Data Science and Machine Learning (IM5033)

Dealing with Imbalanced Datasets in Machine Learning

呂晟維[a], 華崧淇[b]

[a] 國立中央大學資訊管理學系 106403551@cc.ncu.edu.tw
[b] 國立中央大學資訊管理學系 schua1013@g.ncu.edu.tw

Abstract

Data imbalancing is a very popular topic in the field of Machine Learning and Data Science. Especially when people focus on the minority class of the data. This article is mainly about the introduction of imbalanced data and its mathematical definition. It will also demonstrate how we classify a dataset as imbalanced data. By showing several data level and algorithm level approaches, we have the aim of reaching a better understanding of Machine Learning.

Keywords: *Imbalanced dataset, Shannon entropy, Degree of imbalanced, Data level and algorithm approach, SMOTE*

1. What is an imbalanced dataset?

Imbalanced dataset normally occurs on classification problems. Since we know in real world cases, the proportion of each class (so called label) may not be exactly the same. The dataset is often skewed. For instance, in binary classification we may gather a dataset with 40% of class A and 60% of class B. Classes that make up a large proportion of the dataset are called **majority classes**. Those make up a smaller proportion are called **minority classes**.

1.1. In which proportion the dataset is imbalanced?

We browse through many articles and papers and provide our own opinion. We can divide the question into 3 circumstances depending on the degree of imbalance. If the dataset is mild imbalanced, it should not cause any significant performance degradation. On the other hand, if the dataset is moderately or extremely imbalanced, we have less minority classes to train. The performance of the model will not be as effective and we will like to proceed with some modifications on data.

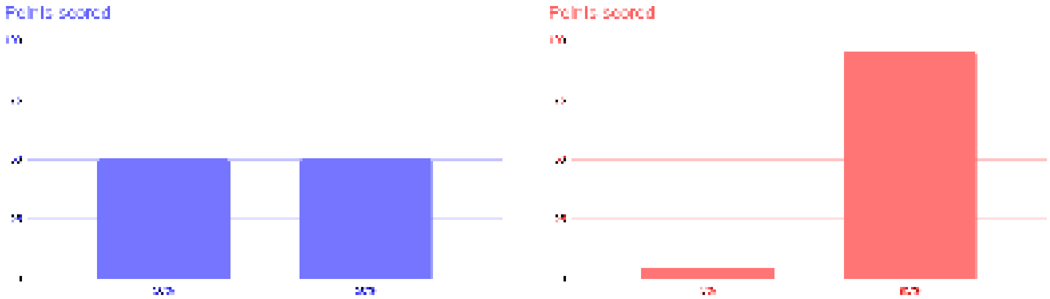


Fig. 1. (a) ideal distribution in binary classification; (b) extremely imbalanced case in binary classification

This table shows our definition of degree of imbalance. Sometimes visualization may be ambiguous, we can determine the degree of imbalance by calculating the proportion of minority class. If the minority class accounts for more than 30 percent, it is balanced and does not need any modification.

Table 1. Determinants of degree of imbalance

Degree of imbalance	Proportion of minority class	Imbalanced	Need modify
Mild	> 30%	No	No
Moderate	10 ~ 30%	Yes	Yes
Extreme	< 10%	Yes	Yes

1.2. Is there any more professional measurement? Introducing Shannon Entropy

From the above articles, we find that the proportion of minority class is a clear determinant of imbalance. However, there is a more statistical and more professional criteria - *entropy*. Entropy originates from thermodynamics measurement in physics. In 1948, Claude E. Shannon provided that in information theory, entropy is a counter-measurement of uncertainty. In this case, we can regard uncertainty as the degree of imbalance in the dataset. The higher the entropy is, the less uncertainty/degree of imbalance it has. The entropy of *x* is formally defined as:

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

(Note: P of *xi* is the probability of *xi* in the distribution)

In statistic or machine learning, we can describe as follows: On a data set of *n* instances, if you have *k* classes of size *ci* you can compute entropy as:

$$H = -\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}.$$

(Note: *log* is base of 2)

Number of classes *k* may vary from different dataset. For example, the entropy of a balanced binary classification dataset is 1($-\sum_{i=1}^{2}\frac{1}{2}\log_2\frac{1}{2}$), while it's 1.58($-\sum_{i=1}^{3}\frac{1}{3}\log_2\frac{1}{3}$) with a balanced 3-class dataset. As a result, we divide *H* with *log2(k)* to deal with different numbers of classes. We will name this modified entropy value as *balance factor* or *balance entropy* in this article.

$$\text{Balance} = \frac{H}{\log k} = \frac{-\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}.}{\log k}$$

Therefore, we can view *balance factor* = 0 for a unbalanced dataset and *balance factor* = 1 for a totally balanced dataset.

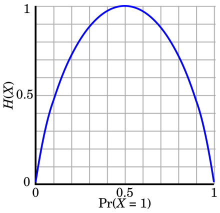


Fig. 2. Entropy equals 1 when both probabilities equals 0.5 in binary situation. (image from wikipedia)

Last, we re-define the determinants of degree of balance:

Table 2. Determinants with balance entropy

Degree of imbalance	Proportion of minority class	Balance entropy	Imbalanced	Need modify
Mild	> 30%	0.8813 < e	No	No
Moderate	10 ~ 30%	0.4690 < e < 0.8813	Yes	Yes
Extreme	< 10%	e < 0.4690	Yes	Yes

2. Introduce an imbalanced dataset

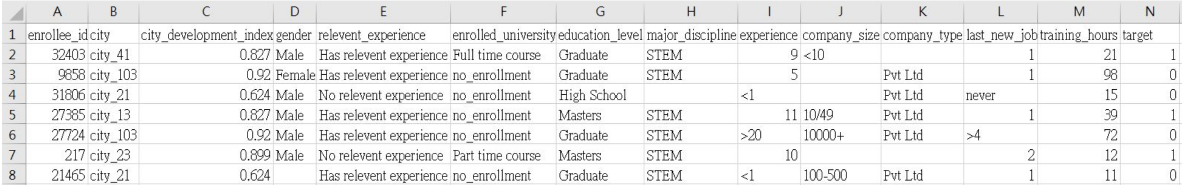
2.1. Source of data & What is about the data

The dataset (HR Analytics: Job Change of Data Scientists) is from Kaggle, an online data science community with a lot of open data and open resources. The data is released from a company that wants to hire data scientists among people who successfully pass some courses which are conducted by the company. HR researchers could predict the probability of a candidate to look for a new job or will work for the company, which not only reduces the cost and time but also helps categorize candidates.

Since there are so many qualified candidates and a few are deciding to seek new jobs. The dataset is apparently imbalanced. We will introduce the data as well as focus on its imbalancing in this section.

2.2. Attributes & Target

The dataset contains 14 columns with 21,287 records with easy understanding titles and contents. Most features are categorical including nominal, ordinal and binary format. Target 0 means the candidate is not looking for job change, target 1 means he or she is looking for a job change. It also contains empty values. There is a need for data preprocessing before applying to models and algorithms. Above all is the dataset is imbalanced.

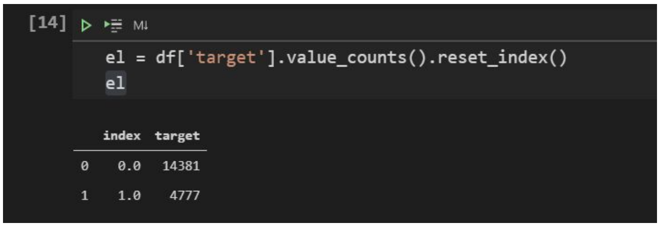


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours	target
2	32403	city_41	0.827	Male	Has relevent experience	Full time course	Graduate	STEM	9	<10		1	21	1
3	9858	city_103	0.92	Female	Has relevent experience	no_enrollment	Graduate	STEM	5		Pvt Ltd	1	98	0
4	31806	city_21	0.624	Male	No relevent experience	no_enrollment	High School		<1		Pvt Ltd	never	15	0
5	27385	city_13	0.827	Male	Has relevent experience	no_enrollment	Masters	STEM	11	10/49	Pvt Ltd	1	39	1
6	27724	city_103	0.92	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	10000+	Pvt Ltd	>4	72	0
7	217	city_23	0.899	Male	No relevent experience	Part time course	Masters	STEM	10			2	12	1
8	21465	city_21	0.624		Has relevent experience	no_enrollment	Graduate	STEM	<1	100-500	Pvt Ltd	1	11	0

Fig. 3. pieces of dataset

2.3. Imbalancing of the dataset

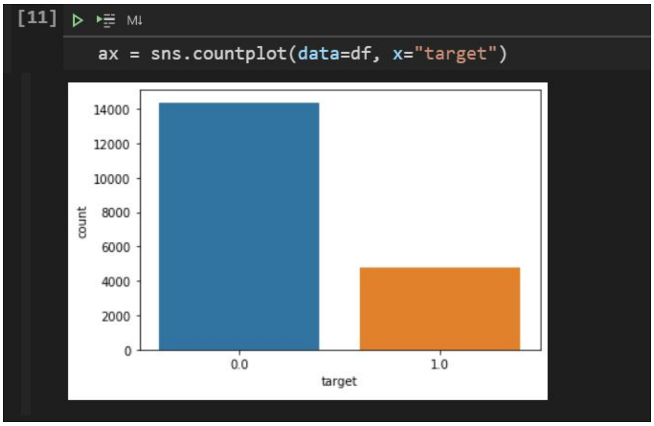
Here we calculate the count of each target regardless of the null values. It has 14,381 records of label ‘0’ and 4,777 records of label ‘1’. Let assump the majority class, label ‘0’ as positive and label ‘1’ as negative. As the percentage of negative class equals *4,777 / (4,777 + 14,381) = 4,777 / 19,158 = 24.93%*. We can classify the dataset as moderate imbalanced.



```
[14]
el = df['target'].value_counts().reset_index()
el

   index  target
0    0.0   14381
1    1.0    4777
```

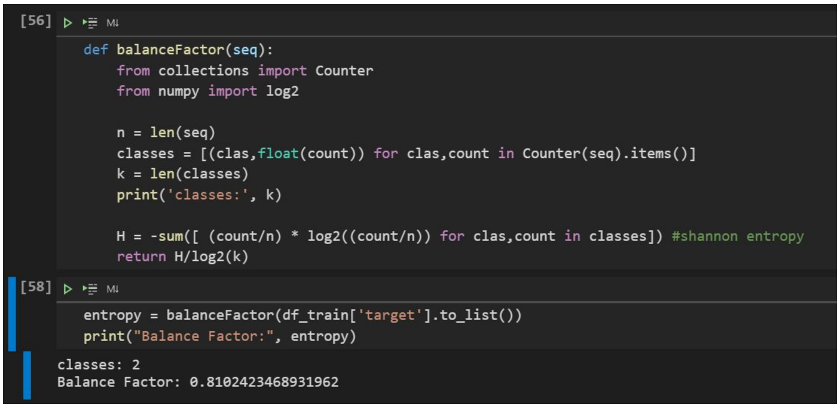
Fig. 4. value counts of target



```
[11]
ax = sns.countplot(data=df, x="target")
```

Fig. 5. visualization of value counts

Then we implement the balance factor with *entropy / log(number of classes)*, resulting in 0.8102, which also falls in the moderate imbalanced interval from table 2 (0.4690 < e < 0.8813).



```
[56]
def balanceFactor(seq):
    from collections import Counter
    from numpy import log2

    n = len(seq)
    classes = [(clas,float(count)) for clas,count in Counter(seq).items()]
    k = len(classes)
    print('classes:', k)

    H = -sum([ (count/n) * log2((count/n)) for clas,count in classes]) #shannon entropy
    return H/log2(k)
[58]
entropy = balanceFactor(df_train['target'].to_list())
print("Balance Factor:", entropy)

classes: 2
Balance Factor: 0.8102423468931962
```

Fig. 6. function measuring balance factor for any input sequence (modified from stack-exchange)

2.4. More discovery about the data

Via visualizing the cardinality between several attributes. We find that there is something to do with education level v.s. relevant experiences, etc. While some attributes are imbalanced either, for example the education level of the qualified candidates are most graduate students. But we regard these circumstances as a kind of distribution. We only focus on the percentage of each target as the variance of imbalancing.

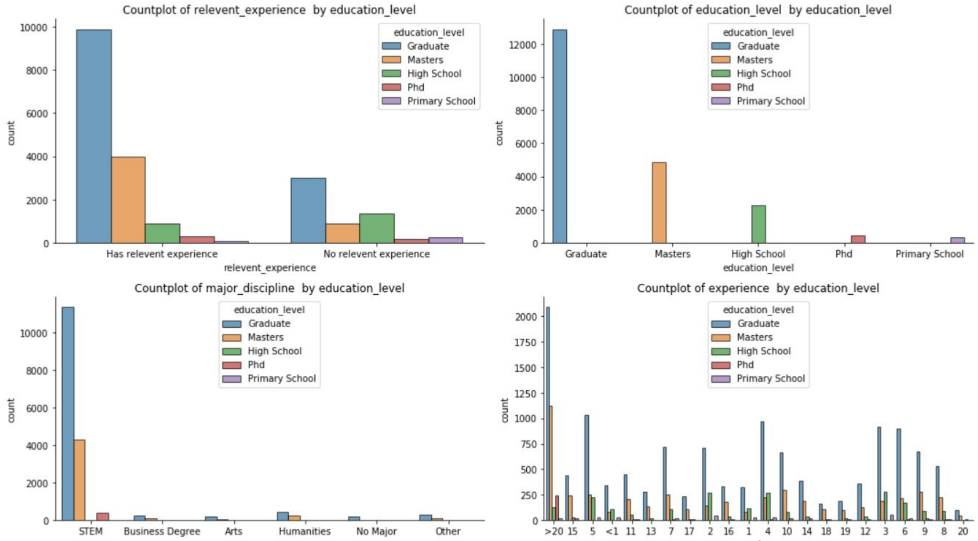


Fig. 7. visualization between attributes (extract from Kaggle's notebook)

2.5. Under what circumstances is it usual to collect imbalanced data?

When producing a questionnaire census, we always try our best to make sure that the samples can represent the population. But in certain work fields or domains, the population itself is imbalanced. Most of the time, people focus on the minority cases which appear on a rare probability. Such as:

- **Medical history in hospitals**: The number of people diagnosed with cancer during general medical examination is much less than the number of healthy people.
- **Settlement of claim in insurance companies**: The number of insurance claims filed is much lower than the total number of people who purchased insurance.
- **Digital advertising click rate**: People who will click on the ads make up a minority of the audience, and those who will successfully purchase the product are the minority of those who click on the ad.
- **Fraud detection**: The ordinary transfers in banks are legitimate transactions, and unusual transfers such as fraudulent transactions or money laundering rarely occur.
- **Gander**: College of Science and Engineering has more male students than female students, while College of Liberal Arts in contrast.

3. Survey on what has been done to tackle the imbalanced problems?

3.1. Classifiers/Algorithms approaches

To fix the imbalanced data problem, we could handle it at the model level. Most machine learning algorithms don't perform well with biased class data. However, we can modify the current training algorithm to level up the performance. This can be accomplished by giving different weights to both the majority and minority classes. The difference in weights will influence the classification of the classes during the training progress. The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class.

Let's take a look into a simple weight of class count. To calculate weight, we could use the formula:

wj = n_samples / (n_classes * n_samplesj)

Here j represents the class, n_samples represent the total number of samples or rows in the dataset, n_classes is the total number of unique classes in the dataset and n_samplesj is the total number of rows of the respective class.

For example: we got 2 classes, the first class has 99990 samples and the second class has 10 samples. By applying the formula above, we can get w1 equals nearly 0.5 and w2 will be 5000.

So why do we use weight? The reason for this is because as the model trains, the error will be multiplied by the weight of the data. The estimator will try to minimize error on the more heavily weighted classes, because they will have a greater effect on error. Without applying different weights, the model treats each data equally important. As a result, we can force the estimator to emphasize the data more important which is the classes that have the larger weight.

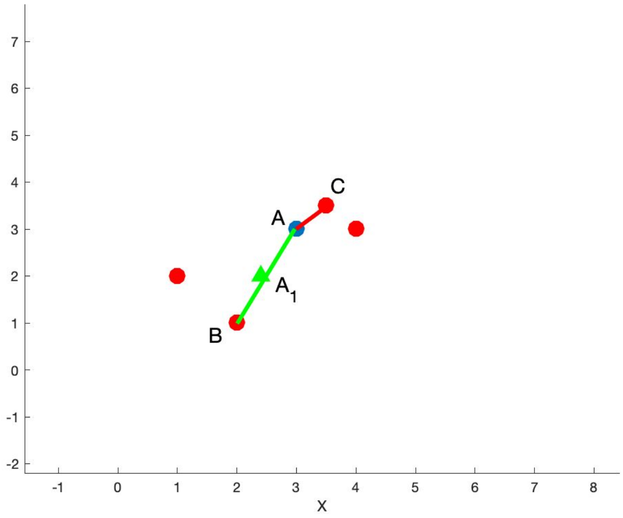
3.2. Data level approaches : resampling techniques

As we know there are two different kinds of imbalanced dataset, positive class as the majority class, and the other one is negative class as the majority class. Sometimes you will find that the minority class data are hard to equal to the amount of majority class. In this occasion, you might want to apply oversampling or undersampling methods to improve your model performance. Undersampling techniques remove data from the majority class in order to balance the class distribution, for instance reducing the skew from a 1:100 to a 1:10, 1:3, or even a 1:1 class distribution. On the other hand, Oversampling duplicate datas from the minority class and adding it to the training dataset so that it reduces the skew in the class distribution.

The simplest undersampling technique involves randomly selecting datas from the majority class and deleting them from the training dataset. This is referred to as random undersampling. For oversampling, the simplest technique is also random sampling. Random oversampling involves randomly selecting data from the minority class, with replacement, and adding them to the training dataset.

Now let's talk about the pros and cons. The disadvantage with undersampling is that it discards potentially useful data. This means that datas is removed without concern for how useful or important they might be. About the disadvantage of oversampling is that by making the same copies of existing examples, it is more possible to encounter overfitting and also due to the increase of training data, the training time will climb.

Moreover, let’s talk about a more advanced technique Synthetic Minority Oversampling Technique or in abbreviation SMOTE. Basically it’s an oversampling technique to create more minority class data. First we randomly select a data point as point A. Secondly we randomly select k nearest data point against A, we can also call it k nearest neighbor. Then we select N amount of the length of all length between A and nearest neighbor. Finally multiply each length by a random number between the range [0,1] respectively. Consequently we will get N data points we newly generated so that we could insert into the minority class. This approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively close in feature space to existing examples from the minority class. However the downside of it is that it is still possible to overlap between minority class and majority class due to the complete ignorance of the majority class data point.



References

[1] Möbius, HR Analytics: Job Change of Data Scientists (2021), *Kaggle*:
https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists

[2] Siti Khotijah, Predict who will move to a new job (2021)
https://www.kaggle.com/khotijahs1/predict-who-will-move-to-a-new-job

[3] Ankit Gupta, Who will leave a job (2021)
https://www.kaggle.com/nkitgupta/who-will-leave-a-job

[4] Google, Imbalanced Data
https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

[5] Devi Ramu, What is an imbalanced dataset? (2020)
https://www.quora.com/What-is-an-imbalanced-dataset

[6] Stackxchange, A general measure of data-set imbalance (2017)
https://stats.stackexchange.com/questions/239973/a-general-measure-of-data-set-imbalance

[7] Guest Blog, Imbalanced Data : How to handle Imbalanced Classification Problems (2017)
Class Imbalance | Handling Imbalanced Data Using Python

[8] Will Badr, Having an Imbalanced Dataset? Here Is How You Can Fix It (2019)
https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

[9] Jason Brownlee, Undersampling Algorithms for Imbalanced Classification (2020), SMOTE for Imbalanced Classification with Python(2020)
https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/
https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[10] Franck Dernoncourt, Opinions about Oversampling in general, and the SMOTE algorithm in particular (2017)
https://stats.stackexchange.com/questions/234016/opinions-about-oversampling-in-general-and-the-smote-algorithm-in-particular