Bài 8: ĐÁNH GIÁ MÔ HÌNH PHÂN LỚP

(Áp dụng chủ yếu với mô hình phân lớp nhị phân)

1. Mô hình phân lớp

Một cách tổng quát mô hình phân lớp có thể được phát biểu là:

$$h_w(X): R^{m\times n} \rightarrow \{c_i\}_{i=1}^{k} \quad (1)$$

Hiệu năng của mô hình phân lớp được đánh giá thông qua đối sánh giá trị dự đoán $\hat{y} = h_w(X_{test})$ với y_{test}. Trong đó $\hat{y}$ và $y_{test} \in \{c_i\}_{i=1}^{k}$. Các phương pháp phổ biến dùng để đánh giá hiệu năng của mô hình ***phân lớp nhị phân*** được trình bày trong mục 2.

2. Các chỉ số thông dụng

2.1. Độ chính xác – Accuracy score

$$accuracy(\hat{y}, y_{test}) = \frac{1}{|\hat{y}|}\sum_{i=1}^{|\hat{y}|}(\hat{y}^i = y_{test}^i) \quad (2)$$

Ví dụ:

```
>>> import numpy as np
>>> from sklearn.metrics import accuracy_score
>>> y_pred = [0, 2, 1, 3]
>>> y_true = [0, 1, 2, 3]
>>> accuracy_score(y_true, y_pred)
0.5
>>> accuracy_score(y_true, y_pred, normalize=False)
2
```

2.2. Độ chính xác *k* mục dự đoán đầu tiên – top-k accuracy

$$top-k\ accuracy(\hat{y}, y_{test}) = \frac{1}{|\hat{y}|}\sum_{i=1}^{|\hat{y}|}\sum_{j=1}^{k}(\hat{y}^{i,j} = y_{test}^i) \quad (3)$$

Ví dụ:

```
>>> import numpy as np
>>> from sklearn.metrics import top_k_accuracy_score
>>> y_true = np.array([0, 1, 2, 2])
>>> y_score = np.array([[0.5, 0.2, 0.2],
...                     [0.3, 0.4, 0.2],
...                     [0.2, 0.4, 0.3],
...                     [0.7, 0.2, 0.1]])
```

```
>>> top_k_accuracy_score(y_true, y_score, k=2)
0.75
>>> # Not normalizing gives the number of "correctly" classified samples
>>> top_k_accuracy_score(y_true, y_score, k=2, normalize=False)
3
```

2.3. Ma trận '*hỗn hợp*' – Confusion matrix

Đối với mô hình phân lớp, một số thuật ngữ chuyên ngành sau cần nắm bắt kỹ:

- true positives, true negatives, false positives, và false negatives;
- Thuật ngữ positive và negative biểu đạt kết quả dự đoán của mô hình phân lớp;
- Thuật ngữ true và false biểu đạt kết quả dự đoán của mô hình phân lớp là đúng hay sai;
- Như vậy, true positive có thể hiểu là mô hình dự đoán positive là đúng; tương tự như vậy các bạn có thể giải nghĩa các thuật ngữ true negatives, false positives, và false negatives còn lại.

Confusion matrix Sources: [6][7][8][9][10][11][12][13][14] view · talk · edit

		Predicted condition			
	Total population = P + N	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) = TPR + TNR − 1	Prevalence threshold (PT) = $\frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP)**, hit	**False negative (FN)**, type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power = $\frac{TP}{P}$ = 1 − FNR	False negative rate (FNR), miss rate = $\frac{FN}{P}$ = 1 − TPR
	Negative (N)	**False positive (FP)**, type I error, false alarm, overestimation	**True negative (TN)**, correct rejection	False positive rate (FPR), probability of false alarm, fall-out = $\frac{FP}{N}$ = 1 − TNR	True negative rate (TNR), specificity (SPC), selectivity = $\frac{TN}{N}$ = 1 − FPR
	Prevalence = $\frac{P}{P+N}$	Positive predictive value (PPV), precision = $\frac{TP}{PP}$ = 1 − FDR	False omission rate (FOR) = $\frac{FN}{PN}$ = 1 − NPV	Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$	Negative likelihood ratio (LR−) = $\frac{FNR}{TNR}$
	Accuracy (ACC) = $\frac{TP+TN}{P+N}$	False discovery rate (FDR) = $\frac{FP}{PP}$ = 1 − PPV	Negative predictive value (NPV) = $\frac{TN}{PN}$ = 1 − FOR	Markedness (MK), deltaP (Δp) = PPV + NPV − 1	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$
	Balanced accuracy (BA) = $\frac{TPR+TNR}{2}$	F_1 score = $\frac{2PPV \times TPR}{PPV+TPR}$ = $\frac{2TP}{2TP+FP+FN}$	Fowlkes–Mallows index (FM) = $\sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) = $\sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index = $\frac{TP}{TP+FN+FP}$

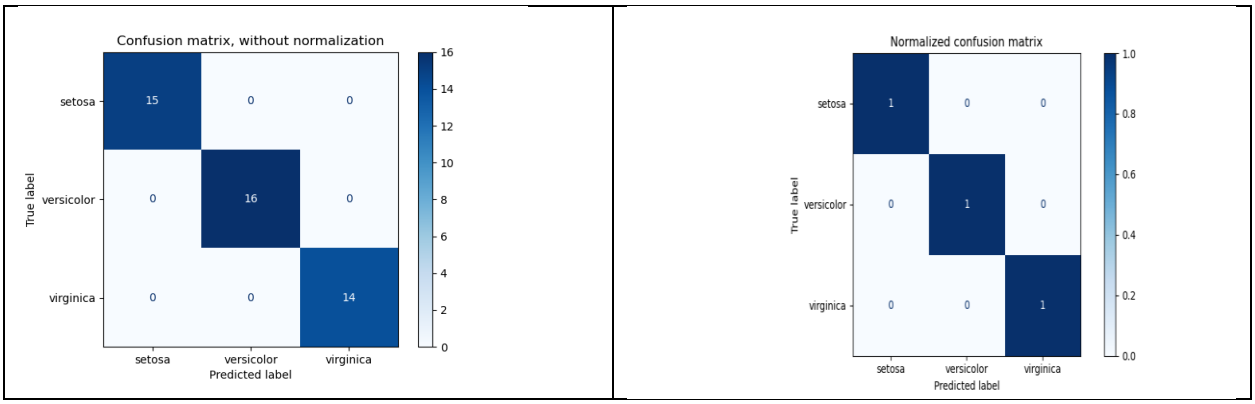
Hình 1: Ma trận hỗn hợp và các phép đo liên quan (nguồn: wikipedia)

Ví dụ: Hiển thị ma trận hỗn hợp dựa vào kết quả dự đoán của mô hình GaussianNB trên tập dữ liệu Iris.

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import datasets
from sklearn.naive_bayes import GaussianNB
```

```
#B1: Đọc dữ liệu
iris = datasets.load_iris()
X = iris.data
y = iris.target
print('X shape: ', X.shape, '; y shape: ', y.shape)
#Xác định số lượng mẫu của từng nhãn lớp - balanced dataset?
unique, counts = np.unique(y, return_counts=True)
result = dict(zip(unique, counts))
print(result)

#B2: Phân chia train - test
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    random_state=15,
                                                    shuffle=True,
                                                    test_size=0.30)
#B3: Khởi tạo và huấn luyện mô hình
model = GaussianNB()
model.fit(X_train, y_train)

#B4: Dự đoán
y_hat = model.predict(X_test)

import matplotlib.pyplot as plt
from sklearn.metrics import ConfusionMatrixDisplay

#B5: Vẽ ma trận hỗn hợp
titles_options = [
    ("Confusion matrix, without normalization", None),
    ("Normalized confusion matrix", "true"),
]
class_names = iris.target_names
for title, normalize in titles_options:
    disp = ConfusionMatrixDisplay.from_estimator(
        model,
        X_test,
        y_test,
        display_labels=class_names,
        cmap=plt.cm.Blues,
        normalize=normalize,
    )
    disp.ax_.set_title(title)

    print(title)
    print(disp.confusion_matrix)

plt.show()
```

Kết quả hiển thị



2.4. Precision và Recall

Công thức tính Precision và Recall dựa vào bảng ma trận hỗn hợp như sau:

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

Chỉ số Precision biểu đạt khả năng mà mô hình phân lớp sẽ không gán nhãn positive cho mẫu có nhãn negative. Trong khi, chỉ số Recall biểu đạt khả năng tìm ra tất cả các mẫu positive của mô hình phân lớp.

Thư viện sklearn cung cấp hàm *precision_score* (tham khảo để biết ý nghĩa các tham số)

```
>>> from sklearn.metrics import precision_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> precision_score(y_true, y_pred, average='macro')
0.22...
>>> # multilabel classification
>>> y_true = [[0, 0, 0], [1, 1, 1], [0, 1, 1]]
>>> y_pred = [[0, 0, 0], [1, 1, 1], [1, 1, 0]]
>>> precision_score(y_true, y_pred, average=None)
array([0.5, 1. , 1. ])
```

Tương tự, thư viện sklearn cung cấp *hàm recall_score* để phục vụ tính toán chỉ số Recall.

```
>>> from sklearn.metrics import recall_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> recall_score(y_true, y_pred, average='macro')
0.33...
```

```
>>> # multilabel classification
>>> y_true = [[0, 0, 0], [1, 1, 1], [0, 1, 1]]
>>> y_pred = [[0, 0, 0], [1, 1, 1], [1, 1, 0]]
>>> recall_score(y_true, y_pred, average=None)
array([1. , 1. , 0.5])
```

2.5. Phép đo *F1 – F1 measure*

Chỉ số F1 được tính theo công thức (6):

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

Giá trị của chỉ số F1: $0 \leq F1 \leq 1$ (0 là trường hợp xấu nhất; 1 là trường hợp tốt nhất).

Ví dụ:

```
>>> from sklearn.metrics import f1_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> f1_score(y_true, y_pred, average='macro')
0.26...
>>> # multilabel classification
>>> y_true = [[0, 0, 0], [1, 1, 1], [0, 1, 1]]
>>> y_pred = [[0, 0, 0], [1, 1, 1], [1, 1, 0]]
>>> f1_score(y_true, y_pred, average=None)
array([0.66666667, 1.        , 0.66666667])
```

2.6. Báo cáo phân lớp tổng hợp

Thư viện sklearn cung cấp báo cáo tổng hợp về hiệu năng phân lớp tính theo các chỉ *precision, recall* và *f1- measure* thông qua *hàm classification_report*.

```
 >>> from sklearn.metrics import classification_report
>>> y_test = [0, 1, 2, 2, 0]
>>> y_hat = [0, 0, 2, 1, 0]
>>> target_names = ['class 0', 'class 1', 'class 2']
>>> print(classification_report(y_hat, y_test, target_names=target_names))
              precision    recall  f1-score   support

     class 0       0.67      1.00      0.80         2
     class 1       0.00      0.00      0.00         1
     class 2       1.00      0.50      0.67         2

    accuracy                           0.60         5
   macro avg       0.56      0.50      0.49         5
weighted avg       0.67      0.60      0.59         5
```