Konfusionsmatrix

Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Zusammenfassung

Die Konfusionsmatrix einer Hypothese auf einem endlichen Datensatz mit K Label-Werten ist die $K \times K$ Matrix, deren Eintrag in Zeile y und Spalte $\widehat{y}$ die Datenpunkte zählt mit wahrem Label y und prediction $\widehat{y}$. Ihre Diagonale zählt die korrekten predictions, und jeder Eintrag außerhalb der Diagonale zählt eine Art von Fehlklassifikation. Die Genauigkeit, die precision und der recall werden aus der Matrix durch Normierung der Diagonale, einer Spalte oder einer Zeile gewonnen. Bei imbalanced data deckt die Konfusionsmatrix Versagen bei der seltenen Klasse auf, das die Genauigkeit allein verbirgt.

Definition

Die Wetterstation in Krems an der Donau (Österreich) zeichnet stündlich die Lufttemperatur auf. Ein linearer Klassifikator wird mittels logistische Regression aus einem Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ von 40 Datenpunkte gelernt, den Tagen 1.–20. März und 5.–24. Dezember 2024, und seine Entscheidungsgrenze, eine gerade Linie, teilt die Ebene der beiden Messwerte (Abb. 1). Die Merkmale eines Datenpunkt sind die 06:00 (UTC) und 15:00 (UTC) Messwerte eines Tages; sein Label gibt an, ob der 06:00 Messwert des folgenden Tages über 0 °C bleibt oder darunter fällt, eine Frostwarnung.

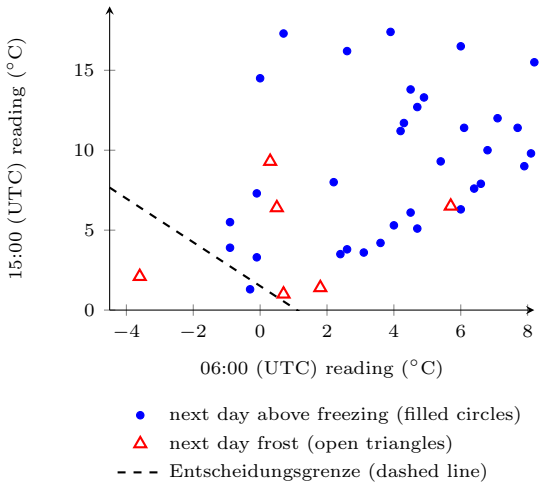
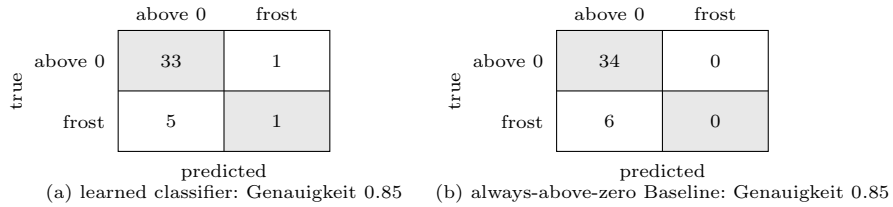


Abbildung 1: Die 40 Datenpunkte der Frostwarnung Krems an der Donau mit der linearen Entscheidungsgrenze, die mittels logistische Regression gelernt wurde. Jeder Datenpunkt ist ein 06:00 (UTC) und 15:00 (UTC) Messwert eines Tages, gekennzeichnet danach, ob der folgende Morgen über 0 °C bleibt (gefüllte Kreise) oder Frost bringt (offene Dreiecke). Daten generiert von `pythondemos/cm.py`

Die meisten Tage bergen kein Risiko: Nur 6 der 40 folgenden Morgen bringen Frost, einer im März und fünf im Dezember. Die konstante prediction „über Null“, eine Baseline, die die Merkmale ignoriert, ist daher an 34 der 40 Tage korrekt, eine Genauigkeit von $34/40 = 0.85$, und der gelernte classifier erreicht die gleiche Genauigkeit von 0.85. Die einzelne Zahl kann die beiden nicht auseinanderhalten, noch aufdecken, welche Fehler einer von beiden macht. Ein classifier mit dieser Genauigkeit könnte die Frosttage übersehen oder Fehlalarme auslösen. Die Konfusionsmatrix schlüsselt die Anzahl der predictions danach auf, was wahr war und was vorhergesagt wurde (Abb. 2). Der gelernte classifier erkennt einen der sechs Frosttage auf Kosten eines Fehlalarms, während die Baseline keinen erkennt und keinen auslöst.



(a) learned classifier: Genauigkeit 0.85 (b) always-above-zero Baseline: Genauigkeit 0.85

Abbildung 2: Die Konfusionsmatrizen des gelernten classifier (a) und der immer-über-Null-Baseline (b) aus Abb. 1. Beide erreichen eine Genauigkeit von 0.85; nur die Konfusionsmatrizen zeigen, dass der gelernte classifier einen Frosttag erkennt und einen Fehlalarm auslöst, während die Baseline keinen erkennt und keinen auslöst. Die schattierten Diagonalzellen enthalten die korrekten predictions. Daten generiert von `pythondemos/cm.py`

Betrachten wir ein endliches Datensatz mit m Datenpunkte, jeder charakterisiert durch einen Merkmalsvektor $\mathbf{x}$ und ein Label y aus einem endlichen label space $\mathcal{Y} = \{1, \ldots, K\}$. Für eine gegebene Hypothese h ist die Konfusionsmatrix eine $K \times K$ Matrix mit einer Zeile pro wahrem Label-Wert und einer Spalte pro vorhergesagtem Wert (Hastie et al., 2009, Abschn. 9.1.2). Der Eintrag in Zeile y und Spalte $\widehat{y}$ ist die Anzahl der Datenpunkte mit wahrem Label y und prediction $h(\mathbf{x}) = \widehat{y}$. Ein Diagonaleintrag ist einer mit $\widehat{y} = y$, bei dem die prediction mit dem Label übereinstimmt, sodass die Diagonale die korrekt klassifizierten Datenpunkte zählt und die Einträge außerhalb der Diagonale jede Art von Fehlklassifikation separat zählen.

Die standardmäßigen Klassifikation-metrics sind Verhältnisse von Einträgen der Konfusionsmatrix. Die Genauigkeit ist die Summe der Diagonale geteilt durch m. Die Normierung einer Zeile durch ihre Summe ergibt den Anteil der Datenpunkte dieser Klasse, die erkannt werden (für die positive Klasse, der recall), während die Normierung einer Spalte den Anteil der predictions dieser Klasse ergibt, die richtig sind, die precision. Für den gelernten classifier aus Abb. 2, mit Frost als der positiven Klasse, beträgt der recall 1/6 und die precision 1/2. Die Baseline hat einen recall von 0, und ihre precision ist überhaupt nicht definiert, da sie niemals Frost vorhersagt (`pythondemos/cm.py`). Bei imbalanced data deckt die Konfusionsmatrix auf, was eine hohe Genauigkeit verbergen kann. Ein classifier, der niemals die seltene Klasse vorhersagt, füllt die Zeile dieser Klasse mit Fehlklassifikationen, während die meisten predictions korrekt bleiben. Die immer-über-Null-Baseline in Abb. 2 tut genau das (siehe Baseline).

Alle oben genannten Zahlen werden auf dem Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ berechnet, den 40 Tagen, aus denen der classifier gelernt wurde. Eine Frostwarnung wird an Tagen beurteilt, die nicht zu $\mathcal{D}^{(\text{train})}$ gehörten, daher muss die Konfusionsmatrix und jedes daraus abgeleitete Verhältnis erneut auf einem Validierungsdatensatz berechnet werden, bevor ein classifier dem anderen vorgezogen wird.

Siehe auch: Klassifikation, Genauigkeit, precision, recall, imbalanced data,

Baseline, Trainingsdatensatz, Validierungsdatensatz, metric, Label, label space, Matrix.

Querverweise

Literatur

1. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.