



LEONARDO WAKED
DAVID CHIA
JASON SALCEDO

MATRICES DE CONFUSION

2024

MATRIZ DE CONFUSIÓN : ¿QUÉ ES?

LA MATRIZ DE CONFUSIÓN, TAMBIÉN CONOCIDA COMO MATRIZ DE ERROR, ES UN INSTRUMENTO TECNOLÓGICO QUE SIRVE PARA CALCULAR EL RENDIMIENTO SOBRE UN MODELO DE CLASIFICACIÓN DEFINIDO. DE ESTE MODO, ES POSIBLE PREDDECIR FÁCILMENTE POR EJEMPLO LOS CORREOS ELECTRÓNICOS QUE DEBEN SER CLASIFICADOS COMO SPAM. ES MÁS, ESTA HERRAMIENTA ES UNA TABLA DE PREDICCIONES DOTADA DE INTELIGENCIA ARTIFICIAL (IA) QUE VISUALIZA EL DESEMPEÑO DE UN ALGORITMO.

		Actual Values	
		Yes	No
Predicted Values	Yes	True Positive	False Positive
	No	False Negative	True Negative

COMO FUNCIONA



LA MATRIZ DE CONFUSIÓN PROPORCIONA UN MEDIO PARA EVALUAR EL ÉXITO DE UN PROBLEMA DE CLASIFICACIÓN Y DÓNDE SE COMETEN ERRORES (ES DECIR, DÓNDE SE VUELVE "CONFUSO").

EJ MATRIZ DE
CONFUSION

0	1.00	1.00	1.00	22
1	0.78	0.69	0.73	26
2	0.93	0.93	0.93	29
3	1.00	0.89	0.94	27
4	0.92	0.96	0.94	23
5	0.96	0.70	0.81	33
6	0.97	0.97	0.97	35
7	0.94	0.91	0.92	33
8	0.62	0.89	0.74	28
9	0.73	0.79	0.76	34

```

# Authors: Clay Woolam <clay@woolam.org>
# License: BSD

import matplotlib.pyplot as plt
import numpy as np
from scipy import stats

from sklearn import datasets
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.semi_supervised import LabelSpreading

digits = datasets.load_digits()
rng = np.random.RandomState(0)
indices = np.arange(len(digits.data))
rng.shuffle(indices)

X = digits.data[indices[:330]]
y = digits.target[indices[:330]]
images = digits.images[indices[:330]]

n_total_samples = len(y)
n_labeled_points = 40
max_iterations = 5

unlabeled_indices = np.arange(n_total_samples)[n_labeled_points:]
f = plt.figure()

for i in range(max_iterations):
    if len(unlabeled_indices) == 0:
        print("No unlabeled items left to label.")
        break
    y_train = np.copy(y)
    y_train[unlabeled_indices] = -1

    lp_model = LabelSpreading(gamma=0.25, max_iter=20)
    lp_model.fit(X, y_train)

    predicted_labels = lp_model.transduction_[unlabeled_indices]
    true_labels = y[unlabeled_indices]

    cm = confusion_matrix(true_labels, predicted_labels, labels=lp_model.classes_)

    print("Iteration %i %s" % (i, 70 * "_"))
    print(
        "Label Spreading model: %d labeled & %d unlabeled (%d total)"
        % (n_labeled_points, n_total_samples - n_labeled_points, n_total_samples)
    )

    print(classification_report(true_labels, predicted_labels))

    print("Confusion matrix")
    print(cm)

    # compute the entropies of transduced label distributions
    pred_entropies = stats.distributions.entropy(lp_model.label_distributions_.T)

    # select up to 5 digit examples that the classifier is most uncertain about
    uncertainty_index = np.argsort(pred_entropies)[-1]
    uncertainty_index = uncertainty_index[
        np.isin(uncertainty_index, unlabeled_indices)
    ][:5]

    # keep track of indices that we get labels for
    delete_indices = np.array([], dtype=int)

    # for more than 5 iterations, visualize the gain only on the first 5
    if i < 5:
        f.text(
            0.05,
            (1 - (i + 1) * 0.183),
            "model %d\nfit with\n%d labels" % ((i + 1), i * 5 + 10),
            size=10,
        )
        for index, image_index in enumerate(uncertainty_index):
            image = images[image_index]

            # for more than 5 iterations, visualize the gain only on the first 5
            if i < 5:
                sub = f.add_subplot(5, 5, index + 1 + (5 * i))
                sub.imshow(image, cmap=plt.cm.gray_r, interpolation="none")
                sub.set_title(
                    "predict: %i\ntrue: %i"
                    % (lp_model.transduction_[image_index], y[image_index]),
                    size=10,
                )
                sub.axis("off")

            # Labeling 5 points, remote from labeled set
            (delete_index,) = np.where(unlabeled_indices == image_index)
            delete_indices = np.concatenate((delete_indices, delete_index))

        unlabeled_indices = np.delete(unlabeled_indices, delete_indices)
        n_labeled_points += len(uncertainty_index)

f.suptitle(
    (
        "Active learning with Label Propagation.\nRows show 5 most "
        "uncertain labels to learn with the next model."
    ),
    y=1.15,
)
plt.subplots_adjust(left=0.2, bottom=0.03, right=0.9, top=0.9, wspace=0.2, hspace=0.85)
plt.show()

```

El algoritmo utilizado en este código es Label Spreading, que es un método de aprendizaje semi-supervisado. Este enfoque permite utilizar tanto datos etiquetados como no etiquetados para mejorar la precisión de la clasificación.

El algoritmo utilizado en este código es Label Spreading, que es un método de aprendizaje semi-supervisado. Este enfoque permite utilizar tanto datos etiquetados como no etiquetados para mejorar la precisión de la clasificación.

PROPORCIÓN DE ENTRENAMIENTO Y TESTEO

ENTRENAMIENTO: EN ESTE CÓDIGO, SE UTILIZAN 40 PUNTOS ETIQUETADOS (`N_LABELED_POINTS = 40`) DE UN TOTAL DE 330 MUESTRAS.

TESTEO: EL RESTO DE LOS PUNTOS (290) SE CONSIDERAN NO ETIQUETADOS Y SE UTILIZAN PARA EVALUAR EL MODELO. EN CADA ITERACIÓN, SE PROPAGAN LAS ETIQUETAS A LOS PUNTOS NO ETIQUETADOS, Y SE EVALÚA EL RENDIMIENTO DEL MODELO.

EXPLICACIÓN DE LA MATRIZ DE CONFUSIÓN

DESPUÉS DE CADA ITERACIÓN, SE GENERA UNA MATRIZ DE CONFUSIÓN UTILIZANDO LOS VERDADEROS Y LOS PREDICHOES. LA MATRIZ DE CONFUSIÓN SE IMPRIME EN CADA ITERACIÓN Y MUESTRA LOS SIGUIENTES ELEMENTOS:

VERDADEROS POSITIVOS (TP): PREDICCIONES CORRECTAS DONDE EL MODELO IDENTIFICÓ CORRECTAMENTE LA CLASE.

FALSOS POSITIVOS (FP): PREDICCIONES INCORRECTAS DONDE EL MODELO CLASIFICÓ INCORRECTAMENTE UN DATO DE OTRA CLASE COMO PERTENECIENTE A LA CLASE POSITIVA.

FALSOS NEGATIVOS (FN): CASOS DONDE EL MODELO NO LOGRÓ IDENTIFICAR UNA INSTANCIA DE LA CLASE POSITIVA.

VERDADEROS NEGATIVOS (TN): PREDICCIONES CORRECTAS DONDE EL MODELO IDENTIFICÓ CORRECTAMENTE LOS DATOS QUE NO PERTENECEN A LA CLASE POSITIVA.

LA MATRIZ DE CONFUSIÓN PERMITE VISUALIZAR EL RENDIMIENTO DEL MODELO Y CALCULAR MÉTRICAS COMO PRECISIÓN, RECALL Y F1-SCORE.

RESUMEN

ESTE CÓDIGO MUESTRA UN ENFOQUE PRÁCTICO DE APRENDIZAJE SEMI-SUPERVISADO UTILIZANDO EL ALGORITMO DE LABEL SPREADING, DONDE SE ENTRENAN Y EVALÚAN MODELOS ITERATIVAMENTE SOBRE UN CONJUNTO DE DATOS DE DÍGITOS. LA MATRIZ DE CONFUSIÓN ES CRUCIAL PARA ENTENDER CÓMO EL MODELO ESTÁ FUNCIONANDO EN CADA ITERACIÓN Y PARA REALIZAR AJUSTES EN EL PROCESO DE ETIQUETADO ACTIVO.