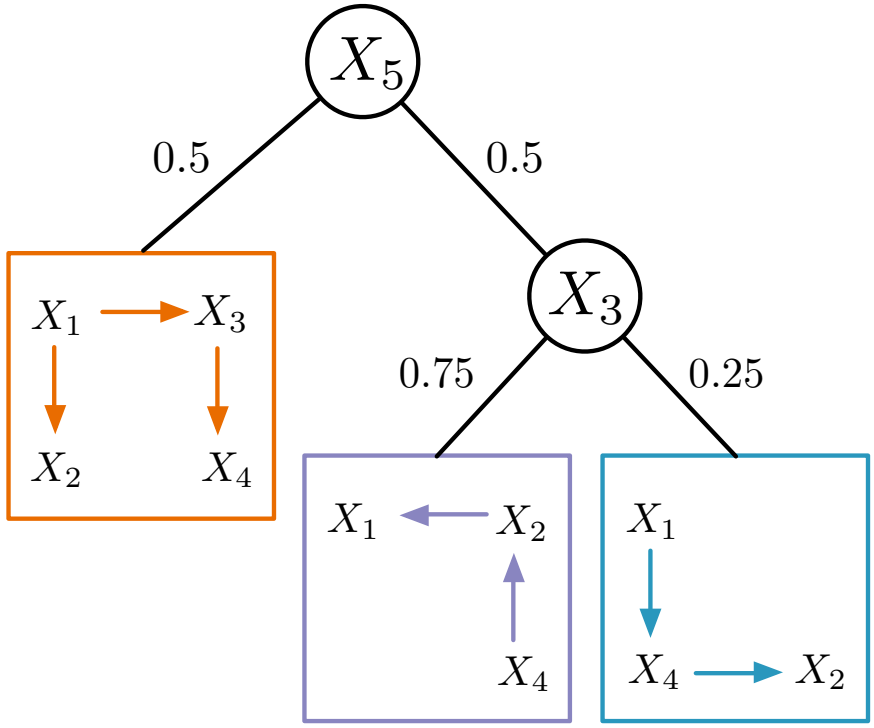




- checking whether there is a decomposition
  - adding OR node on variable X_5 applied on two CLtrees with higher ll

dCSN example III





- recursively apply the decomposition process
  - adding OR node on variable X_3 applied on two CLtrees with higher ll

MPE inference with CNets

An exact MPE assignment can be computed efficiently with CNets:

- linear in the size of the network

Bottom-up evaluation:

- compute MPE assignment $\mathbf{s}$ for a leaf $\mathcal{T}$ w.r.t. its RVs $\mathbf{X}_{\mathcal{T}}$
  - max-out Variable Elimination for CLTrees ($O(|\mathbf{X}_{\mathcal{T}}|)$)
  - keep track of the assignment state $\mathbf{s} = \{X_j = x_j^k | X_j \in \mathbf{X}_{\mathcal{T}}\}$
- for each OR node on RV Z_j and with outgoing edge weights $w_1, \ldots, w_k$ and partial child states $\mathbf{s}_1, \ldots, \mathbf{s}_k$ and probabilities $p_1, \ldots, p_k$
  - if it is an evidence RV, i.e. $Z_j \in \mathbf{X}$: take the corresponding branch state $\mathbf{s}^* \leftarrow \mathbf{s}_j$
  - otherwise choose state $\mathbf{s}^* \leftarrow \mathbf{s}_{j^*}$ such that $j^* = \arg\max_{j=1,\ldots,k} w_j p_j$
  - propagate the newly computed state $\mathbf{s}^*$
- return the root state $\mathbf{s}$

Restricted Cutset Networks (I)

Restricted CNets (RCNets) are a kind of CNets over RVs $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ in which RVs $\mathbf{Y}$ can only appear in the CLTree leaves, i.e. splits in the OR nodes can be taken only RVs $\mathbf{X}$

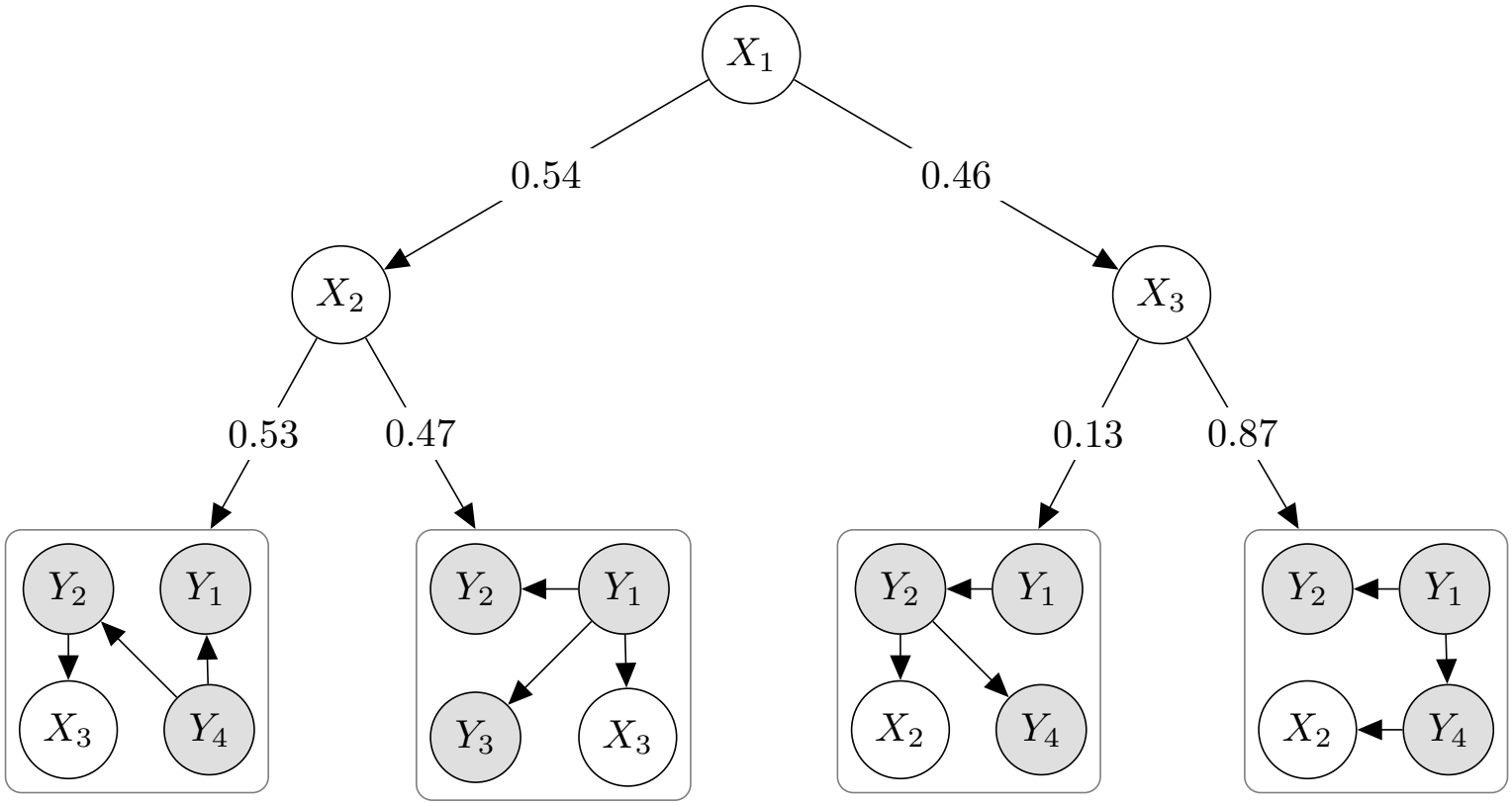
- $X \in \mathbf{X}$ can still appear in the leaves
- the weighted decision tree models the conditioning of the $\mathbf{Y}$ given the $\mathbf{X}$
  - *forcing* the structure to model $\mathbf{X} \rightarrow \mathbf{Y}$ more than $\mathbf{Y} \rightarrow \mathbf{X}$
- still generatively modeling $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$

Additional constrained CLTree learning variant:

- parents RVs can only be $\mathbf{Y}$s
  - feature RVs to be independent given label RVs
  - corrupting the $\mathbf{MI}$ matrix

$$\mathbf{MI}_{\mathbf{X},\mathbf{X}} \leftarrow \mathbf{0}$$
$$\mathbf{MI}_{\mathbf{Y},\mathbf{Y}} \leftarrow \mathbf{MI}_{\mathbf{Y},\mathbf{Y}} + \max \mathbf{MI}$$

Restricted Cutset Networks (II)



Restricted CNets

The Main Algorithm

Algorithm 1 LearnRestrictedCLT($\mathcal{D}$, $\mathbf{X}$, $\mathbf{Y}$)

Input: a set of instances $\mathcal{D}$ over a set of features $\mathbf{X}$ and labels $\mathbf{Y}$
Output: $\langle \mathcal{T}, \boldsymbol{\theta} \rangle$, a tree $\mathcal{T}$ with parameters $\boldsymbol{\theta}$ encoding a pdf over $\mathbf{X} \cup \mathbf{Y}$
$\mathbf{MI} \leftarrow \mathbf{0}_{|\mathbf{X}\cup\mathbf{Y}|\times|\mathbf{X}\cup\mathbf{Y}|}$
for each $V_i, V_j \in \mathbf{X} \cup \mathbf{Y}$ **do**
    $MI_{ij} \leftarrow$ estimateMutualInformation$(V_i, V_j, \mathcal{D})$
$\mathbf{MI}_{\mathbf{X},\mathbf{X}} \leftarrow \mathbf{0}$ ▷ label conditional independence of the $\mathbf{X}$s
$\mathbf{MI}_{\mathbf{Y},\mathbf{Y}} \leftarrow \mathbf{MI}_{\mathbf{Y},\mathbf{Y}} + \max(\mathbf{MI})$ ▷ forcing dependencies among the $\mathbf{Y}$s
$T \leftarrow$ maximumSpanningTree$(\mathbf{MI})$
$\mathcal{T} \leftarrow$ traverseTree(T)
$\boldsymbol{\theta} \leftarrow$ computeFactors$(\mathcal{D}, \mathcal{T})$
return $\langle \mathcal{T}, \boldsymbol{\theta} \rangle$

Experiments (I)

Evaluation Metrics[8]

$$\text{Hamming Score} = \frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}\mathbb{I}(y_j^i = \hat{y}_j^i),$$

$$\text{Exact Match} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\mathbf{y}^i = \hat{\mathbf{y}}^i),$$

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{y}^i \wedge \hat{\mathbf{y}}^i|}{|\mathbf{y}^i \vee \hat{\mathbf{y}}^i|},$$

- Exact Match measure computes the percentage of instances whose predicted set of labels $\hat{\mathbf{y}}$ matches the true set of labels $\mathbf{y}$ *exactly*
- Hamming Score rewards methods for predicting individual labels well
- Accuracy is a label set-based measure defined by the Jaccard similarity coefficients between the predicted and true set of labels

[8] **Dembczyński et al., "On label dependence and loss minimization in multi-label classification", 2012**

Experiments (II)

Datasets description

	Domain	M	N	L	LCard	LDens	LDist
Arts-Yahoo	Text	500	7484	26	1.653	0.063	599
Business-Yahoo	Text	500	11214	30	1.598	0.053	233
CAL500	Music	68	502	174	26.043	0.149	502
Emotions	Music	72	593	6	1.868	0.311	27
Flags	Images	19	194	7	3.391	0.484	54
Health-Yahoo	Text	500	9205	32	1.644	0.051	335
Human	Biology	440	3106	14	1.185	0.084	85
Plant	Biology	440	978	12	1.078	0.089	32
Scene	Images	294	2407	6	1.073	0.178	15
Yeast	Biology	103	2417	14	4.237	0.302	198

- M: number of attributes
- N: instances
- L: labels

Experiments (III)

Algorithms

- Binary Relevance
  - Naive Bayes classifier — `BRNB`
  - Tree Augmented Naive Bayes classifier — `BRTAN`
- Classifier Chain
  - Naive Bayes classifier — `CCNB`
  - Tree Augmented Naive Bayes classifier — `CCTAN`
- Bayesian Chain Classifier[9] — `BCC`
  - builds a probabilistic CC after learning the structure of a BN for MLC (highly competitive against other Bayesian multi-dimensional classifiers)
- Cutset Networks — `CNET`
- Restricted Cutset Networks — `RCNET`

[9] **Zaragoza et al., "Bayesian Chain Classifiers for Multidimensional Classification", 2011**

Results

Accuracy

Dataset	RCNET	CCTAN	BRTAN	CNET	CCNB	BRNB	BCC
Arts	0.432	0.372	0.391	0.399	0.376	0.358	0.368
Business	0.728	0.703	0.704	0.719	0.68	0.660	0.685
CAL500	0.203	0.188	0.251	0.187	0.184	0.252	0.202
Emotions	0.554	0.563	0.542	0.499	0.553	0.512	0.547
Flags	0.563	0.565	0.544	0.526	0.542	0.545	0.53
Health	0.603	0.611	0.610	0.587	0.578	0.582	0.578
Human	0.361	0.301	0.249	0.306	0.258	0.197	0.248
Plant	0.397	0.313	0.308	0.354	0.298	0.278	0.292
Scene	0.669	0.658	0.634	0.530	0.530	0.528	0.523
Yeast	0.464	0.452	0.479	0.442	0.404	0.429	0.425
Avrg rank	**1.7**	**2.8**	**3.2**	**4.0**	**5.1**	**5.4**	**5.6**

Results (II)

Hamming score

Dataset	RCNET	BRTAN	CCTAN	CNET	BRNB	BCC	CCNB
Arts	0.937	0.931	0.940	0.937	0.926	0.925	0.923
Business	0.973	0.965	0.965	0.974	0.957	0.948	0.942
CAL500	0.854	0.814	0.823	0.854	0.813	0.829	0.684
Emotions	0.783	0.797	0.786	0.737	0.787	0.760	0.768
Flags	0.709	0.710	0.709	0.680	0.723	0.714	0.708
Health	0.965	0.963	0.964	0.964	0.959	0.957	0.955
Human	0.894	0.890	0.870	0.894	0.884	0.820	0.827
Plant	0.892	0.894	0.901	0.888	0.890	0.884	0.884
Scene	0.879	0.898	0.888	0.838	0.874	0.813	0.816
Yeast	0.773	0.765	0.749	0.765	0.732	0.730	0.720
Avrg rank	**2.2**	**2.8**	**2.9**	**3.3**	**4.1**	**5.5**	**6.4**

Results (III)

Exact Match

Dataset	RCNET	CCTAN	CNET	BRTAN	CCNB	BRNB	BCC
Arts	0.313	0.275	0.302	0.247	0.251	0.234	0.187
Business	0.567	0.548	0.565	0.545	0.520	0.518	0.410
CAL500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Emotions	0.295	0.282	0.260	0.295	0.265	0.280	0.207
Flags	0.170	0.212	0.185	0.108	0.129	0.114	0.109
Health	0.456	0.478	0.448	0.437	0.407	0.425	0.247
Human	0.263	0.139	0.260	0.141	0.085	0.100	0.150
Plant	0.344	0.247	0.333	0.207	0.185	0.181	0.110
Scene	0.554	0.498	0.492	0.518	0.248	0.375	0.422
Yeast	0.147	0.169	0.129	0.120	0.117	0.101	0.074
Avrg rank	**1.4**	**2.3**	**2.7**	**3.5**	**4.9**	**5.2**	**5.7**

Conclusions

- tackled the MLC problem by employing CNets
  - CNets to efficiently and exactly solve an MPE formulation of MLC
  - a structure learning algorithmic variant for CNets to cope with the prominence of label dependencies in MLC
- the proposed model is able to represent the dependencies among multiple label variables
- an exact inference procedure for label predictions
- experimental evaluation on 10 real-world datasets
  - the approach can effectively improve the accuracy, exact match and hamming scores
  - highly competitive against more complex ensemble approaches

Code available at `http://www.di.uniba.it/~ndm/dcsn/`

Thank you!