Genauigkeit

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

Genauigkeit ist der Anteil korrekter predictions, die von einer Hypothese auf einem Datensatz mit einem endlichen label space gemacht werden, gleich eins minus dem durchschnittlichen 0/1 loss. Sie wird häufig als einstellige metric für Klassifikation-Methoden verwendet. Genauigkeit ist nicht als objective function für das Training geeignet, da der 0/1 loss non-smooth und nicht konvex ist. Empirische Risikominimierung (ERM) verwendet stattdessen einen Ersatz-Verlust, eine konvexe oder smoothe obere Schranke für den 0/1 loss, und die Genauigkeit wird auf einem Validierungsdatensatz oder einem Testdatensatz angegeben. Bei imbalanced data kann die Genauigkeit allein einen classifier nicht von einer Baseline unterscheiden, die die Merkmale ignoriert, was andere Bewertungsmethoden wie die Konfusionsmatrix erfordert.

Definition

Eine Frostwarnung für Krems an der Donau ist an manchen Tagen richtig und an anderen falsch, und der Anteil der Tage, an denen sie richtig ist, ist ihre Genauigkeit. Im Allgemeinen ist die Genauigkeit der Anteil korrekter predictions, die von einer Hypothese $h : \mathcal{X} \to \mathcal{Y}$ auf einem Datensatz $\mathcal{D} = \big\{\big(\mathbf{x}^{(r)}, y^{(r)}\big)\big\}_{r=1}^{m}$ von Merkmalsvektoren $\mathbf{x}^{(r)}$ und Labels $y^{(r)}$ gemacht werden, wobei jedes Label ein Wert $y \in \mathcal{Y}$ aus einem endlichen label space $\mathcal{Y}$ ist (Goodfellow et al., 2016, Sect. 5.1.2; Shalev-Shwartz and Ben-David, 2014, Sect. 2.2). Mit dem 0/1 loss $L^{(0/1)}(\mathbf{z}, h)$, der 1 für eine falsche prediction auf dem Datenpunkt $\mathbf{z}$ und 0 sonst ist, ist die Genauigkeit

$$\begin{aligned}\mathrm{acc}(h \mid \mathcal{D}) &:= 1 - \frac{1}{m}\sum_{r=1}^{m} L^{(0/1)}\Big(\big(\mathbf{x}^{(r)}, y^{(r)}\big), h\Big) \\ &= \frac{1}{m}\Big|\Big\{r \in \{1, \ldots, m\} : h\big(\mathbf{x}^{(r)}\big) = y^{(r)}\Big\}\Big|.\end{aligned}$$

Die Genauigkeit reicht von 0 (keine prediction ist korrekt) bis 1 (jede prediction ist korrekt). Äquivalent dazu entspricht die Genauigkeit eins minus dem durchschnittlichen 0/1 loss auf dem Datensatz. Zum Beispiel weist bei der Bild-Klassifikation mit dem label space $\mathcal{Y} = \{\text{Katze}, \text{Hund}, \text{Vogel}\}$, die Hypothese jedem Bild, dargestellt durch seinen Merkmalsvektor $\mathbf{x}$, eine Kategorie $h(\mathbf{x})$ zu.

Die Genauigkeit ist dann der Anteil der Bilder, deren zugewiesene Kategorie dem wahren Label y entspricht: 94 korrekte Zuweisungen unter 100 Bildern ergeben eine Genauigkeit von 0.94.

Abb. 1 vergleicht zwei classifiers auf demselben Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ von 60 Datenpunkte, dem Datensatz, aus dem beide gelernt wurden. Der linearer Klassifikator h, dessen Entscheidungsgrenze eine Linie ist, klassifiziert drei Datenpunkte in der Nähe dieser Linie falsch. Der nichtlineare classifier h' krümmt sich um jeden dieser Datenpunkte und klassifiziert sie korrekt.

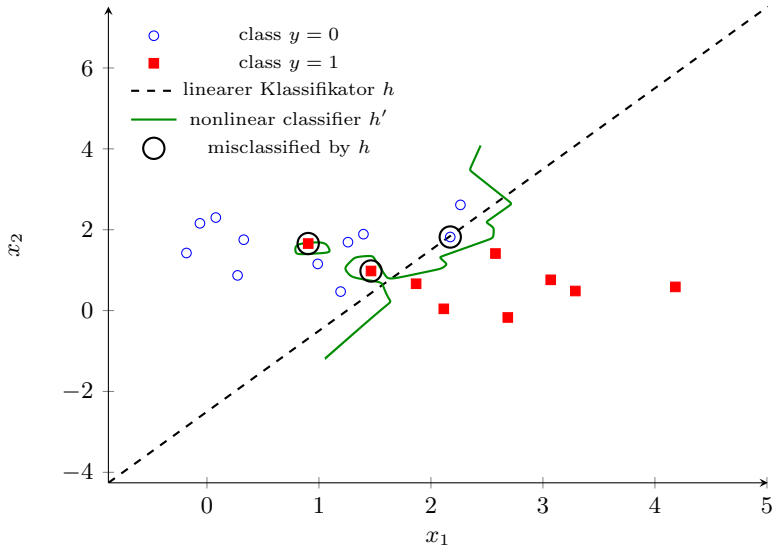


Abbildung 1: Zwei classifiers, angewendet auf dasselbe Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ von $m = 60$ Datenpunkte mit Klassen $y \in \{0, 1\}$ (blaue Kreise und rote Quadrate); 20 davon sind gezeichnet, jeder dritte der korrekt klassifizierten und die drei falsch klassifizierten. Der linearer Klassifikator h (gestrichelte Linie) platziert drei Datenpunkte auf der falschen Seite seiner Entscheidungsgrenze; sie sind schwarz umrandet und liegen nahe an der gestrichelten Linie, und sie ergeben $\text{acc}(h \mid \mathcal{D}^{(\text{train})}) = 57/60 = 0.95$. Der nichtlineare classifier h' (durchgezogene grüne Kurve) krümmt sich um diese drei Datenpunkte und ergibt $\text{acc}(h' \mid \mathcal{D}^{(\text{train})}) = 1$. Daten generiert von `pythondemos/accuracy.py`

Der classifier h' in Abb. 1 erreicht die optimale Genauigkeit von 1 auf $\mathcal{D}^{(\text{train})}$: seine Entscheidungsgrenze folgt eng den Positionen der einzelnen Datenpunkte, die der linearer Klassifikator falsch klassifiziert. Der classifier h' ist daher empfindlich gegenüber kleinen Störungen der Merkmalsvektoren, wie z.B. Messrauschen. Eine leichte Verschiebung des Merkmalsvektor eines Datenpunkt in der Nähe der Entscheidungsgrenze kann die prediction umkehren. Die Optimierung der Genauigkeit auf dem Trainingsdatensatz allein gewährleistet daher keine gute Generalisierung (Shalev-Shwartz and Ben-David, 2014, Sect. 2.2.1). Gute Generalisierung bedeutet einen kleinen Verlust auf Datenpunkte außerhalb des

Trainingsdatensatz. Genauigkeit wird oft als einstellige metric verwendet, um gelernte classifiers zu vergleichen: auf einem Validierungsdatensatz $\mathcal{D}^{(\text{val})}$, um unter Kandidaten-Hypothesen während der validation auszuwählen, und auf einem Testdatensatz $\mathcal{D}^{(\text{test})}$, dessen Datenpunkte weder in das Training noch in die Auswahl eingingen, um die ausgewählte Hypothese zu bewerten.

Genauigkeit könnte auch als objective function für das Training verwendet werden. Allerdings ist die Genauigkeit über den 0/1 loss definiert, der als function der model parameters stückweise konstant und weder smooth noch konvex ist (Shalev-Shwartz and Ben-David, 2014, Sect. 12.3), sodass auch die Genauigkeit stückweise konstant ist: Der Gradient der Genauigkeit ist überall dort, wo er existiert, null und zeigt daher keine Verbesserungsrichtung an. Gradientbasierte optimization methods können die Genauigkeit nicht direkt maximieren. Stattdessen verwenden empirische Risikominimierung (ERM)-basierte Methoden einen Ersatz-Verlust, der den 0/1 loss nach oben begrenzt und konvex oder smooth gewählt wird, damit diese Methoden einem Gradient folgen können (Shalev-Shwartz and Ben-David, 2014, Sect. 12.3). Zwei Beispiele für einen solchen Ersatz-Verlust sind der logistische Verlustfunktion, der konvex (Shalev-Shwartz and Ben-David, 2014, Sect. 9.3) und smooth (Shalev-Shwartz and Ben-David, 2014, Example 12.6) ist, und der hinge loss, der konvex (Shalev-Shwartz and Ben-David, 2014, Sect. 12.3) aber non-smooth ist: seine einseitigen derivatives bei margin 1 sind -1 und 0, sodass er dort nicht differentiable ist.

Abb. 2 zeigt die drei loss functions als functions des margin für die binäre Klassifikation mit $\mathcal{Y} = \{-1, 1\}$, wobei eine reellwertige Hypothese h über das Vorzeichen von $h(\mathbf{x})$ vorhersagt. Jede loss function hängt vom Datenpunkt nur über den margin $y \cdot h(\mathbf{x})$ ab, wodurch loss functions für die binäre Klassifikation verglichen werden (Hastie et al., 2009, Sect. 10.6). Der hinge loss liegt überall über dem 0/1 loss (Shalev-Shwartz and Ben-David, 2014, Sect. 12.3); der logistische Verlustfunktion ist $\ln 2$ bei margin 0, sodass es der um $1/\ln 2$ skalierte logistische Verlustfunktion ist, der über dem 0/1 loss liegt.

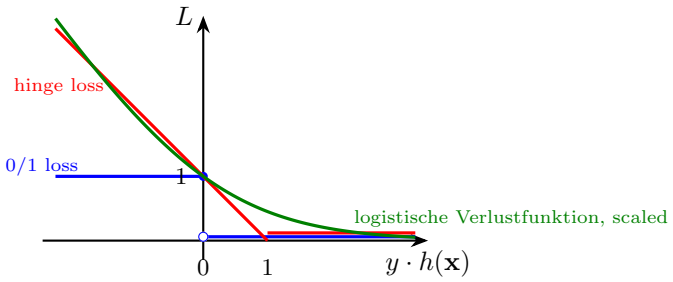
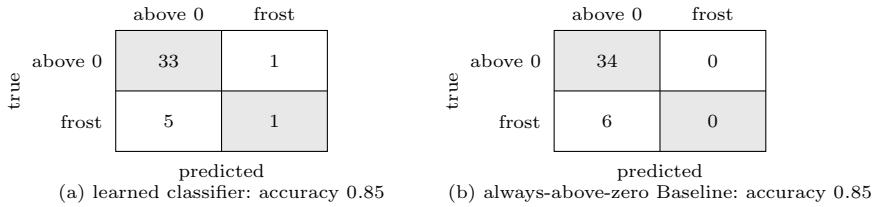


Abbildung 2: Drei loss functions für die binäre Klassifikation als functions des margin $y \cdot h(\mathbf{x})$: der 0/1 loss (blauer Schritt; der gefüllte Punkt markiert seinen Wert 1 bei margin 0, der offene Punkt den Wert, den er nicht annimmt), der hinge loss (rote Kurve) und der um $1/\ln 2$ skalierte logistische Verlustfunktion (grüne Kurve). Alle drei gehen durch 1 bei margin 0. Beide Surrogate sind konvex, liegen über dem 0/1 loss und bestrafen kleine margins, wo der 0/1 loss bereits 0 ist; die beiden Nullzweige sind leicht über der Achse gezeichnet, damit sie sichtbar bleiben

Genauigkeit ist die metric der Wahl, wenn die Klassen ausgeglichen sind und jeder Fehler gleich viel kostet; bei imbalanced data ist sie es nicht. Betrachten Sie eine Frostwarnung für Krems an der Donau (Österreich): Die Merkmale eines Datenpunkt sind die minimale und die Lufttemperatur um 06:00 und um 15:00 (UTC) eines Tages, und sein Label gibt an, ob der Messwert um 06:00 des folgenden Tages unter 0 °C fällt. Frosttage sind die Minderheitsklasse: Von einem Trainingsdatensatz von 40 aufgezeichneten Tagen folgen nur 6 Frost.

Die konstante prediction „über Null“ (eine Baseline, die die Merkmale ignoriert) ist daher an 34 der 40 Tage korrekt, eine Genauigkeit von 0.85, und ein classifier, der durch logistische Regression aus den 40 Datenpunkte gelernt wurde, erreicht dieselbe Genauigkeit von 0.85. Beide machen die gleiche Anzahl von Fehlern, sodass die Genauigkeit die beiden nicht unterscheiden kann; sie unterscheiden sich darin, wo sie richtig liegen, was andere Bewertungsmethoden wie die Konfusionsmatrix erfordert. Die Konfusionsmatrizen in Abb. 3 zeigen, dass der gelernte classifier einen der sechs Frosttage erkennt und einen Fehlalarm auslöst, während die Baseline keinen erkennt und keinen auslöst.



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

Abbildung 3: Die Konfusionsmatrizen des Frostwarnungs-classifier, gelernt durch logistische Regression (a), und der Immer-über-Null-Baseline (b). Die schattierten diagonalen Zellen enthalten die korrekten predictions: $33 + 1 = 34$ für den gelernten classifier und $34+0 = 34$ für die Baseline, sodass beide dieselbe Genauigkeit von 0.85 mit jeweils 6 Fehlern erreichen. Die Zellen unterscheiden sich darin, wo die korrekten und die falschen predictions liegen: Der gelernte classifier erkennt einen Frosttag und löst einen Fehlalarm aus, während die Baseline keinen erkennt und keinen auslöst. Daten generiert von `pythondemos/cm.py`

Siehe auch: 0/1 loss, Verlust, Klassifikation, F_1 score, Konfusionsmatrix, imbalanced data.

Querverweise

Literatur

1. Goodfellow et al. (2016). Deep Learning. MIT Press, Cambridge, MA, USA.

2. Shalev-Shwartz and Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge Univ. Press, New York, NY, USA.

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