TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

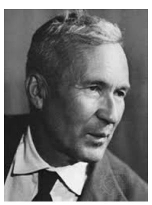
Learning Theory I

The Occam Generalization Guarantee

aka: The Free Lunch Theorem

Chomsky vs. Kolmogorov and Hinton

Noam Chomsky: Natural language grammar cannot be learned by a universal learning algorithm. This position is supported by the "no free lunch theorem".



Andrey Kolmogorov, Geoff Hinton: Universal learning algorithms exist. This position is supported by the "free lunch theorem".

The No Free Lunch Theorem

Without prior knowledge, such as universal grammar, it is impossible to make a prediction for an input you have not seen in the training data.

Proof: Select a predictor h uniformly at random from all functions from $\mathcal{X}$ to $\mathcal{Y}$ and then take the data distribution to draw pairs $(x, h(x))$ where x is drawn uniformly from $\mathcal{X}$. No learning algorithm can predict $h(x)$ where x does not occur in the training data.

The Occam Guarantee (Free Lunch Theorem)

Consider a classifier f written in Python with an arbitrarily large standard library.

Let $|f|$ be the number of bits needed to represent f (any compression algorithm is allowed).

The Occam Guarantee (Free Lunch Theorem)

$$0 \leq \mathcal{L}(h, x, y) \leq L_{\mathrm{max}}$$

$$\begin{aligned} \mathcal{L}(h) &= E_{(x,y)\sim \mathrm{Pop}}\; \mathcal{L}(h, x, y) \\ \hat{\mathcal{L}}(h) &= E_{(x,y)\sim \mathrm{Train}}\; \mathcal{L}(h, x, y) \end{aligned}$$

Theorem: With probability at least $1 - \delta$ over the draw of the training data the following holds simultaneously for all f.

$$\mathcal{L}(f) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(f) + \frac{5L_{\mathrm{max}}}{N_{\mathrm{Train}}}\left((\ln 2)|f| + \ln \frac{1}{\delta}\right)\right)$$

Occam Guarantee (Probability Form)

Code length is inter-convertable with with probability.

$$P(h) = 2^{-|h|} \quad \text{or} \quad |h| = -\log_2 P(h)$$

Instead of fixing the language (e.g., Python with a large library) we fix a prior $P(h)$.

Theorem: With probability at least $1 - \delta$ over the draw of training data the following holds simultaneously for all h.

$$\mathcal{L}(h) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N_{\text{Train}}}(-\ln \delta P(h))\right)$$

Occam vs. Bayes

For $\mathcal{L}(h,x,y) = -\ln P_h(y|x) \leq L_{\max}$ we have

$$\text{Occam:} \qquad \mathcal{L}(h) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N_{\text{Train}}}(-\ln \delta P(h))\right)$$

$$h^* = \operatorname*{argmin}_h \hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N_{\text{Train}}}(-\ln P(h))$$

$$\text{Bayes:} \qquad h^* = \operatorname*{argmax}_h P(h|\text{Train})$$

$$h^* = \operatorname*{argmin}_h \hat{\mathcal{L}}(h) + \frac{1}{N_{\text{Train}}}(-\ln P(h))$$

Proof

Define

$$\epsilon(h) = \sqrt{\frac{2\mathcal{L}(h)\,(-\ln \delta P(h))}{L_{\max} N_{\mathrm{Train}}}}.$$

By the relative Chernov bound we have

$$P_{\mathrm{Train}\sim\mathrm{Pop}}\left(\frac{\hat{\mathcal{L}}(h)}{L_{\max}} \leq \frac{\mathcal{L}(h)}{L_{\max}} - \epsilon(h)\right) \leq e^{-N_{\mathrm{Train}}\frac{\epsilon(h)^2 L_{\max}}{2\mathcal{L}(h)}} = \delta P(h).$$

Proof

$$P_{\mathrm{Train}\sim\mathrm{Pop}}\left(\hat{\mathcal{L}}(h) \leq \mathcal{L}(h) - L_{\mathrm{max}}\epsilon(h)\right) \leq \delta P(h).$$

$$P_{\mathrm{Train}\sim\mathrm{Pop}}\left(\exists h\ \hat{\mathcal{L}}(h) \leq \mathcal{L}(h) - L_{\mathrm{max}}\epsilon(h)\right) \leq \sum_{h} \delta P(h) = \delta$$

$$P_{\mathrm{Train}\sim\mathrm{Pop}}\left(\forall h\ \mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + L_{\mathrm{max}}\epsilon(h)\right) \geq 1 - \delta$$

Proof

$$\mathcal{L}(h) \leq \widehat{\mathcal{L}}(h) + L_{\max}\sqrt{\mathcal{L}(h)\left(\frac{2L_{\max}\left(-\ln \delta P(h)\right)}{N_{\mathrm{Train}}}\right)}$$

using

$$\sqrt{ab} = \inf_{\lambda>0}\; \frac{a}{2\lambda} + \frac{\lambda b}{2}$$

we get

$$\mathcal{L}(h) \leq \widehat{\mathcal{L}}(h) + \frac{\mathcal{L}(h)}{2\lambda} + \frac{\lambda L_{\max}\left(-\ln \delta P(h)\right)}{N_{\mathrm{Train}}}$$

Proof

$$\mathcal{L}(h) \leq \widehat{\mathcal{L}}(h) + \frac{\mathcal{L}(h)}{2\lambda} + \frac{\lambda L_{\max}\left(-\ln \delta P(h)\right)}{N_{\mathrm{Train}}}$$

Solving for $\mathcal{L}(h)$ yields

$$\mathcal{L}(h) \leq \frac{1}{1-\frac{1}{2\lambda}}\left(\hat{\mathcal{L}}(h) + \frac{\lambda L_{\max}}{N_{\mathrm{Train}}}\left(-\ln \delta P(h)\right)\right)$$

Setting $\lambda = 5$ brings the leading factor to $10/9$ which seems sufficiently close to 1 that larger values of λ need not be considered.

END