PAC-Bayes Generalization Bounds: from theory to experiment

Pierre Bayle
Princeton University

PRINCETON UNIVERSITY

Problem

In statistical machine learning, we want the generalization gap - i.e. the difference between the generalization error and the empirical error - to be as small as possible.

We consider a binary classification problem. Let $\mathcal{X}$ denote a set, D any distribution on $\mathcal{X}$, c the actual classifier we want to learn, $S = ((\mathbf{x}_i, c(\mathbf{x}_i)))_{i=1}^m$ the training set, where each $\mathbf{x}_i$ is chosen independently from D, $\mathcal{H}$ a set of hypotheses.

For $h \in \mathcal{H}$, we define the generalization error and the empirical error

$$err_D(h) = Pr_{\mathbf{x}\sim D}(c(\mathbf{x}) \neq h(\mathbf{x}))$$

$$\hat{err}(h) = \frac{1}{m}\sum_{i=1}^{m} I[c(\mathbf{x}_i) \neq h(\mathbf{x}_i)]$$

Usual bounds

With probability $1-\delta$ over the draw of m training samples, the following holds for all h in $\mathcal{H}$:

- If $\mathcal{H}$ finite

$$err_D(h) \leq \hat{err}(h) + \sqrt{\frac{\ln(2|\mathcal{H}|) + \ln\frac{1}{\delta}}{2m}} \quad (1)$$

- If $\mathcal{H}$ infinite with finite VC-dimension d

$$err_D(h) \leq \hat{err}(h) + \sqrt{\frac{2\ln\frac{2}{\delta}}{m}} + \sqrt{\frac{2d\ln\frac{em}{d}}{m}} \quad (2)$$

Note: the VC-dimension of the class of linear threshold functions in $\mathbb{R}^n$ is $n+1$.

PAC-Bayes framework

The PAC-Bayes analysis involves a hypothesis space $\mathcal{H}$ and a prior distribution P over $\mathcal{H}$ to get a posterior Q over $\mathcal{H}$. The distribution P has to be chosen before learning, however the bounds hold for all Q, thus Q is not the classical Bayesian posterior.

The average theoretical error $err_D(Q)$ can be seen as the theoretical error of a meta-classifier that chooses a classifier randomly from the distribution Q.

$$err_D(Q) = \mathbb{E}_{h\sim Q}\left[err_D(h)\right]$$

The average experimental error $\hat{err}(Q)$ can be seen as the experimental error of the meta-classifier.

$$\hat{err}(Q) = \mathbb{E}_{h\sim Q}\left[\hat{err}(h)\right]$$

Bound in this random setting

With probability $1-\delta$ over the draw of m training samples, the following holds for all h in $\mathcal{H}$:

$$err_D(Q) \leq \hat{err}(Q) + \sqrt{\frac{RE(Q||P) + \ln\frac{m}{\delta}}{2(m-1)}} \quad (3)$$

where $RE(Q||P) = \mathbb{E}_{c\sim Q}\left[\ln\frac{Q(c)}{P(c)}\right]$ is the relative entropy (also known as Kullback-Leibler divergence).

To minimize the upper bound, the posterior distribution Q should have a small empirical error $\hat{err}(Q)$ and at the same time should be close to the prior (so that $RE(Q||P)$ is minimized). Indeed, having a good prior is key because the closer our posterior is to our prior, the smaller the relative entropy and thus the smaller the second term of the upper bound on our theoretical error is. There is a trade-off here.

No gain for a uniform prior with finite $\mathcal{H}$

Let's assume that $\mathcal{H}$ is finite, that the prior P is uniform and that Q always outputs the same classifier: there exists $h^* \in \mathcal{H}$ such that $Q(h^*) = 1$ and $Q(h) = 0$ for $h \neq h^*$.

Inequality 3 becomes

$$err_D(h^*) \leq \hat{err}(h^*) + \sqrt{\frac{\ln|\mathcal{H}| + \ln\frac{m}{\delta}}{2(m-1)}} \quad (4)$$

The upper bound of 1 is tighter than the upper bound of 4.

Gain for Gaussian prior and posterior: illustration with the SVM on MNIST

We use the MNIST dataset for handwritten digits, where each instance is a vectorized image of a digit. The dimension of these vectorized images is $28 * 28 = 784$. Focusing on a binary classification problem, we consider digits 0 and 1. The training dataset is made of 2000 examples and the testing dataset of 200 examples.

Let's use a prior $P \sim \mathcal{N}(0, I)$ and a posterior $Q \sim \mathcal{N}(u\mathbf{w}, I)$, where u and $\mathbf{w}$ are to choose in an optimal way. With the choice of the forms of the prior P and of the posterior Q, we can compute explicitly $RE(Q||P) = \frac{u^2}{2}$. With $\mathbf{v} \sim P$, we define the classifier thanks to $\mathbf{v'} = \mathbf{v} + u\mathbf{w}$: a vector of pixels $\mathbf{x}$ is labeled according to the sign of $\mathbf{v'}^T\mathbf{x}$.

We learn $\mathbf{w}$ thanks to a small subset of data. Inequality 3 holds for all posteriors Q. So if we want to get the best upper bound for $err_D(Q)$, we need to choose a posterior to minimize the right-hand side of 3. The optimization problem is the following:

$$\min_{u>0} \quad \hat{err}(Q)(\mathbf{w}, u) + \sqrt{\frac{\frac{u^2}{2} + \ln\frac{m}{\delta}}{2(m-1)}}$$

Tight bounds in the PAC-Bayes framework

Let's compare the upper bound on the generalization error (right-hand side of inequalities 2 or 3) to the error on the testing dataset. The results for the PAC-Bayes classifier are average of 100 results for random vectors $\mathbf{v}_k + u\mathbf{w} \sim \mathcal{N}(u\mathbf{w}, I)$.

	Upper bound	Test error	Ratio test error/upper bound
PAC	1.401	0.071	5.07%
PAC-Bayes	0.248	0.132	53.2%

We can fully appreciate the findings of the PAC-Bayes analysis regarding the tight bound we obtained: the upper bound given by the PAC-Bayes framework is way more realistic than the one given by the usual PAC framework.

References

[1] Shalev-Shwartz, S. and Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms.

[2] Shawe-Taylor, J. (2009). PAC-Bayes Analysis: Background and Applications. *Chicago/TTI Workshop.*

[3] Xie, X. and Sun, S. (2015). PAC-Bayes Analysis for Twin Support Vector Machines. *2015 International Joint Conference on Neural Networks.*