On the "Rough Use" of Machine Learning Techniques: a Story about Unrealistic Prediction

Chih-Jen Lin

National Taiwan Univ.

MBZUAI

Outline

Outline

Introduction

- This workshop is about reproducability, a very important topic in machine learning.
- In this talk, I would like to discuss something related, for which I call "inappropriate" or "rough" use of machine learning techniques.
- Typically for "inappropriate use" of ML techniques, we think about cases such as
  - reporting training instead of test performance, or
  - comparing two methods without suitable hyper-parameter searches

Introduction (Cont'd)

- But the reality is that there are more sophisticated examples.
- I will discuss a real and intriguing example.

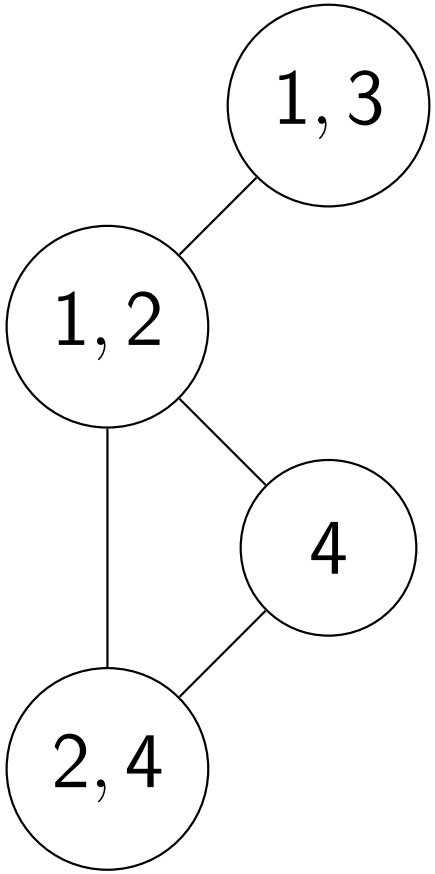
Outline

A Story about Predictions Using Ground Truth

- Predictions using ground truth are impossible in deploying a machine learning model
- But surprisingly unrealistic predictions were used in almost the entire field of graph representation learning
- We reported this story in a paper (Lin et al., 2022)

Graph Representation Learning

- Graph representation learning is a research area to transform a graph into some dense and low dimension embeddings
- This field is quite large, with tens of thousands of papers
- Many use node classification to evaluate the quality of embeddings



Unrealistic Prediction

- A node may have multiple labels, so we have a multi-label classification problem
- The existing prediction process is often as follows
  1. Assumes #associated labels of each test instance is known
  2. Predict this number of labels by selecting those with the largest decision values

Unrealistic Prediction: Example

True labels	Decision values on labels					Prediction	
	1	2	3	4	5	#labels unknown	#labels known
1, 2, 3	0.5	-0.1	0.6	-0.2	-0.5	1, 3	1, 2, 3
4, 5	-0.4	0.2	-0.2	0.6	0.4	2, 4, 5	4, 5
3, 5	-0.7	-0.9	-0.1	-0.4	-0.5		3, 4

- There are five labels; each row is for an instance
- Decision value $\geq 0 \Rightarrow$ has this label; < 0 otherwise

Unrealistic Prediction: Example

True labels	Decision values on labels					Prediction	
	1	2	3	4	5	#labels unknown	#labels known
1, 2, 3	0.5	-0.1	0.6	-0.2	-0.5	1, 3	1, 2, 3
4, 5	-0.4	0.2	-0.2	0.6	0.4	2, 4, 5	4, 5
3, 5	-0.7	-0.9	-0.1	-0.4	-0.5		3, 4

- For the 3rd instance, all decision values are negative
- If # labels is unknown, we will not predict any label

Unrealistic Prediction: Example

True labels	Decision values on labels					Prediction	
	1	2	3	4	5	#labels unknown	#labels known
1, 2, 3	0.5	-0.1	0.6	-0.2	-0.5	1, 3	1, 2, 3
4, 5	-0.4	0.2	-0.2	0.6	0.4	2, 4, 5	4, 5
3, 5	-0.7	-0.9	-0.1	-0.4	-0.5		3, 4

- In the practical use, # labels is unknown
- If # labels is assumed to be known, overestimation tends to occur in evaluation (detailed theory omitted)

Wide Use of Unrealistic Predictions

- People did acknowledge that the setting is unrealistic
- Faerman et al. (2018): "Precisely, this method uses the actual number of labels k each test instance has. [...] In real world applications, it is fairly uncommon that users have such knowledge in advance"

Wide Use of Unrealistic Predictions (Cont'd)

- So why were unrealistic predictions widely used?
- Many papers naturally follow conventions from previous works

  Chanpuriya and Musco (2020): "As in Perozzi et al. (2014) and Qiu et al. (2018), we assume that the number of labels for each test example is given"

Wide Use of Unrealistic Predictions (Cont'd)

- Multi-label classification is considered difficult for researchers in graph-representation learning

  Li et al. (2016): "As the datasets are not only multi-class but also multi-label, we usually need a thresholding method to test the results. But literature gives a negative opinion of arbitrarily choosing thresholding methods"
- We will briefly discuss multi-label classification and explain what the thresholding issue is

Multi-label Classification

- Assume k is the number of labels.
- A simple multi-label method is to assume independence of labels and decompose the problem into k binary sub-problems:

$$f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_k(\boldsymbol{x}))$$

- Then

$$f_j(\boldsymbol{x}) = \begin{cases} \geq 0 & \text{has label } j \\ < 0 & \text{has not} \end{cases}$$

- The strategy is also known as binary relevance

One-vs-rest (Binary Relevance)

- We learn $f_j(\boldsymbol{x})$ by minimizing

  training errors of data with label j

  $+$

  training errors of data without label j

- We call this one (data of one label as positive) versus the rest (data of rest labels as negative)

Problems of One-vs-rest

- Macro-F1 results on three graph representation learning methods (larger better)

Training and prediction methods	Macro-F1		
	DeepWalk	Node2vec	LINE
unrealistic	0.304	0.306	0.258
one-vs-rest	0.195	0.191	0.128

- One-vs-rest has significantly worse performance than unrealistic predictions.

Problems of One-vs-rest (Cont'd)

- Because the data set to get $f_j(\boldsymbol{x})$ is often imbalanced, $f_j(\boldsymbol{x})$ tends to predict that $\boldsymbol{x}$ has no label j
- This issue is well known in the area of multi-label classification, and techniques have long been developed to address the issue
- For example, two useful techniques are
  - thresholding, and
  - cost-sensitive learning (details not shown)

Thresholding Technique

- If

$$f_j(\boldsymbol{x}) \leq 0 \text{ for every test instance } \boldsymbol{x},$$

we can make instances more easily predict label j by considering

$$\Delta_j > 0, \text{ and } f_j(\boldsymbol{x}) \leftarrow f_j(\boldsymbol{x}) + \Delta_j$$

- Δ_j is the threshold value and originally $\Delta_j = 0$

Thresholding Technique (Cont'd)

- Techniques to find suitable Δ_j were developed long time ago (Yang, 2001; Lewis et al., 2004; Fan and Lin, 2007)

Thresholding Technique (Cont'd)

- Results for node classification

Training and prediction methods	Macro-F1		
	DeepWalk	Node2vec	LINE
unrealistic	0.304	0.306	0.258
one-vs-rest	0.195	0.191	0.128
thresholding	0.299	0.302	0.264

- Thresholding achieves much better results than one-vs-rest

Rough Use of Machine Learning

- In graph-representation learning, node classification is used to evaluate the quality of embeddings
- In comparing
  1. embedding generation method A and
  2. embedding generation method B,

  the rank by the unrealistic predictions may be the same as that by an appropriate setting
- Then the unrealistic prediction may be fine

Rough Use of Machine Learning (Cont'd)

- However, the practical deployment can be an issue
- Thus I call this a "rough use" of ML methods: maybe fine in some circumstances, but not appropriate in other situations
- But why people did not use more suitable settings such as thresholding?
- We will give deeper investigation.

Outline

Thresholding Technique

- We have shown in an earlier slide that thresholding is useful
- We noticed this early and have an open-source MATLAB (now also Python) code since long time ago.
- But it seems the thresholding technique is not widely used
- The tens of thousands papers in graph representation learning did not use it.
- Why?

Thresholding Technique (Cont'd)

- Besides the obvious issue that people follow past works, we discuss other interesting aspects.

Deciding the Threshold

- Let's recall the thresholding technique
- If

$$f_j(\boldsymbol{x}) \leq 0 \text{ for every test instance } \boldsymbol{x},$$

we can make instances more easily predict label j by considering

$$\Delta_j > 0, \text{ and } f_j(\boldsymbol{x}) \leftarrow f_j(\boldsymbol{x}) + \Delta_j$$

- Turns out that finding a suitable threshold Δ_j is more difficult than we thought.

Deciding the Threshold (Cont'd)

- For each label j, deciding Δ_j for

$$f_j(\boldsymbol{x}) \leftarrow f_j(\boldsymbol{x}) + \Delta_j$$

  is like to find a hyper-parameter.
- Naturally we think about applying a cross validation (CV) procedure.
- In multi-label classification, a common evaluation criterion is Macro-F1, the average of F-measures overall all labels.

Deciding the Threshold (Cont'd)

- That is,

$$\text{Macro-F1} = \frac{1}{k}\sum_{j=1}^{k} \text{F-measure for label } j.$$

  We will discuss details of F-measure later.
- An important property is that F-measure across labels are independent.
- Thus, naturally we select Δ_j to achieve the best CV F-measure for label j.
- But soon people experimentally found that for rare labels this way is prone to overfitting.

Deciding the Threshold (Cont'd)

- The resulting Δ_j leads to poor test F-measure.
- Strategies to address the overfitting issue were developed by (Yang, 2001; Lewis et al., 2004; Fan and Lin, 2007)
- But they are weird heuristics:

  If CV performance is worse than a specified value,

  $\Delta_j \leftarrow$ the largest validation decision value
- It is difficult to understand the reason of doing so.

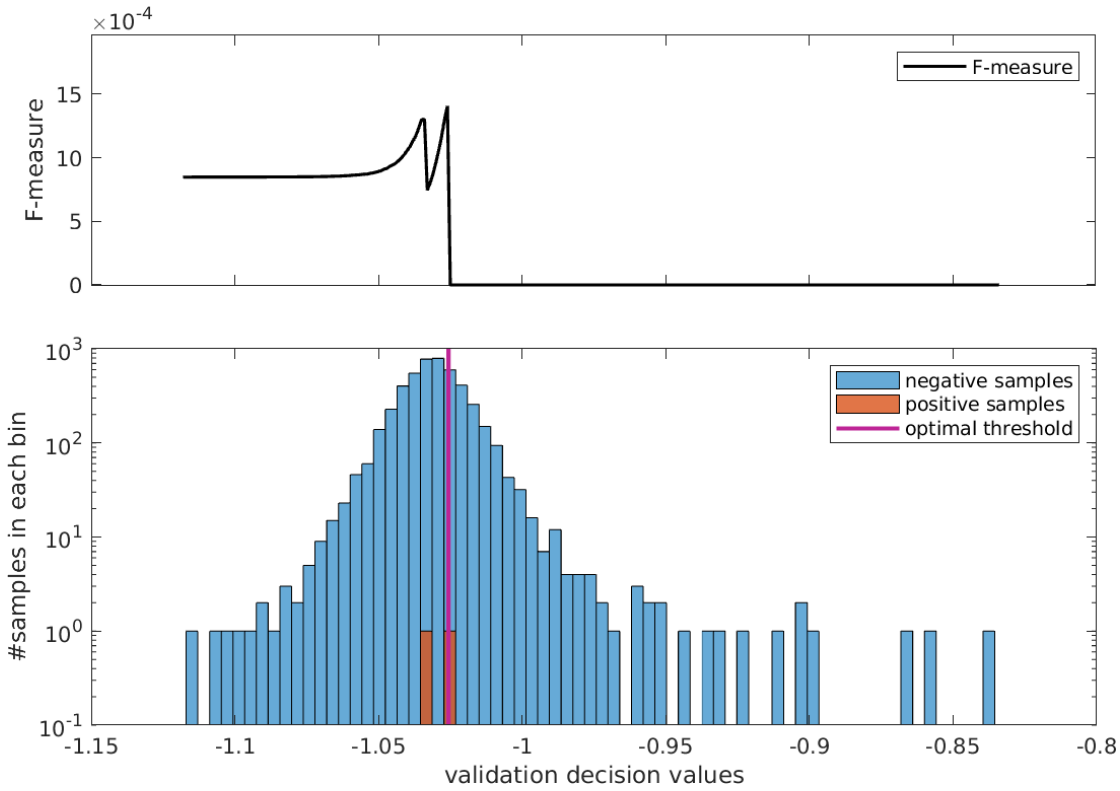
Deciding the Threshold (Cont'd)

- In Fan and Lin (2007) we tried to explain the heuristic, but were not successful. The report was never published.
- Only in Lin and Lin (2023), we fully understood the overfitting phenomenon.

Threshold Selection Overfits F-measure Easily

An example validation set for a rare label, extracted from a real dataset:

- Optimizing F-measure gives a too-low threshold, producing many false positives.
- Why?



Threshold Selection Overfits F-measure Easily (Cont'd)

- F-measure:

$$\mathrm{F} = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}$$

- A reasonable threshold:

$$\mathrm{tp} = 0, \mathrm{fp} = 0, \mathrm{fn} = 2 \text{ implies } \mathrm{F} = 0$$

- A too-low threshold:

$$\mathrm{tp} = 1, \mathrm{fp} = 1000, \mathrm{fn} = 1 \Rightarrow \mathrm{F} = \frac{2}{2 + 1000 + 1} > 0$$

- F-measure is desperate for a non-zero tp no matter the cost.

Fixing the Overfitting Issue

Besides explaining the overfitting issue, in Lin and Lin (2023) we further finish the following results (though details not shown)

- We explain how a previous heuristic mitigates the overfitting, but also point out its problems.
- We propose methods based on "smoothed F-measure," which is backed by detailed analysis and theoretical justifications.

Discussion and Conclusions

- Our discussion shows that thresholding is a sophisticated technique, so many may hesitate to use it.
- Unfortunately, we are not experts of all machine learning methods, so the rough use of machine learning methods is common and sometimes unavoidable.
- We should work together to improve the practical machine learning use.