Aggregating Independent and Dependent Models to Learn Multi-label Classifiers

E. Montañés, J.R. Quevedo, J.J. del Coz

Artificial Intelligence Center – University of Oviedo

Outline

Outline

The goal

- To study **dependence and independence on their own**
- To show that **both help** to improve the multi-label classification
- To propose **aggregating vs. stacking**
- To compare **actual labels vs. predicted ones**
- To compare **binary vs. probabilistic outputs**

Outline

Multi-label vs. mono-label classification

Mono-label classification

Binary classification

X_1	X_2	X_3	X_4	C
3	T	1.5	-2	1
2	F	1.5	-4	0
9	T	2.3	-2	1
4	F	7.8	-1	0
3	F	1.5	-9	1

Multi-class classification

X_1	X_2	X_3	X_4	C
3	T	1.5	-2	1
2	F	1.5	-4	2
9	T	2.3	-2	2
4	F	7.8	-1	3
3	F	1.5	-9	1

Multi-label classification

X_1	X_2	X_3	X_4	C_1	C_2	C_3	C_4	C_5
3	T	1.5	-2	1	0	0	1	0
2	F	1.5	-4	0	0	0	0	1
9	T	2.3	-2	0	1	1	1	1
4	F	7.8	-1	1	0	0	1	0
3	F	1.5	-9	1	1	0	1	0

Many applications

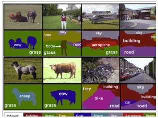
Text classification

Social networks

Medical diagnosis

Weather forecast

Object detection



Face recognition

Formal statement

Point of departure

$\mathcal{L} = \{\ell_1, \ell_2, \dots, \ell_m\}$ a set of labels

$\mathcal{X}$ an input space

$\mathcal{Y} = \mathcal{P}(\mathcal{L}) \sim \{0,1\}^m$ (the power set of $\mathcal{L}$)

$$\Downarrow$$

$$S = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \dots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\} \in \mathcal{X} \times \mathcal{Y}$$

$$\approx$$

$$\mathbf{P}(\mathbf{X}, \mathbf{Y})$$

The target

Induce $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ from S

$$\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \dots h_m(\boldsymbol{x}))$$

$h_j : \mathcal{X} \longrightarrow \{0,1\}$ predicts if ℓ_j is attached to $\boldsymbol{x}$

Evaluating multi-label classification

Example-based measures

- **classification** rather than ranking
- capture correlations among labels at **example level**
- used in **stacking approaches**

Biased measures

- Jaccard index
- Precision, Recall, F_1

Other measures

- Hamming loss
- 0/1 loss

Outline

Binary relevance

- Each label is learned **independently** of the rest
- **Linear complexity** respect to the number of labels
- Does **not** consider **label dependence**

$$\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots h_m(\boldsymbol{x}))$$
$$h_j : \mathcal{X} \longrightarrow \{0,1\}$$

Stacking approaches

Two groups of classifiers are learned

- The **independent** ones

$$\boldsymbol{h}^1(\boldsymbol{x}) = (h^1_1(\boldsymbol{x}), \ldots, h^1_m(\boldsymbol{x}))$$
$$h^1_j : \mathcal{X} \longrightarrow \{0, 1\}$$

- The **dependent** ones

$$\boldsymbol{h}^2(\boldsymbol{x}, \boldsymbol{h}^1(\boldsymbol{x})) = (h^2_1(\boldsymbol{x}, \boldsymbol{h}^1(\boldsymbol{x})), \ldots, h^2_m(\boldsymbol{x}, \boldsymbol{h}^1(\boldsymbol{x})))$$
$$h^2_j : \mathcal{X} \times \{0, 1\}^m \longrightarrow \{0, 1\}$$

$$\Downarrow$$

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}^2(\boldsymbol{x}, \boldsymbol{h}^1(\boldsymbol{x}))$$

Classifier chains

- **One classifier per label**
- **Same complexity** as binary relevance
- A **chain of classifiers** is built according to certain order of the labels

$$\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}, h_1(\boldsymbol{x})), h_3(\boldsymbol{x}, h_1(\boldsymbol{x}), h_2(\boldsymbol{x}, h_1(\boldsymbol{x}))), \ldots)$$
$$h_j : \mathcal{X} \times \{0,1\}^{j-1} \longrightarrow \{0,1\}$$

Ensemble version

- Several orders of labels are ensembled
- Diminishes the effect of the label order

Probabilistic version

- The probability product rule is used
- The complexity increases considerably in testing

Other approaches

MLkNN

Instance-based learner

- Posterior (over neighbors) and prior probability based on frequency counting
- Bayes rule gives the labels' probability

Instance-Based Learning by Logistic Regression (IBLR)

Unifies instance-based learning and logistic regression

RAndom k-labELsets (RAkEL)

Ensemble of Label Power set (LP) classifiers

Other approaches

MLkNN

Instance-based learner

Instance-Based Learning by Logistic Regression (IBLR)

Unifies instance-based learning and logistic regression

- Labels of the neighbors as additional features
- Classification by logistic regression

RAndom k-labELsets (RAkEL)

Ensemble of Label Power set (LP) classifiers

Other approaches

MLkNN

Instance-based learner

Instance-Based Learning by Logistic Regression (IBLR)

Unifies instance-based learning and logistic regression

RAndom k-labELsets (RAkEL)

Ensemble of Label Power set (LP) classifiers

- It randomly selects a k-labelset $\boldsymbol{Y}_i$ from $\mathcal{L}$ without replacement
- It learns a LP classifier of the form $\mathcal{X} \rightarrow \mathcal{P}(\boldsymbol{Y}_i)$
- A voting process determines the final classification

Outline

Aggregating Independent and Dependent classifiers (AID)

Our hypothesis

Both approaches are **not exclusive**, but **complementary**

- The **independent** ones

$$\boldsymbol{h}^1(\boldsymbol{x}) = (h_1^1(\boldsymbol{x}), \dots, h_m^1(\boldsymbol{x}))$$
$$h_j^1 : \mathcal{X} \longrightarrow \{0,1\}$$

- The **dependent** ones

$$\boldsymbol{h}^2(\boldsymbol{x},\boldsymbol{y}) = (h_1^2(\boldsymbol{x}, y_2, \dots, y_m), \dots, h_m^2(\boldsymbol{x}, y_1, \dots, y_{m-1}))$$
$$h_j^2 : \mathcal{X} \times \{0,1\}^{m-1} \longrightarrow \{0,1\}$$

$$\Downarrow$$

$$\boldsymbol{h}(\boldsymbol{x}) = \oplus(\ (h_1^1(\boldsymbol{x}), \dots, h_m^1(\boldsymbol{x})),$$
$$(h_1^2(\boldsymbol{x}, h_2^1(\boldsymbol{x}), \dots, h_m^1(\boldsymbol{x})), \dots, h_m^2(\boldsymbol{x}, h_1^1(\boldsymbol{x}), \dots, h_{m-1}^1(\boldsymbol{x})))\ )$$

Comparing AID with other methods (I)

With regard to ...

... binary relevance

- It **also considers correlations** among labels

... stacking approaches

- The outputs of independent classifiers are **additionally employed** to decide the predicted labels
- **Actual labels** rather than predicted labels (more reliable information)

Comparing AID with other methods (II)

With regard to ...

... chain classifiers

- **Free of in-chain dependence**
- **Richer estimations** since all correlations are considered
- Although it **only** offers **greedy approximations** of the entire join distribution

... MLkNN, IBLR & RAkEL

- **Interpretability** of different kinds of labels predicted
  - Those coming just from the description of the examples
  - Those coming from other labels

About the complexity

	BR	STA	CC	PCC	ECC	EPCC	AID
Models	m	$2m$	m	m	Nm	Nm	$2m$
Testing complexity	Linear	Linear	Linear	Exp.	Linear	Exp.	Linear

Outline

Settings

The learning process

- For binary relevance, CC, stacking & AID
  - **logistic regression** as binary base learner
  - A **grid search for C** over $\{10^p \mid p \in [-3, \ldots, 3]\}$ optimizing the accuracy through a balanced 2-fold cross validation repeated 5 times
- **Default parameters** for MLkNN, IBLR & RAkEL

The evaluation

Using a 10-fold cross validation, we estimate

- Jaccard index
- Precision, Recall, & F_1
- Hamming loss & 0/1 loss

AID vs. Stacking

Average ranks over all data sets

	BR	AID	STA^{y}	AID$^{y'}$	STA	AID^{p}	STA^{p}
Precision	4.45	**2.45**	3.59	4.91	4.82	3.09	4.68
Recall	6.18	**1.45**	3.00	3.55	4.27	3.55	6.00
F1	5.82	**1.36**	2.95	4.45	4.91	3.18	5.32
Jaccard	5.64	**1.45**	2.95	4.64	4.91	3.18	5.23
Hamming	**2.27**	4.91	5.00	4.59	4.41	2.77	4.05
0/1loss	4.91	**2.64**	2.73	5.05	4.77	3.27	4.64

1. **STA** or **AID**?
2. **Actual label** data or **predictions** of independent models?
3. In the testing phase, **binary** or **probabilistic** features?
4. To **aggregate** or to **stack**?

AID vs. other methods

Average ranks over all data sets

	BR	MLkNN	IBLR	RAkEL	ECC	AID	AID^p
Precision	4.05	5.55	4.55	3.45	4.14	**3.05**	3.23
Recall	5.14	6.45	4.55	2.73	4.86	**1.41**	2.86
F1	4.95	6.36	4.55	2.82	4.86	**1.59**	2.86
Jaccard	4.77	6.27	4.00	2.91	4.86	**2.05**	3.14
Hamming	3.05	4.36	4.18	4.36	**3.00**	5.18	3.86
0/1loss	4.50	5.14	3.73	3.55	4.23	**2.91**	3.95

- **AID is the best** for all measures except for Hamming loss
- In **Hamming loss, ECC is the best** and AID is the worst
- **AID^p** is quite **steady** for all measures

Outline

Conclusions

- **Interpretability** of two kinds of labels
- **Actual labels** better than **predicted ones**
- **Aggregating** better than **stacking**
- AID exhibits **competitive results**, but not for Hamming loss
- AID has **linear complexity** in both training and testing stages