Deep Learning with a Rethinking Structure for Multi-label Classification

Yao-Yuan Yang and Yi-An Lin and
Hong-Min Chu and Hsuan-Tien Lin

Department of Computer Science & Information Engineering
National Taiwan University

November 19, 2019

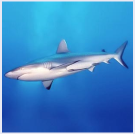
Which animal?

dog

rabbit

cat

guinea pig



shark

Which animals?

dog

1

rabbit

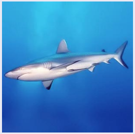
1

cat

1

guinea pig

1



shark

0

Multi-label Classification (MLC)

Problem definition

- train set $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$
- label vector $\mathbf{y} \in \{0, 1\}^K$
  - $\mathbf{y}[k] = 1$ if and only if the k-th label is relevant to $\mathbf{x}$
- learn a f that maps $\mathbf{x}$ to $\mathbf{y}$
- test data $(\mathbf{x}, \mathbf{y})$, prediction $\hat{\mathbf{y}} = f(\mathbf{x})$
- goal is to make $\hat{\mathbf{y}}$ close to ground truth $\mathbf{y}$

Evaluation

- cost function $C(\mathbf{y}, \hat{\mathbf{y}}) \in \mathcal{R}$: the cost of predicting $\mathbf{y}$ as $\hat{\mathbf{y}}$
- F1 score, Accuracy score, Hamming loss, Rank loss

Baseline 1: Binary Relevance (BR)

Binary Relevance (BR)

train a dog, rabbit, cat, guinea pig, shark classifier independently

$f_1(\mathbf{x})$

dog

$f_2(\mathbf{x})$

rabbit

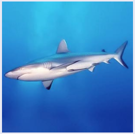
$f_3(\mathbf{x})$

cat

$f_4(\mathbf{x})$

guinea pig

$f_5(\mathbf{x})$



shark

Label correlation

- If we predict guinea pig as 1, we have the information about the shark is not likely predicted as 1
- BR ignores the label correlation information

Baseline 2: Classifier Chain (CC)

Classifier Chain (CC)

- Learn each classifiers dependently (sequentially)

	BR	CC
dog	$f_1(\mathbf{x})$	$\hat{y}[1] = f_1(\mathbf{x})$
rabbit	$f_2(\mathbf{x})$	$\hat{y}[2] = f_2([\mathbf{x}; \hat{y}[1]])$
cat	$f_3(\mathbf{x})$	$\hat{y}[3] = f_3([\mathbf{x}; \hat{y}[1]; \hat{y}[2]])$
guinea pig	$f_4(\mathbf{x})$	$\hat{y}[4] = f_4([\mathbf{x}; \hat{y}[1]; \hat{y}[2]; \hat{y}[3]])$
shark	$f_5(\mathbf{x})$	$\hat{y}[5] = f_5([\mathbf{x}; \hat{y}[1]; \hat{y}[2]; \hat{y}[3]; \hat{y}[4]])$

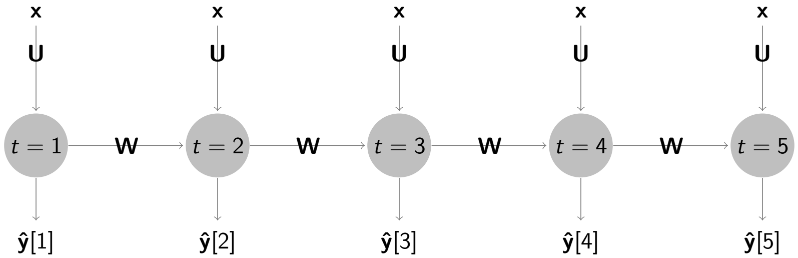
Extension with neural network (for feature extraction)

- Ability to attach other deep learning architecture for specific data feature extraction [WYM+16, CCYW17]
- CC can be seen as remembering previous prediction - recurrent neural network - to model this prediction sequence

Baseline 3: Recurrent Neural Network (RNN)

CC liked Recurrent Neural Network (RNN) [WYM+16]

- Solves sequence prediction problem
- Feature transformation **U**, Memory transformation **W**



- Problem with predicting label sequentially: label ordering
  - Searching for optimal ordering [LT15]
- Imbalance information received for each label in the sequence

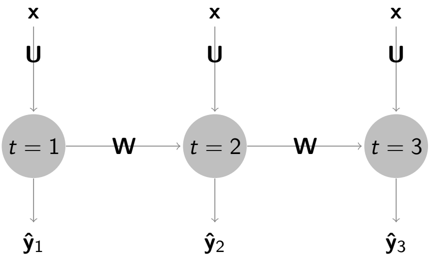
Instead of treating the prediction of labels as a sequence, treat the prediction of **label vectors** as a sequence

List of labels:

- person
- dog
- cat
- spoon
- cup
- ...

RethinkNet

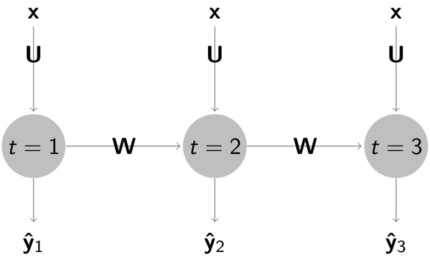
- Modify from Baseline 3
- Output $\hat{\mathbf{y}}_i \in \{0,1\}^K$ instead of $\hat{\mathbf{y}}[i] \in \{0,1\}$



1. person, cup, fork, bowl, chair, table
2. person, bottle, cup, fork, knife, spoon, bowl, chair, table
3. person, cup, fork, knife, spoon, bowl, chair, table

RethinkNet

- Modify from Baseline 3
- Output $\hat{\mathbf{y}}_i \in \{0,1\}^K$ instead of $\hat{\mathbf{y}}[i] \in \{0,1\}$

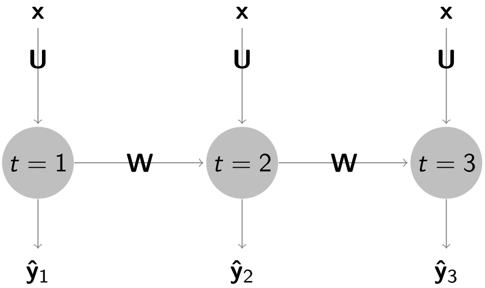


1. person, cup, fork, bowl, chair, table
2. person, bottle, cup, fork, knife, spoon, bowl, chair, table
3. person, cup, fork, knife, spoon, bowl, chair, table

RethinkNet (modeling label correlation)

Considering a linear case

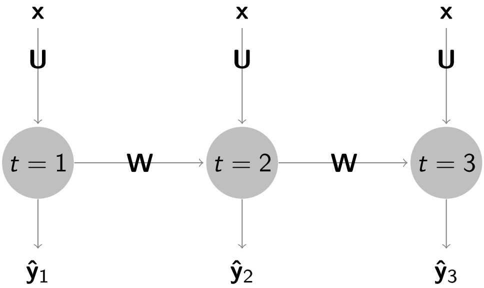
- $\mathbf{U} \in R^{K \times d}$ and $\mathbf{W} \in R^{K \times K}$
- $\hat{\mathbf{y}}_t = \sigma(\mathbf{Ux} + \mathbf{W}\hat{\mathbf{y}}_{t-1})$ (feature: $\mathbf{Ux}$, memory: $\mathbf{W}\hat{\mathbf{y}}_{t-1}$)
- $\hat{\mathbf{y}}_t[j] = \sigma(\mathbf{Ux} + \sum_{i=1}^{K} \hat{\mathbf{y}}_{t-1}[i] * \mathbf{W}[i,j])$
- $\mathbf{W}[i,j]$ matches the correlation between i-th and j-th label



RethinkNet (modeling label correlation)

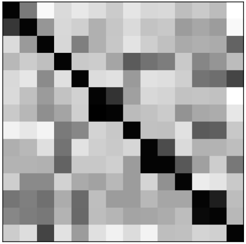
Considering a linear case

- $\mathbf{U} \in R^{K \times d}$ and $\mathbf{W} \in R^{K \times K}$
- $\hat{\mathbf{y}}_t = \sigma(\mathbf{U}\mathbf{x} + \mathbf{W}\hat{\mathbf{y}}_{t-1})$ (feature: $\mathbf{U}\mathbf{x}$, memory: $\mathbf{W}\hat{\mathbf{y}}_{t-1}$)
- $\hat{\mathbf{y}}_t[j] = \sigma(\mathbf{U}\mathbf{x}[j] + \sum_{i=1}^{K} \hat{\mathbf{y}}_{t-1}[i] * \mathbf{W}[i,j])$
- $\mathbf{W}[i,j]$ matches the correlation between i-th and j-th label

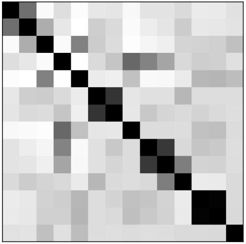


RethinkNet (modeling label correlation)

To verify we plot the memory transform **W** and the correlation coefficient of the yeast data set.



memory transform

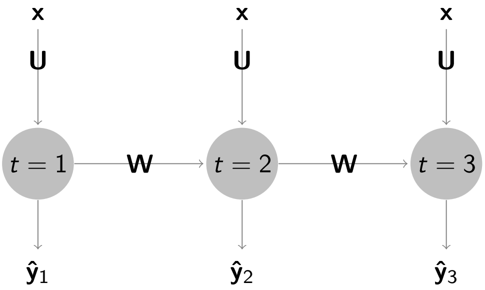


correlation coefficient

RethinkNet (Fine-tune)

Fine-tuning towards different evaluation criteria

- Different measures for the closeness between y and $\hat{y}$
- Improve through iteration on specific criterion
- Adjust the weights to the labels by comparing temporary predictions between time steps



Experiments

Experiment (11 MLC data sets)

- RethinkNet versus the other algorithms based on t-test at 95% confidence level (#win/#tie/#loss)

	PCC	CFT	CC-DP	CC	CC-RNN	BR
hamming (↓)	6/1/4	3/4/4	5/2/1	6/1/4	8/3/0	3/6/2
rank loss (↓)	5/1/5	5/2/4	7/1/0	10/1/0	10/1/0	10/1/0
f1 (↑)	6/2/3	5/4/2	5/2/1	8/3/0	10/1/0	9/2/0
acc (↑)	7/1/3	5/4/2	5/1/2	7/4/0	9/2/0	9/2/0
total	**24**/5/15	**18**/14/12	**22**/6/4	**31**/9/4	**37**/7/0	**31**/11/2

Conclusion

- Taken a memory view and developed a novel MLC algorithm – RethinkNet
  - Instead of treating labels as a sequence, treat label vectors as a sequence
  - Models the of label correlation without label ordering problem
  - Fine-tune towards different evaluation criteria
- Empirical study demonstrates that RethinkNet is able to outperform existing algorithms

Thank you for listening. Any question?

References

Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang, *Order-free RNN with visual attention for multi-label classification*, arXiv preprint arXiv:1707.05495 (2017).

Weiwei Liu and Ivor Tsang, *On the optimality of classifier chain for multi-label classification*, NIPS, 2015.

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu, *Cnn-rnn: A unified framework for multi-label image classification*, CVPR, 2016.