empirical risk minimization (ERM)

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Synonyms

empirical loss minimization

Abstract

Machine learning (ML) methods aim to find a hypothesis that yields accurate predictions, reflected by a small loss. Empirical risk minimization (ERM) formalizes this: it selects a hypothesis with minimal empirical risk, i.e., minimal average loss on a training set $\mathcal{D}^{(\text{train})}$. Different ML methods arise from different choices of the model $\mathcal{H}$ and the loss function L, both of which depend on the feature space $\mathcal{X}$ and the label space $\mathcal{Y}$ of a given method. The entry also covers the online form of ERM (Follow-The-Leader (FTL)) and contrasts ERM with reinforcement learning (RL) methods that use a different access mechanism for the loss values.

Definition

A weather forecaster predicting tomorrow's maximum daytime temperature is useful only if its predictions are accurate on most days. This forecaster is a hypothesis: it maps a day's features to a prediction of tomorrow's temperature, and its loss quantifies how far that prediction lies from the measured value. More generally, machine learning (ML) methods aim to find a hypothesis that incurs a small loss for any data point. But what does *any* data point mean? One way to make this notion precise is to use a probabilistic model for the generation of data points.

If data points are assumed to be drawn from a probability distribution, then the risk of a hypothesis is defined as the expectation of its loss under that probability distribution. The ideal choice is then the hypothesis with minimal risk. However, in most ML applications, the underlying probability distribution is not known. Instead, only a finite training set $\mathcal{D}^{(\text{train})}$ of data points is available. ERM replaces the intractable risk with its sample-average surrogate, the empirical risk on $\mathcal{D}^{(\text{train})}$, and picks the hypothesis that minimizes this surrogate (Hastie et al., 2009; Murphy, 2012; Shalev-Shwartz and Ben-David, 2014; Vapnik, 2000).

For a fixed choice of loss function and hypothesis space, ERM is a map $\mathcal{A}$ that reads in a training set $\mathcal{D}^{(\text{train})} = \{\mathbf{z}^{(r)}\}_{r=1}^{m}$ of data points $\mathbf{z}^{(r)} = \big(\mathbf{x}^{(r)}, y^{(r)}\big)$ and

returns a learned hypothesis $\hat{h} = \mathcal{A}(\mathcal{D}^{(\text{train})}) \in \mathcal{H}$ that minimizes the empirical risk on $\mathcal{D}^{(\text{train})}$,

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

The hypothesis $\hat{h}$ is chosen from the underlying hypothesis space (or model) $\mathcal{H}$. For an ERM-based method, model training is to compute $\mathcal{A}$.

In practice, the map $\mathcal{A}$ is computed by an iterative optimization method. Many of these optimization methods can be represented as a fixed-point iteration that starts with an initial hypothesis $h^{(0)}$ and then repeatedly applies an update operator $\mathcal{F}$,

$$h^{(t+1)} = \mathcal{F}\big(h^{(t)}\big), \text{ for } t = 0, 1, \ldots.$$

The update operator $\mathcal{F}$ depends on the training set, the loss function, and the hypothesis space. It is chosen such that its fixed points are minimizers of the empirical risk.

For ERM using a parametric model $\mathcal{H}$, the fixed-point iteration can be formulated directly in terms of the model parameters $\mathbf{w}$ instead of the hypothesis h. In particular, starting with initial model parameters $\mathbf{w}^{(0)}$, the update operator $\mathcal{F}$ is applied repeatedly to the model parameters,

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

One important example of such a fixed-point iteration is gradient descent (GD), which updates the model parameters in the direction of the negative gradient of the empirical risk (which needs to be differentiable).

ERM presupposes that the loss $L\left(\mathbf{z}^{(r)}, h\right)$ can be evaluated for every hypothesis $h \in \mathcal{H}$ and every training data point $\mathbf{z}^{(r)}$. This full-feedback requirement is what distinguishes ERM from reinforcement learning (RL) algorithms, which use partial feedback. In particular, at each time instant t, an RL algorithm chooses a hypothesis $\hat{h}^{(t)}$ delivering a prediction (or action) and only measures the loss incurred by this hypothesis. It has no information about the loss that would have been incurred by any other hypothesis $h \in \mathcal{H} \setminus \{\hat{h}^{(t)}\}$.

ERM can also be implemented as an online algorithm, which is useful when the data points in $\mathcal{D}^{(\text{train})}$ can only be accessed sequentially. Such sequential access arises, for instance, when memory limits prevent loading the entire training set at once. The canonical online form of ERM is the Follow-The-Leader (FTL) algorithm, which, at every round, solves a partial ERM problem on all data points seen so far. Assuming that the data point $\mathbf{z}^{(t)}$ arrives at time instant $t = 1, 2, \ldots$, FTL updates the learned hypothesis $\hat{h}^{(t)}$ by

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

Implementing this partial ERM separately for each time instant can become computationally expensive (and wasteful), as it does not exploit the similarity between consecutive partial ERM problems. This similarity can be exploited by

online gradient descent (online GD), which performs a single GD step on the new data point $\mathbf{z}^{(t)}$ at every time instant t (Hazan, 2022, Sect. 3.1).

Different ML methods arise from different design choices for the two ingredients of the map $\mathcal{A}$: the model $\mathcal{H}$ and the loss function L (Jung, 2022, Ch. 3). Both depend on the nature of the data points, i.e., on the feature space $\mathcal{X}$ and the label space $\mathcal{Y}$. For example, image data points with a high-dimensional feature space and a finite label space typically call for an artificial neural network (ANN) together with the logistic loss. Conversely, data points with few real-valued features and a real-valued label often allow for a linear model together with the squared error loss.

Fig. 1 illustrates ERM for a linear model on data points with a single feature x and label y. Each hypothesis is a linear map $h(x) = w_1 x + w_0$ with model parameters w_0, w_1. ERM picks the model parameters that minimize the empirical risk on $\mathcal{D}^{(\text{train})}$.

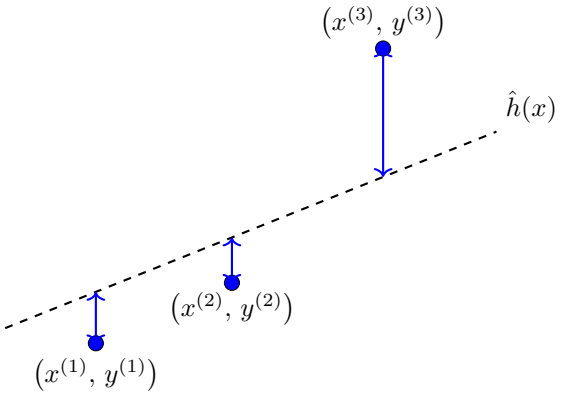

Figure 1: ERM learns a hypothesis $\hat{h} \in \mathcal{H}$ out of a model $\mathcal{H}$ by minimizing the empirical risk (or average loss) $\frac{1}{m}\sum_{r=1}^{m} L\left(\mathbf{z}^{(r)}, h\right)$ over candidate hypotheses $h \in \mathcal{H}$ on a training set $\mathcal{D}^{(\text{train})}$. The arrows indicate the prediction errors incurred by the learned hypothesis $\hat{h}$ (dashed line) for each data point in the training set (filled circles)

Cross References

optimization problem, loss, loss function, empirical risk, risk, training set, hypothesis space, model, hypothesis, algorithm, training, generalization, optimization method, GD, online algorithm, online GD, FTL, RL

References

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

2. Jung (2022). Machine Learning: The Basics. Springer Nature, Singapore, Singapore.

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

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

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

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