Machine learning theory

PAC-Bayesian Theory

Hamid Beigy

Sharif university of technology

June 15, 2020

Table of contents

1. Introduction
2. Bayesian methods
3. PAC-Bayes theory
4. Summary

Introduction

- PAC (Probably Approximately Correct) Learning provides guarantees on the expected error (approximately) of prediction rules that hold with high probability (probably) with respect to representativeness of the observed sample.
- In PAC approach, we choose hypothesis class H as the prior knowledge.
- The PAC approach has the advantage that one can prove guarantees for generalization error without assuming the truth of the prior.
- How to incorporate more complicated prior knowledge.
- The Bayesian approach has the advantage of using arbitrary domain knowledge in the form of a Bayesian prior.
- A PAC-Bayesian approach to machine learning attempts to combine the advantages of both PAC and Bayesian approaches.
- A PAC-Bayesian approach bases the bias of the learning algorithm on an arbitrary prior distribution, thus allowing the incorporation of domain knowledge, and yet provides a guarantee on generalization error that is independent of any truth of the prior.

Bayesian methods

- Let the data is drawn from a distribution that comes from some parametric family.

Example (Gaussian distribution)

Let σ be a known fixed parameter. Then, $\mathbb{P}[y \mid \mathbf{x}; \mathbf{w}] = \mathcal{N}\left(\langle \mathbf{w}, \mathbf{x} \rangle, \sigma^2\right) = \langle \mathbf{w}, \mathbf{x} \rangle + \mathcal{N}\left(0, \sigma^2\right)$ is a parametric family.

- Given a sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, we define the likelihood of $\mathbf{w}$ as

$$\mathcal{L}(\mathbf{w}, S) = \log\left(\mathbb{P}\left[y_1, \ldots, y_m \mid \mathbf{x}_1, \ldots, \mathbf{x}_m; \mathbf{w}\right]\right) = \sum_{i=1}^{m} \log\left(\mathbb{P}\left[y_i \mid \mathbf{x}_i; \mathbf{w}\right]\right)$$

- The maximum livelihood is the given value of $\mathbf{w}$ that maximizes $\mathcal{L}(\mathbf{w}, S)$ $\left(\mathbf{w} = \underset{\mathbf{w}'}{\operatorname{argmax}}\ \mathcal{L}(\mathbf{w}', S)\right)$

Example (Gaussian distribution)

1. Let σ be a known fixed parameter. Then, $\mathbb{P}[y \mid \mathbf{x}; \mathbf{w}] = \mathcal{N}\left(\langle \mathbf{w}, \mathbf{x} \rangle, \sigma^2\right) = \langle \mathbf{w}, \mathbf{x} \rangle + \mathcal{N}\left(0, \sigma^2\right)$ is a parametric family.
2. This means that $\mathbb{P}[y_i \mid \mathbf{x}_i; \mathbf{w}] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \langle \mathbf{w}, \mathbf{x} \rangle)^2}{\sigma^2}\right)$ and the likelihood is $\mathcal{L}(\mathbf{w}, S) = -\sum_{i=1}^{m} \frac{1}{\sigma^2} \frac{(y_i - \langle \mathbf{w}, \mathbf{x} \rangle)^2}{\sigma^2} + C$, where C is a normalization factor that does not depend on $\mathbf{w}$.
3. This means that maximum likelihood is equivalent to minimizing square loss.
4. We want to maximize $\mathbb{P}[\mathbf{w} \mid \mathbf{x}, y]$.

- To find $\mathbb{P}[\mathbf{w} \mid \mathbf{x}, y]$, we need to a prior distribution $\mathbb{P}[\mathbf{w}]$.
- We have $\mathbb{P}[y \mid \mathbf{x}, \mathbf{w}]$ and $\mathbb{P}[\mathbf{w}]$ from Bayes Theorem, hence, we have

$$\mathbb{P}[\mathbf{w} \mid \mathbf{x}, y] = \frac{\mathbb{P}[y \mid \mathbf{x}, \mathbf{w}]\, \mathbb{P}[\mathbf{w}]}{\mathbb{P}[y \mid \mathbf{x}]} \propto \mathbb{P}[y \mid \mathbf{x}, \mathbf{w}]\, \mathbb{P}[\mathbf{w}].$$

- The maximum a posteriori (MAP) model is

$$\mathbf{w} = \underset{\mathbf{w}'}{\operatorname{argmax}}\ \mathbb{P}\left[\mathbf{y} \mid \mathbf{X}, \mathbf{w}'\right] \mathbb{P}\left[\mathbf{w}'\right] = \underset{\mathbf{w}'}{\operatorname{argmax}}\ \mathcal{L}(\mathbf{w}', S) + \log \mathbb{P}\left[\mathbf{w}'\right]$$

Example (Gaussian distribution (cont.))

1. Let $\mathbb{P}[\mathbf{w}] = \mathcal{N}(0, \sigma_{\mathbf{w}}^2 \mathbf{I})$ be prior distribution on $\mathbf{w}$.
2. Now, we have

$$\begin{aligned}
\mathbf{w} &= \underset{\mathbf{w}'}{\operatorname{argmax}}\ -\sum_{i=1}^{m} \frac{1}{\sigma^2} \frac{(y_i - \langle \mathbf{w}', \mathbf{x} \rangle)^2}{\sigma^2} - \frac{1}{\sigma^2} \left\| \mathbf{w}' \right\|_2^2 \\
&= \underset{\mathbf{w}'}{\operatorname{argmin}}\ \sum_{i=1}^{m} \frac{1}{\sigma^2} \frac{(y_i - \langle \mathbf{w}', \mathbf{x} \rangle)^2}{\sigma^2} + \frac{1}{\sigma^2} \left\| \mathbf{w}' \right\|_2^2
\end{aligned}$$

3. This is equivalent to doing regularized ERM with L_2 regularization.
4. If we use Laplacian distribution instead of Gaussian, we will get L_1 regularization.

- MAP picks the best model, given our model and data.
- Why do we have to pick one model?
- We have seen that the optimal classifier can be calculated given $\mathbb{P}[y \mid \mathbf{x}]$.
- The Bayesian approach does exactly that, so we get

$$\mathbb{P}[y \mid \mathbf{x}, S] = \int_{\mathbf{w}} \mathbb{P}[y \mid \mathbf{x}, \mathbf{w}] \, \mathbb{P}[\mathbf{w} \mid S] \, d\, \mathbb{P}[\mathbf{w}]$$

- In some cases (such as Guassian), this as an analytic solution, but most of the time there isn't any.

PAC-Bayes theory

- In agnostic PAC learning, this prior is defined as selecting the hypothesis class H.
- In SRM learning, this prior is defined as the weights assigned to different hypothesis class H_n.
- In MDL, this prior is defined as the description length of hypothesis h.
- In the above models, the output of the learning algorithm is a single hypothesis h, i.e $h = A(S)$.
- In PAC-Bayes, algorithms return a distribution Q on H.

Example (Loss of posterior)

Let Q be a distribution on H, $\mathcal{D}$ a distribution on $\mathcal{X} \times \mathcal{Y}$ and S a finite sample. Define

$$\mathbf{R}(Q) = \mathbb{E}_{h\sim Q}[\mathbf{R}(h)] = \mathbb{E}_{h\sim Q}\left[\mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]\right]$$

$$\hat{\mathbf{R}}(Q) = \mathbb{E}_{h\sim Q}\left[\hat{\mathbf{R}}(h)\right] = \mathbb{E}_{h\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\ell(h,z)\right]$$

- The learning algorithm is
  1. Define prior distribution P on H.
  2. Get sample $S \sim \mathcal{D}^m$.
  3. Define/find posterior distribution Q on H.

- We can turn a posterior into a learning algorithm.

Definition (Gibbs classifier)

Let Q be a distribution on H. The Gibbs classifier is the following randomized hypothesis

1. Pick $h \in H$ according to $Q(h)$.
2. Observe $\mathbf{x}$.
3. Return $h(\mathbf{x})$.

- It is straightforward to show that the expected loss Gibbs classifier equals to $\mathbf{R}(Q)$.

Example

1. Let $H = \{h_1, \ldots, h_k\}$.
2. Let P be a uniform distribution over H.
3. Let Q be defined as

$$Q(h) = \begin{cases} 1 & \text{if } h = h_{erm} \\ 0 & \text{if } h \neq h_{erm} \end{cases}$$

Example

1. For $\mathbf{w} \in \mathbb{R}^n$, define

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} +1 & \text{with probability } \frac{1}{Z} e^{\langle \mathbf{w}, \mathbf{x} \rangle} \\ -1 & \text{with probability } \frac{1}{Z} e^{-\langle \mathbf{w}, \mathbf{x} \rangle} \end{cases}$$

2. The prior P is $\mathcal{N}\left(0, \sigma^2 \mathbf{I}\right)$, i.e. $P(h_{\mathbf{w}}) \propto \exp(-\|\mathbf{w}\|^2 / \sigma^2)$.
3. Given sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \sim \mathcal{D}^m$, and sample $h \sim P$ and output $S = \{(\mathbf{x}_1, h(y_1)), \ldots, (\mathbf{x}_m, h(y_m))\}$. Then likelihood equals to

$$\mathbb{P}\left[y_1, \ldots, y_m \mid h_{\mathbf{w}}, \mathbf{x}_1, \ldots, \mathbf{x}_m\right] = \prod_i \frac{1}{Z} e^{\langle \mathbf{w}, \mathbf{x}_i \rangle} \propto \exp\left(\sum_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle\right).$$

4. Using Bayes' rule, we can form the posterior

$$\begin{aligned} \mathbb{P}\left[h_{\mathbf{w}} \mid y_1, \ldots, y_m, \mathbf{x}_1, \ldots, \mathbf{x}_m\right] &\propto \left(\exp\left(\sum_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle\right)\right)\left(\exp\left(-\frac{\|\mathbf{w}\|^2}{\sigma^2}\right)\right) \\ &\propto \left(\exp\left(\sum_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle\right) - \frac{\|\mathbf{w}\|^2}{\sigma^2}\right) \end{aligned}$$

We will see that the critical factor determining the complexity of the learning algorithm will become $KL(Q||P)$, the Kullback-Liebler divergence from Q to P instead of the Rademacher complexity.

- We want to show that if Q is similar to P, the classifier generalizes well.
- Kullback-Leibler (KL) divergence is how to measure the similarity of two distributions.

Definition (KL divergence)

Let P and Q be continuous or discrete distributions. Then, KL divergence of distributions P and Q defined as

$$KL(Q||P) = \underset{x\sim Q}{\mathbb{E}}\left[\ln\left(\frac{Q(x)}{P(x)}\right)\right].$$

- Note that KL divergence is not symmetric, i.e. $KL(Q||P) \neq KL(P||Q)$.
- The intuition behind this definition comes from information theory.
- Assume we have a finite alphabet and message x is sent with probability $P(x)$.
- Shannon's coding theorem states that code of x with $\log_2(1/P(x))$ bits is an optimal coding and the expected bits per letter is $\mathbb{E}_{x\sim P}\left[\log_2\left(\frac{1}{P(x)}\right)\right] = H(P)$.
- Consider now that we use the optimal code for P, but the letters where sent according to Q. The expected bits per letter is now

$$\underset{x\sim Q}{\mathbb{E}}\left[\log_2\left(\frac{1}{P(x)}\right)\right] = \underset{x\sim Q}{\mathbb{E}}\left[\log_2\left(\frac{Q(x)}{P(x)}\right) + \log_2\left(\frac{1}{Q(x)}\right)\right] = H(Q) + KL(Q||P).$$

- $KL(Q||P)$ is the extra number of bits expected per letter from using P instead of Q to create the codebook.
- This shows that $KL(Q||P) \geq 0$.

Example

Let P be some distribution on $\mathbf{x}_1, \ldots, \mathbf{x}_m$ and Q be 1 on $\mathbf{x}_i$ then, $KL(Q||P) = \ln\left(\frac{1}{P(x_i)}\right)$.

Example

Let $P(\mathbf{x}_i) = 0$ and $Q(\mathbf{x}_i) > 0$, then $KL(Q||P) = \infty$.

Example

Let $\alpha, \beta \in [0,1]$, then $KL(\alpha||\beta) = KL(Ber(\alpha)||Ber(\beta)) = \alpha \ln\left(\frac{\alpha}{\beta}\right) + (1-\alpha)\ln\left(\frac{1-\alpha}{1-\beta}\right)$.

Show the above equation.

Example

Let $Q = \mathcal{N}(\mu_0, \Sigma_0)$ and $P = \mathcal{N}(\mu_1, \Sigma_1)$ be two n-dimensional Gaussian distributions. Then,

$$KL(Q||P) = \frac{1}{2}\left(\mathrm{Tr}\left[\Sigma_1^{-1}\Sigma_0\right] + (\mu_1 - \mu_0)\Sigma_1^{-1}(\mu_1 - \mu_0) - n - \frac{\det(\Sigma_0)}{\det(\Sigma_1)}\right)$$

Show the above equation.

Lemma

If X is a real valued random number satisfying $\mathbb{P}\left[X \leq x\right] \leq e^{-mf(x)}$, *then* $\mathbb{E}\left[e^{(m-1)f(x)}\right] \leq m$.

Lemma

With probability greater then $(1-\delta)$ *over* S,

$$\mathop{\mathbb{E}}_{h \sim P}\left[e^{(m-1)KL(\hat{\mathbf{R}}(h)||\mathbf{R}(h))}\right] \leq \frac{m}{\delta}.$$

Lemma (Shift of measure)

$$\mathop{\mathbb{E}}_{x \sim Q}\left[f(x)\right] \leq KL(Q||P) + \ln \mathop{\mathbb{E}}_{x \sim P}\left[e^{f(x)}\right].$$

Theorem (PAC Bayes bound)

Let Q and P be distributions on H and $\mathcal{D}$ be a distribution on $\mathcal{X} \times \mathcal{Y}$. Also let $\ell(h, z) \in [0, 1]$ and $S \sim \mathcal{D}^m$ be a sample of size m, then with probability greater or equal to $(1 - \delta)$ over S we have

$$KL(\hat{\mathbf{R}}(Q)||\mathbf{R}(Q)) \leq \frac{KL(P||Q) + \ln\left(\frac{m+1}{\delta}\right)}{m}.$$

1. The left-hand side is the KL divergence between two numbers; while the right-hand side is the KL divergence between distributions.
2. We assume no connection between $\mathcal{D}$ and P (an agnostic analysis).

Proof (PAC Bayes bound).

1. Define $f(h) = KL(\hat{\mathbf{R}}(h)||\mathbf{R}(h))$. Using the Lemma Shift of measure and its preceding lemma, we get
$$\mathop{\mathbb{E}}_{h\sim Q}[mf(h)] \leq KL(Q||P) + \ln \mathop{\mathbb{E}}_{h\sim P}\left[e^{mf(h)}\right] \leq KL(Q||P) + \ln\left(\frac{m+1}{\delta}\right)$$
2. Since KL divergence is convex, so from the Jensen inequality
$$\begin{aligned} KL(\hat{\mathbf{R}}(Q)||\mathbf{R}(Q)) &= KL(\mathop{\mathbb{E}}_{h\sim Q}\left[\hat{\mathbf{R}}(h)\right] || \mathop{\mathbb{E}}_{h\sim Q}[\mathbf{R}(h)]) \\ &\leq \mathop{\mathbb{E}}_{h\sim Q}\left[KL(\hat{\mathbf{R}}(h)||\mathbf{R}(h))\right] = \mathop{\mathbb{E}}_{h\sim Q}[f(h))] \end{aligned}$$

$\square$

- We bounded $KL(\hat{\mathbf{R}}(Q)||\mathbf{R}(Q))$.
- Now, we bound $\mathbf{R}(Q) - \hat{\mathbf{R}}(Q)$.

Lemma

Let $a, b \in [0,1]$ and $KL(a||b) \leq x$, then $b \leq a + \sqrt{\frac{x}{2}}$ and $b \leq a + 2x + \sqrt{2ax}$, where the second is much stronger if a is very small.

Theorem (Generalization bounds)

Let Q and P be distributions on H and $\mathcal{D}$ be a distribution on $\mathcal{X} \times \mathcal{Y}$. Let also $\ell(h, z) \in [0,1]$ and $S \sim \mathcal{D}^m$ be a sample, then with probability greater or equal to $(1-\delta)$ over S we have

$$\mathbf{R}(Q) \leq \hat{\mathbf{R}}(Q) + \sqrt{\frac{KL(Q||P) + \ln\left(\frac{m+1}{\delta}\right)}{2m}}$$

$$\mathbf{R}(Q) \leq \hat{\mathbf{R}}(Q) + 2\frac{KL(Q||P) + \ln\left(\frac{m+1}{\delta}\right)}{m} + \sqrt{2\hat{\mathbf{R}}(Q)\frac{KL(Q||P) + \ln\left(\frac{m+1}{\delta}\right)}{m}}$$

Summary

- Shawe-Taylor et al. gave PAC analysis of Bayesian estimators.
- McAllester gave PAC-Bayesian bound.
- PAC-Bayes bounds hold even if prior incorrect;while Bayesian inference must assume prior is correct.
- PAC-Bayes bounds hold for all posteriors; while in Bayesian learning, posterior computed by Bayesian inference, depends on statistical modeling
- PAC-Bayes bounds can be used to define prior, hence no need to be known explicitly; while in Bayesian learning, input effectively excluded from the analysis, randomness lies in the noise model generating the output.
- We analyzed Gibbs classifier. Another solution is to sample many $h_i \sim Q$ i.i.d. and output the majority vote.
- PAC-Bayes theory gives the tightest known generalization bounds for SVMs, with fairly simple proofs.
- PAC-Bayesian analysis applies directly to algorithms that output distributions on the hypothesis class, rather than a single best hypothesis.
- However, it is possible to de-randomize the PAC-Bayes bound to get bounds for algorithms that output deterministic hypothesis.

Readings

1. Chapter 31 of Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning : From theory to algorithms*. Cambridge University Press, 2014.
2. The papers given in References [4, 2, 3, 1].

David A. McAllester. "A PAC-Bayesian Tutorial with A Dropout Bound". In: *CoRR* abs/1307.2118 (2013).

David A. McAllester. "PAC-Bayesian Stochastic Model Selection". In: *Machine Learning* 51.1 (2003), pp. 5–21.

David A. McAllester. "Simplified PAC-Bayesian Margin Bounds". In: *Lecture Notes in Computer Science*. Vol. 2777. Springer, 2003, pp. 203–215.

David A. McAllester. "Some PAC-Bayesian Theorems". In: *Machine Learning* 37.3 (1999), pp. 355–363.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning : From theory to algorithms*. Cambridge University Press, 2014.

Questions?