ulm university universität ulm

Metrics for Multi-class and Multi-label Classification

Maximilian Wenzel
Data Science Seminar Summer Term 2021

Motivation for Metrics in Classification

- Classification: Categorize an instance/sample into a class or multiple classes
- How to determine the performance of a classifier?
  - Count the number of correct and incorrect predictions
  - Summarize counts using evaluation metrics
- Problems with metrics
  - Metrics usually not standardized for application domains
  - Small variations in metrics may even lead to different classifier rankings
  - Number of possibilities to evaluate classifiers for multi-class and multi-label problems increases

Binary Classification: Confusion Matrix

– Counts the number of correct and incorrect predictions of classifier h

		Actual Class: *Cat*	Actual Class: *Not Cat*	Predictions per Class
Predicted Class	*Cat*	9	2	11
Predicted Class	*Not Cat*	1	8	9
Instances per Class		10	10	20

– Raw confusion matrix is difficult to interpret

➔ Use metrics to summarize absolute confusion matrix values

Excerpt of Binary Classification Metrics

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

- **Recall**: Percentage of instances that have been correctly classified as positive

$$r = \frac{TP}{TP + FN}$$

- **Precision**: Percentage of positive predictions that were actually correct

$$p = \frac{TP}{TP + FP}$$

- $\boldsymbol{F_1}$**-score**: harmonic mean of **recall** and **precision**

$$F_1 = \frac{2 \cdot p \cdot r}{p + r} = \left(\frac{p^{-1} \cdot r^{-1}}{2}\right)^{-1}$$

$$p = \frac{TP}{TP + FP}$$

Issue with Imbalanced Datasets

Balanced Dataset:
Equal amount of instances per class

		Actual Class: Cat	Actual Class: $Not\ Cat$	Predictions per Class
Predicted Class	Cat	9	2	11
	$Not\ Cat$	1	8	9
Instances per Class		10	10	20

$$p = \frac{9}{9+2} = \frac{9}{11} \approx 0.82$$

Imbalanced Dataset:
Different amount of instances per class

		Actual Class: Cat	Actual Class: $Not\ Cat$	Predictions per Class
Predicted Class	Cat	9	4	13
	$Not\ Cat$	1	16	17
Instances per Class		10	20	30

$$p = \frac{9}{9+4} = \frac{9}{13} \approx 0.69$$

- Although same proportion of $\frac{TP}{FN}$ and $\frac{FP}{TN}$ different results for metric
- Metrics which use values from both "actual class" columns are sensitive to imbalanced datasets

Multi-class Classification

- Given:
  - Instance $x \in \mathcal{X} \subseteq \mathbb{R}^d$
  - Label space $\mathcal{Y} \subseteq \{0,1\}^m$, one-hot-coded vectors
  - Classifier $h\colon \mathcal{X} \to \mathcal{Y}$, predicts exactly **one** class per instance

Example: Cat $y_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}$, Dog $y_2 = \begin{pmatrix}0\\1\\0\end{pmatrix}$, Mouse $y_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}$

What kind of animal is this?

$x \in \mathcal{X}$  $\longrightarrow h \longrightarrow$ Cat, $\begin{pmatrix}1\\0\\0\end{pmatrix} \in \mathcal{Y}$ ✓

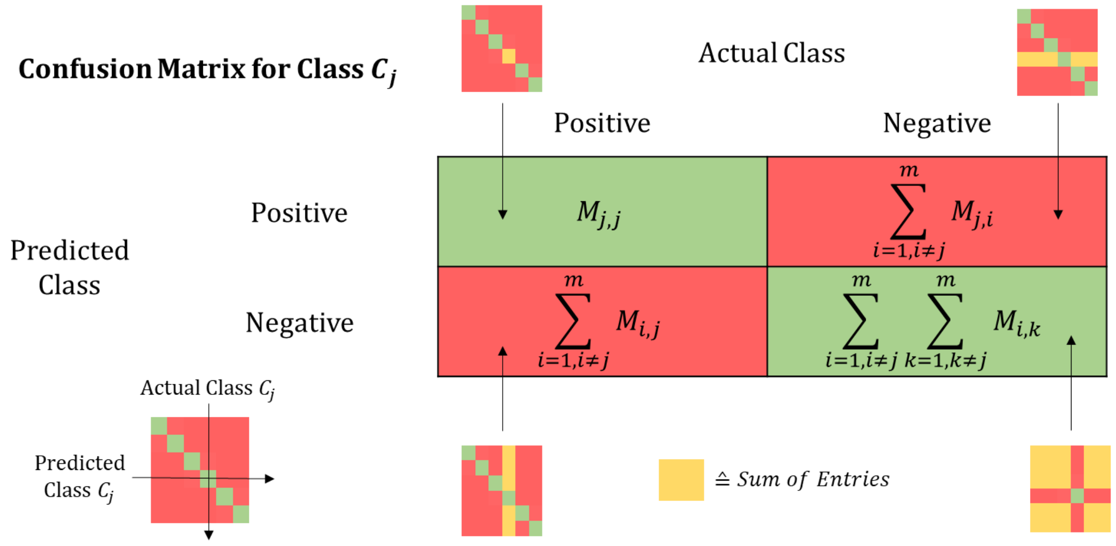
Multi-class Confusion Matrix

		Actual Class: Cat	Dog	$Mouse$	Predictions per Class
Predicted Class	Cat	9	3	1	13
	Dog	1	6	2	9
	$Mouse$	0	1	7	8
Instances per Class		10	10	10	30

- Confusion matrix becomes more complex: For m classes, $m \times m$ confusion matrix
- How to summarize now the performance of a given classifier?
- **Solution**:
  - Create for each class C_j a binary confusion matrix
  - Summarize all per-class results using an **averaging strategy**

Per-class Confusion Matrix

- Converts the problem into a binary classification problem
  - Class C_j and class "not C_j"



- Previously introduced metrics can be computed **per class**

- **Problem**: How to summarize the results over all classes?

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Averaging Strategies

– **Macro Averaging**: Arithmetic mean of all per-class metrics

Example: macro-averaged Recall → $$r_M = \frac{1}{m}\sum_{j=1}^{m}\frac{TP_j}{TP_j + FN_j}$$

All per-class results weighted equally

– **Micro Averaging**: Sum up numerator and denominator separately of the appropriate metric and compute the result

$$r_\mu = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m} TP_j + FN_j}$$

Sensitive to imbalanced datasets

– **Weighted Averaging**: weight the per-class metrics by the number of instances of the appropriate class

$$r_w = \frac{1}{n}\sum_{j=1}^{m}\frac{n_j \cdot TP_j}{TP_j + FN_j}$$

Intentionally weighted by number of instances per class

Averaging the F_1-score

– Micro-averaged F_1 analogous to the standard approach:

$$F_{1\mu} = \frac{2 \cdot p_\mu \cdot r_\mu}{p_\mu + r_\mu} = p_\mu = r_\mu$$

Since $\sum_{j=1}^{m} FP_j = \sum_{j=1}^{m} FN_j$

– Two distinct approaches to compute the macro-averaged F_1-score

- $\mathcal{F}_1$, the averaged F_1

$$\mathcal{F}_1 = \frac{1}{m}\sum_{j=1}^{m} \frac{2 \cdot p_j \cdot r_j}{p_j + r_j}$$

The standard approach, recommended by Opitz and Burst (2019)

- $\mathbb{F}_1$, the F_1 of averages

$$\mathbb{F}_1 = \frac{2 \cdot p_M \cdot r_M}{p_M + r_M}$$

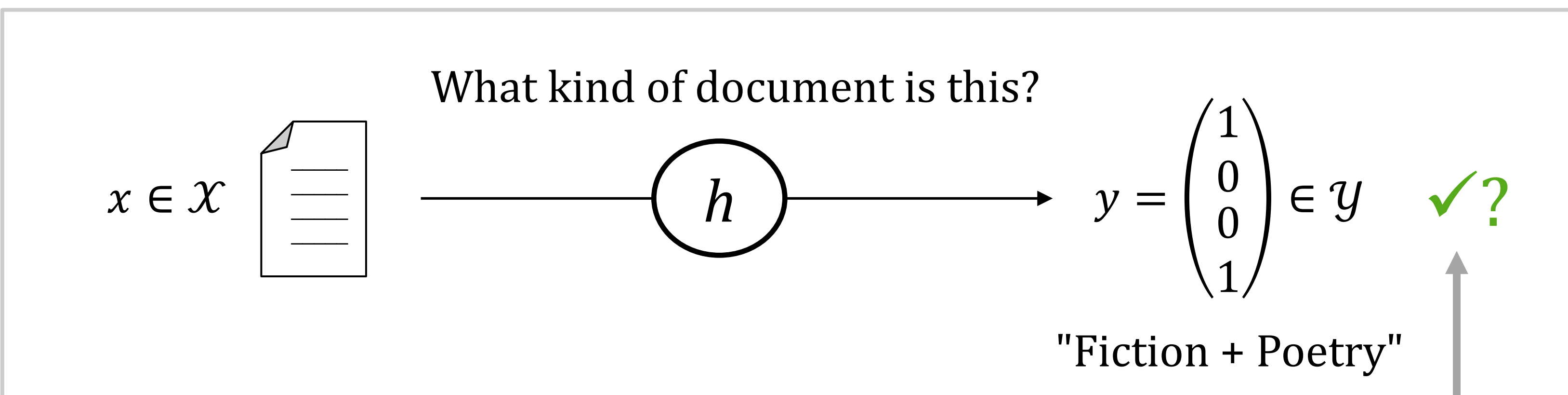
Individual values p_j and r_j not as much influence
➔ May be overly benevolent

➔ Different strategies also applicable to the weighted-aproach

Multi-label Classification

- Given:
  - Instance $x \in \mathcal{X} \subseteq \mathbb{R}^d$
  - Label space $\mathcal{Y} \subseteq \{0,1\}^m$
  - Classifier $h: \mathcal{X} \rightarrow \mathcal{Y}$, may predict **multiple** classes/labels per instance

Example: Text classification



What if prediction is only partially correct?

Multi-label Classification: Viewpoint of Correctness

Multi-label Correctness

Per-class

Evaluation and averaging strategies identical to multi-class case

Per-instance

- Consider each instance separately
- How many labels have been predicted correctly/incorrectly?

➔ Derive **per-instance confusion matrix**

Summarize confusion matrices with averaging strategies

Per-instance Evaluation: Which Averaging Strategies?

- Per-instance evaluation makes only sense with **macro averaging strategies → each instance is equally weighted**
  - Micro- and weighted-averaged result would weight instances differently

- **Example**: weighted-average
  - Each per-instance result is weighted by the factor $TP_j + FN_j$ per instance x_j

	Meaning	
	Per-class	**Per-instance**
$\boldsymbol{TP_j + FN_j}$	#instances per class C_j	#labels in actual label set y_j

Best Practice When Dealing with Metrics

– Always explicitly indicate which metric has been deployed
  - Include the metric as equation
  - If the metric has been implemented by a library (e.g. Python SciKit-learn), look up the concrete implementation
  - If possible include the test dataset evaluation
    ➔ Computation of metric can be reproduced

MESINESP task: JSON file of the test dataset evaluation


```
{ "documents": [
{ "id": "id_test_article_1", "labels": [ "code1", "code2", "code3" ] } ,
{ "id": "id_test_article_2", "labels": [ "code5", "code2", "code21" ] }
] }
```


Source: https://temu.bsc.es/mesinesp2/evaluation/