Classifier Chains for Multi-label Classification

Jesse Read, Bernhard Pfahringer, Geoff Holmes, Eibe Frank

University of Waikato, Hamilton, New Zealand

Introduction

- Multi-label Data
  - Each instance is associated with *multiple* labels
  - Given instances $x_1, x_2, \cdots, x_n$ and a *predefined* set of labels L:
  - single-label data: $(x_1, l_1), (x_2, l_2), \cdots, (x_n, l_n)$ where each $l_i \in L$
  - multi-label data: $(x_1, S_1), (x_2, S_2), \cdots, (x_n, S_n)$ where each $S_i \subseteq L$
  - For example, a film can be labeled {`romance`,`comedy`}
- Applications
  - Scene, Video classification
  - Text classification
  - Medical classification
  - Biology, Genomics
- Multi-label Issues
  - label correlations: consider {`romance`,`comedy`} vs {`romance`,`horror`}
  - computational complexity

Prior Work

- Binary relevance method (BR): binary problem for each label
  - simple, intuitive
  - efficient: can be run in parallel or serial
  - useful for incremental contexts
  - *but doesn't account for label correlations*
  - *e.g.* Nearest neighbor approaches based on BR, e.g. `MLkNN`
  - *e.g.* Stacking approaches, e.g. meta level stacking (MS)
  - *e.g.* Pairwise approaches, e.g. calibrated label ranking
- Label powerset method: label sets are treated as single labels
  - takes into account label correlations
  - *but can become computationally complex*
  - *e.g.* RAKEL: ensemble of subsets
  - *e.g.* EPS: ensemble of pruned sets
- Other methods
  - often model label correlations in a complex way, prone to overfitting
- Classifier Chains (CC)
  - To account for label correlations while retaining advantages of BR: able to scale up to larger problems with e.g. SVMs as the base classifier.

Classifier Chains (CC)

- Binary Relevance (BR)
  - $|L|$ classifiers $C_1 \cdots C_{|L|}$ predict the relevance of each $l_i \in L$
  - each $C_i : x \to Y[i] \in \{0, 1\}$ where $Y[i] = 1$ if $l_i \in Y, Y \subseteq L$
- Classifier Chains (CC)
  - $|L|$ classifiers $C_1 \cdots C_{|L|}$ predict the relevance of each $l_i \in L$
  - each $C_i : (x \cup Y[1] \cup \cdots \cup Y[i-1]) \to Y[i] \in \{0, 1\}$
  - i.e. extending the feature space x with *the binary relevances of all previous labels in the chain*

E.G. For $L = \{$`romance`,`horror`,`comedy`,`drama`,`action`,`western`$\}$ ($|L| = 6$):

Classifiers	Classifications
$C_1 : x \to \{$`romance`,`!romance`$\}$	`romance`
$C_2 : x \cup$ `romance` $\to \{$`horror`,`!horror`$\}$	`!horror`
$C_3 : x \cup$ `romance` $\cup$ `!horror` $\to \{$`comedy`,`!comedy`$\}$	`comedy`
$C_4 : x \cup$ `romance` $\cup$ `!horror` $\cup$ `comedy` $\to \{$`drama`,`!drama`$\}$	`!drama`
$C_5 : x \cup$ `romance` $\cup$ `!horror` $\cup$ `comedy` $\cup$ `!drama` $\to \cdots$	`!action`
$C_6 : x \cup$ `romance` $\cup$ `!horror` $\cup$ `comedy` $\cup$ `!drama` $\cup \cdots \to \cdots$	`!western`
	$Y \subseteq L = \{$`romance`,`comedy`$\}$

- similar advantages to binary relevance method
- time complexity similar in practice
- takes into account label correlations
- *one chain can't be run in parallel* (but can be run in serial)
- *how to order the chain?*

Ensembles of Classifier Chains (ECC)

- Ensembles
  - known for augmenting accuracy
  - more label correlations can be learnt, without overfitting
  - solves 'chain order' issue: each chain random order
  - generic vote/score/threshold classification method
  - can also be applied to binary relevance method, i.e. EBR

Experiments

- Evaluation:
  - Label set evaluation: subset Accuracy, Macro F-measure
  - Per-label evaluation: LogLoss, $AU(\overline{PRC})$
  - 5x2 cross validation; train/test on large datasets

Datasets

	$\lvert D\rvert$	$\lvert L\rvert$	$\lvert X\rvert$	$LC(D)$	$PD(D)$
Scene	2407	6	294n	1.07	0.006
Yeast	2417	14	103n	4.24	0.082
Medical	978	45	1449	1.25	0.096
Slashdot	3782	22	1079	1.18	0.041
Enron	1702	53	1001	3.38	0.442
Reuters	6000	103	500n	1.46	0.147
Ohsumed	13929	23	1002	1.66	0.082
Tmc2007	28596	22	500	2.16	0.047
MediaMill	43907	101	120n	4.38	0.149
Bibtex	7395	159	1836	2.40	0.386
IMDB	95424	28	1001	1.92	0.036
Delicious	16105	983	500	19.02	0.981

Algorithms

BR-based algorithms	
`BR`	Binary Relevance
`CC`	Classifier Chain
`SM`	Subset Mapping
`MS`	Meta Stacking
`MLkNN`	Meta Stacking
Ensemble algorithms	
`EBM`	Ensembles of BR
`ECC`	Ensembles of CC
`EPS`	Ensembles of PS
`RAKEL`	RAndom K labEL subsets

(SVMs as base classifiers)

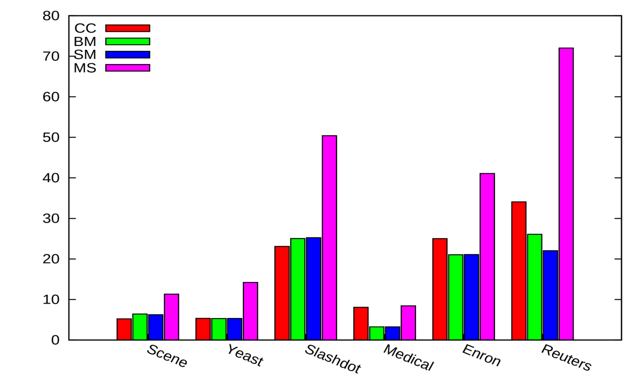
Results (Summary)

- Comparing CC to BR and BR-related methods.

Table: Standard Datasets.

	CC	BR	SM	MS
Accuracy	**5**	0	1	0
Macro F1	**5**	0	1	0
Micro F1	**3**	1	0	2
Exact M.	**6**	0	0	0
Total wins	**19**	1	2	2

Figure: Build times (sec).



- CC justified over default BR, other similar BR-based methods
- CC's complexity usually comparable to BR in practice, except for special cases (e.g. *Medical* which has a relatively large label set L)

- Comparing ECC to EBR and RAKEL, EPS, `MLkNN`

Table: Standard Datasets.

	ECC	EBR	EPS	MLkNN	RAKEL
Accuracy	2	0	**3**	0	1
Macro F1	1	0	**4**	0	1
Log Loss	**3**	0	1	1	1
$AU(\overline{PRC})$	**3**	0	0	**3**	0
Total wins	**9**	0	8	4	3

Table: Large Datasets.

	ECC	EBR	EPS†	MLkNN	RAKEL†
Accuracy	**4**	0	1	1	0
Macro F1	**3**	0	1	1	1
Log Loss	1	1	0	**4**	0
$AU(\overline{PRC})$	**4**	0	0	2	0
Total wins	**12**	1	2	8	1

† 2 DNF for RAKEL; 1 for EPS

- Binary methods (e.g. ECC, `MLkNN`) are better at *per-label* evaluation
- whereas other methods are better at *label-set* evaluation
- Binary methods are better on large datasets, even at *label-set* evaluation
- indicating that directly modelling label correlations (e.g. EPS, RAKEL) is less helpful with larger numbers of training instances
- ECC is the best performer overall

Table: Fastest method for build, test times (excl. EBR, `MLkNN`)

Dataset	Build	Test	Dataset	Build	Test
Scene	EPS	RAK	OHSUMED	ECC	ECC
Yeast	ECC	ECC	TMC2007	EPS	ECC
Slashdot	RAK	RAK	Bibtex	ECC	ECC
Medical	RAK	RAK	MediaMill	ECC	ECC
Enron	EPS	ECC	IMDB	RAK	ECC
Reuters	ECC	ECC	Delicious	EPS	EPS

2 DNF for RAKEL; 1 for EPS

- ECC's efficiency is most noticeable on the larger datasets

Conclusion

- Ensembles of Classifier Chains
  - classifier chains improve on the binary relevance method
  - takes into account label correlations without overfitting
  - efficient, can be run in parallel and serial
  - performs well, especially on large data sets