empirische Risikominimierung (ERM)

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 empirische Risikominimierung (ERM) wählt eine Hypothese aus, indem sie den durchschnittlichen Verlust auf einem gegebenen Trainingsdatensatz $\mathcal{D}^{(\text{train})}$, das empirical risk, minimiert. Machine learning (ML)-Methoden zielen auf eine Hypothese ab, die einen geringen Verlust für jeden Datenpunkt verursacht, und ERM ersetzt diese unbekannte Größe durch ihren Durchschnitt über das Trainingsdatensatz. Verschiedene ML-Methoden ergeben sich aus unterschiedlichen Wahlen des Modell $\mathcal{H}$ und der Verlustfunktion L, die beide vom Merkmalsraum $\mathcal{X}$ und dem Labelraum $\mathcal{Y}$ einer gegebenen Methode abhängen.

Definition

Eine Wettervorhersage, die die morgige Temperatur um 15:00 (UTC) vorhersagt, ist nur dann nützlich, wenn ihre Vorhersagen an der großen Mehrheit der Tage genau sind, nicht nur an den Tagen, an denen sie trainiert wurde. Diese Vorhersage ist eine Hypothese: Sie bildet die Merkmale eines Tages auf eine Vorhersage der morgigen Temperatur ab, und ihr Verlust quantifiziert, wie weit diese Vorhersage vom gemessenen Wert abweicht. Allgemeiner zielen machine learning (ML)-Methoden darauf ab, eine Hypothese zu finden, die einen geringen Verlust für jeden Datenpunkt verursacht. Der Ausdruck *jeder Datenpunkt* wird durch ein Wahrscheinlichkeitsmodell für die Generierung von Datenpunkte präzisiert, wie folgt.

Wenn Datenpunkte als aus einer Wahrscheinlichkeitsverteilung gezogen angenommen werden, dann ist das Risiko einer Hypothese als der Erwartungswert ihres Verlust unter dieser Wahrscheinlichkeitsverteilung definiert. Die ideale Wahl ist dann die Hypothese mit minimalem Risiko. In den meisten ML-Anwendungen ist jedoch die zugrunde liegende Wahrscheinlichkeitsverteilung nicht bekannt. Stattdessen ist nur ein endliches Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ von Datenpunkte verfügbar. ERM ersetzt das unbekannte Risiko durch seinen Stichprobe-durchschnittlichen Ersatz, das empirical risk auf $\mathcal{D}^{(\text{train})}$, und wählt die Hypothese, die diesen Ersatz minimiert (Hastie et al., 2009; Murphy, 2012; Shalev-Shwartz and Ben-David, 2014; Vapnik, 2000).

Für eine feste Wahl von Verlustfunktion und Hypothesenraum ist ERM eine Abbildung $\mathcal{A}$, die ein Trainingsdatensatz $\mathcal{D}^{(\text{train})} = \{\mathbf{z}^{(r)}\}_{r=1}^{m}$ von Datenpunkte $\mathbf{z}^{(r)} = \left(\mathbf{x}^{(r)}, y^{(r)}\right)$ einliest. Die Abbildung gibt eine gelernte Hypothese $\widehat{h} =$

$\mathcal{A}(\mathcal{D}^{(\text{train})}) \in \mathcal{H}$ zurück, die das empirical risk auf $\mathcal{D}^{(\text{train})}$ minimiert,

$$\widehat{h} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{m} \sum_{r=1}^{m} L\left(\mathbf{z}^{(r)}, h\right). \tag{1}$$

Die Hypothese $\widehat{h}$ in (1) wird aus dem zugrunde liegenden Hypothesenraum (oder Modell) $\mathcal{H}$ gewählt. Für eine ERM-basierte Methode besteht das Modell Training darin, $\mathcal{A}$ zu berechnen.

Verschiedene ML-Methoden ergeben sich aus unterschiedlichen Entwurfsentscheidungen für die beiden Komponenten der Abbildung $\mathcal{A}$: das Modell $\mathcal{H}$ und die Verlustfunktion L. Beide hängen von der Natur der Datenpunkte ab, d.h. vom Merkmalsraum $\mathcal{X}$ und dem Labelraum $\mathcal{Y}$. Zum Beispiel werden Bild-Datenpunkte mit einem hochdimensionalen Merkmalsraum und einem endlichen Labelraum typischerweise von einem künstliches neuronales Netz (KNN) zusammen mit dem logistische Verlustfunktion behandelt (Goodfellow et al., 2016, Kap. 9). Umgekehrt erlauben Datenpunkte mit wenigen reellwertigen Merkmale und einem reellwertigen Label oft ein lineares Modell zusammen mit dem squared error loss (Hastie et al., 2009, Kap. 3).

Abb. 1 illustriert ERM für ein lineares Modell auf Datenpunkte mit einem einzelnen Merkmal x und Label y. Jede Hypothese ist eine lineare Abbildung $h(x) = w_1 x + w_0$ mit Modellparameter w_0, w_1. ERM wählt die Modellparameter, die das empirical risk auf $\mathcal{D}^{(\text{train})}$ minimieren.

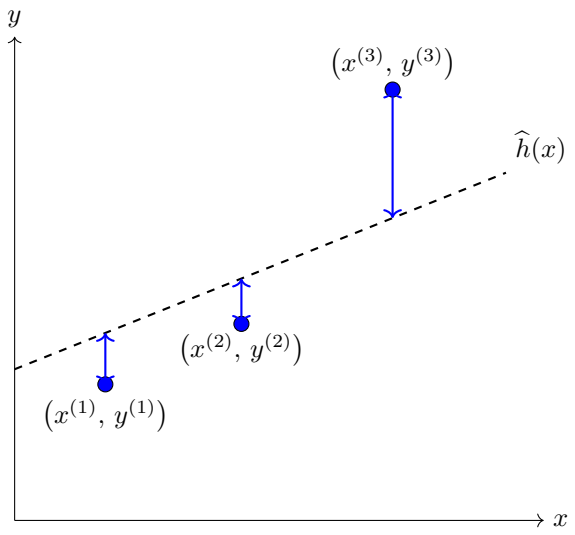

Abbildung 1: ERM lernt eine Hypothese $\widehat{h} \in \mathcal{H}$ aus einem Modell $\mathcal{H}$, indem es das empirical risk (oder den durchschnittlichen Verlust) $\frac{1}{m} \sum_{r=1}^{m} L\left(\mathbf{z}^{(r)}, h\right)$ über Kandidaten-Hypothesen $h \in \mathcal{H}$ auf einem Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ minimiert. Die Pfeile zeigen die Vorhersage-Fehler an, die durch die gelernte Hypothese $\widehat{h}$ (gestrichelte Linie) für jeden Datenpunkt im Trainingsdatensatz (gefüllte Kreise) verursacht werden; die Achsen tragen das Merkmal x und das Label y.

In der Praxis wird die Abbildung $\mathcal{A}$ durch eine iterative Optimierungsverfahren berechnet. Viele dieser Optimierungsverfahren können als fixed-point iteration dargestellt werden, die mit einer anfänglichen Hypothese $h^{(0)}$ beginnt und dann wiederholt einen Update-Operator $\mathcal{F}$ anwendet (Bauschke and Combettes, 2011, Abschn. 5.2, Kor. 27.9),

$$h^{(t+1)} = \mathcal{F}\left(h^{(t)}\right), \text{ für } t = 0, 1, \ldots.$$

Der Update-Operator $\mathcal{F}$ hängt vom Trainingsdatensatz, der Verlustfunktion und dem Hypothesenraum ab. Er wird so gewählt, dass seine Fixpunkte die stationären Punkte des empirical risk sind. Die stationären Punkte sind ihre Minimierer, wenn das empirical risk eine konvex Funktion der Hypothese ist (Boyd and Vandenberghe, 2004, Abschn. 4.2.3), und müssen es sonst nicht sein, z.B. für ein KNN.

Für ERM, das ein parametrisches Modell $\mathcal{H}$ verwendet, kann die fixed-point iteration direkt in Bezug auf die Modellparameter $\mathbf{w}$ anstelle der Hypothese h formuliert werden. Insbesondere, beginnend mit anfänglichen Modellparameter $\mathbf{w}^{(0)}$, wird der Update-Operator $\mathcal{F}$ wiederholt auf die Modellparameter angewendet,

$$\mathbf{w}^{(t+1)} = \mathcal{F}\left(\mathbf{w}^{(t)}\right), \text{ für } t = 0, 1, \ldots.$$

Ein Beispiel für eine solche fixed-point iteration ist Gradientenabstieg, die die Modellparameter in Richtung des negativen Gradient des empirical risk aktualisiert (das differentiable sein muss).

Zwei Anforderungen grenzen den Anwendungsbereich von ERM ab: vollständiges Feedback zum Verlust und gleichzeitiger Zugriff auf das gesamte Trainingsdatensatz. Erstens setzt ERM voraus, dass der Verlust $L\left(\mathbf{z}^{(r)}, h\right)$ für jede Hypothese $h \in \mathcal{H}$ und jeden Training Datenpunkt $\mathbf{z}^{(r)}$ ausgewertet werden kann. Diese Anforderung an vollständiges Feedback unterscheidet ERM von bestärkendes Lernen (RL)-Algorithmen, die partielles Feedback verwenden. Insbesondere wählt ein RL-Algorithmus zu jedem Zeitpunkt $t = 1, 2, \ldots$ eine Aktion, die Vorhersage seiner aktuellen Hypothese $\widehat{h}^{(t)}$, und beobachtet nur den Verlust, der durch diese Aktion verursacht wird. Es hat keine Informationen über den Verlust, der von einer anderen Hypothese $h \in \mathcal{H} \setminus \{\widehat{h}^{(t)}\}$ verursacht worden wäre.

Zweitens verwendet ERM das gesamte Trainingsdatensatz auf einmal. Diese Anforderung kann gelockert werden: ERM kann auch als online algorithm implementiert werden, was nützlich ist, wenn die Datenpunkte in $\mathcal{D}^{(\text{train})}$ nur sequenziell zugänglich sind. Ein solcher sequenzieller Zugriff entsteht beispielsweise, wenn Speicherbeschränkungen das gleichzeitige Laden des gesamten Trainingsdatensatz verhindern. Die kanonische Online-Form von ERM ist der Follow-The-Leader (FTL)-Algorithmus, der in jeder Runde ein partielles ERM-Problem auf allen bisher gesehenen Datenpunkte löst. Angenommen, der Datenpunkt $\mathbf{z}^{(t)}$ trifft zum Zeitpunkt $t = 1, 2, \ldots$ ein, FTL aktualisiert die gelernte Hypothese $\widehat{h}^{(t)}$ durch

$$\widehat{h}^{(t)} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{t} \sum_{r=1}^{t} L\left(\mathbf{z}^{(r)}, h\right).$$

Die separate Implementierung dieses partiellen ERM für jeden Zeitpunkt kann rechenintensiv (und verschwenderisch) werden, da sie die Ähnlichkeit zwischen aufeinanderfolgenden partiellen ERM-Problemen nicht ausnutzt. Diese Ähnlichkeit kann durch online gradient descent (online GD) ausgenutzt werden, das zu jedem Zeitpunkt t einen einzelnen Gradientenabstieg-Schritt auf dem neuen Datenpunkt $\mathbf{z}^{(t)}$ durchführt (Hazan, 2022, Abschn. 3.1).

Synonyme: empirische Verlustminimierung.

Siehe auch: empirical risk, Risiko, Verlust, Trainingsdatensatz, Hypothesenraum, Hypothese, Gradientenabstieg, FTL, RL.

Querverweise

Literatur

1. Bauschke and Combettes (2011). Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer Science+Business Media, New York, NY, USA.

2. Boyd and Vandenberghe (2004). Convex Optimization. Cambridge Univ. Press, Cambridge, U.K.

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

4. Hazan (2022). Introduction to Online Convex Optimization. 2nd ed., MIT Press, Cambridge, MA, USA.

5. Murphy (2012). Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge, MA, USA.

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

7. Vapnik (2000). The Nature of Statistical Learning Theory. 2nd ed., Springer, New York, NY, USA.

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