MPML final report

Wei-Tsung, Kao, Yen-Li, Laih

B05901009, B05901184

Abstract

In this report, we started by following the line of PAC-learnability and introduced the non-uniform learnability, which use union bound to reach uniform result. After that, we presented a PAC-Bayes bound that deal with the function class that contains uncountable functions. An example of how we can leverage this kind of bound is shown and discussed. In order to use this bound for practical use, we then introduced how we can replace the stochastic term in PAC-Bayes into deterministic term using a noise-resilient framework. We will mention two ways to use this noise-resilient framework.

1 Introduction

In class, we have learned lots about PAC learning framework. Given a function class $\mathcal{F}$, a loss function l, and $\epsilon, \delta \in (0,1)$, we would like to derive:

$$\mathbb{P}(L(f) - \hat{L}(f) \geq \epsilon) \leq \delta,\ \forall f \in \mathcal{F}$$

or say with probability at least $1-\delta$, we have:

$$L(f) - \hat{L}(f) < \epsilon,\ \forall f \in \mathcal{F}$$

If the loss function is bounded (e.g. 0-1 loss), as we have learned in class, we can use concentration inequality like bounded difference inequality to get the result, with probability at least $1-\delta$:

$$L(f) - \hat{L}(f) < \sqrt{\frac{1}{2m}\ln\frac{2}{\delta}},\ for\ any\ f \in \mathcal{F}$$

But the above is not uniform result over function class $\mathcal{F}$. To get the uniform result over the function class $\mathcal{F}$, we still need some effort. If there're countable functions in $\mathcal{F}$, we can simply use union bound over all function and get the result, with probability at least $1-\delta$:

$$L(f) - \hat{L}(f) < \sqrt{\frac{\frac{1}{w(f)} + \ln\frac{2}{\delta}}{2m}},\ \forall f \in \mathcal{F}$$

$$\sum_{f\in\mathcal{F}} w(f) \leq 1$$

Here m is number of training samples, $w(f)$ is weighting factor on the confidence δ, which means that we have different confidence about different classifier f. Notice that it requires sum of all w be less or equal to one because we use union bound. This kind of approach is also called *nonuniform learnablility* because for different f, the generalization bound is different.

But if there're uncountable functions in $\mathcal{F}$, union bound doesn't work, and we need to use other skills. For example, we can try to bound the uniform deviation of the function class and use Rademacher complexity or

VC-dimension, which are related to the complexity of the function class, to derive the uniform convergence result. We can also use algorithm centric approach such as stability. In stability, we want to find an upper bound on the uniform replace-one stability, which may depend on the algorithm, and then we get a uniform result on the generalization gap. In modern machine learning, the learning model (or say the function class $\mathcal{F}$) is always large, and the model structure is always very complex, too. It's not that easy to derive a tight upper bound on the VC-dimension of model, or the result may not very useful because it may depend on number of paprameters of the model, which makes the generalization bound loose. For algorithm centric approach, it doesn't observe the property of the function class very much.

So the question emerges: Are there other approaches or frameworks that can be used to derive uniform result on modern machine model? That is why we survey PAC-Bayes framework.

2 Background

When dealing with function class containing uncountable functions we have to introduce a new kind of inequality - PAC-Bayes bound. To talk about this bound we first need to define a posterior distribution over our function class $\mathcal{F}$, which we denote by Q. From the aspect of a supervised learning problem, where $\mathcal{F}$ contains functions from $\mathcal{X}$ to $\mathcal{Y}$, we can think of Q as a randomized classifier, which randomly drawn a function $f \in \mathcal{F}$ from the distributions as the prediction rule when given an input x $\in \mathcal{X}$. In practical, one can get this kind of randomized classifier by adding noise on parameter of a deterministic model. We define the loss of Q on an example z $\equiv$ (x,y) to be.

$$l(Q,z) \overset{def}{=} \underset{f \sim Q}{E}[l(f,z)]$$

Now the training loss and testing loss of Q can be defined as

$$L_S(Q) \overset{def}{=} \underset{f \sim Q}{E}[L_S(h)] \quad and \quad L_D(Q) \overset{def}{=} \underset{f \sim Q}{E}[L_D(h)]$$

After understanding the nature of Q we can introduce the PAC-Bayes bound.

Theorem 1. *Let D be an arbitrary distribution over an example domain Z. Let F be a function class and let $l : F \times Z \to [0, 1]$ be a loss function. Let P be a prior distribution over F and let $\delta \in (0, 1)$. Then, with probability of at least $1 - \delta$ over the choice of an i.i.d. training set $S = \{z_1, ..., z_m\}$ sampled according to D, for all distributions Q over F (even such that depend on S), we have*

$$L_D(Q) \leq L_S(Q) + \sqrt{\frac{KL(Q\|P) + \ln\frac{2m}{\delta}}{2(m-1)}}$$

where $KL(Q\|P)$ is the Kullback-Leibler divergence.

The proof is left to appendix. Notice that there is no constraint on the function class F, and prior distribution P can be an arbitrary distribution over the function class F as long as it is independent of the training set S. The PAC-Bayes bound can actually lead to the following learning rule.

Given a prior P return a posterior Q that minimizes the objective function

$$L_S(Q) + \sqrt{\frac{KL(Q\|P) + \ln\frac{2m}{\delta}}{2(m-1)}}$$

This rule is actually very similar to the common *regularized risk minimization* principle, where the first term is the empirical loss and the second term acting as the regularization on Q. Notice that the KL divergence in the second term measures the distance between Q and P and can be seen as a term that stops Q from over-fitting on the training sample S.

Here we show an simple example of how we can use the PAC-Bayes bound. First we define the posterior Q as $N(w, \sigma^2)$, a normal distribution with mean w and variance σ^2. The parameter w is learned by an training algorithm on the training set S. We then choose the prior distribution as $N(0, \sigma^2)$, hence from the definition of KL divergence we have $KL(Q\|P) = \frac{1}{2}\|w\|_2^2$, notice that the value of sigma can be arbitrary. We obtain the following equation.

$$L_D(Q) \leq L_S(Q) + \sqrt{\frac{\frac{1}{2}\|w\|_2^2 + \ln\frac{2m}{\delta}}{2(m-1)}}$$

we can see that by choosing the same sigma for Q and P, we can replace the KL divergence term with the l_2 norm of w. This kind of regularization is used commonly in practical and the PAC-Bayes bound is one way to explain why this kind of regularization works.

However, the prior P we choose above is fairly simple and doesn't include too much information. In fact, we can actually choose certain P based on some domain knowledge on the example domain Z or task that can even further bound the KL term. For example, in a more high level aspect, we can assume that the learned posterior Q is more likely to distribute over the smoother functions due to some property of the example domain. Thus, we can choose prior P over some smooth functions, leading to a smaller KL divergence between P and Q. Again, the only constraint of P is that it doesn't depend on the training set S.

Theoretically, the best prior distribution P that can minimize the expected value of $KL(Q\|P)$ can be acquirde when one knows the data distribution D from which S is drawn. In fact the P can be expressed as

$$P(f) = \mathrm{E}_{S\sim D^m}[Q_A(S)]$$

where $Q_A(S)$ is the posterior distribution learned by the learning algorithm A on training set S. The Proof is as follow.

$$\begin{aligned}\mathrm{E}_S[KL(Q_A(S)\|P)] &= \mathrm{E}_{S,f\sim Q_A(S)}[\ln\frac{Q_A(S)(f)}{P(f)}] \\ &= \mathrm{E}_{S,f\sim Q_A(S)}[\ln\frac{Q_A(S)(f)}{\widetilde{Q}_A(f)}] + \mathrm{E}_{f\sim\widetilde{Q}_A}[\ln\frac{\widetilde{Q}(f)}{P(f)}] \\ &= \mathrm{E}_S[KL(Q_A(S)\|\widetilde{Q}_A)] + KL(\widetilde{Q}_A\|P)\end{aligned}$$

where $\widetilde{Q}_A(f) = \mathrm{E}_{S\sim D^m}[Q_A(S)]$
Thus we can minimize $\mathrm{E}_S[KL(Q_A(S)\|P)]$ by choosing P(f) = $\widetilde{Q}_A(f)$

However, note that in practical, one can never know the true data distribution D.

3 Randomized to Deterministic

The sections above tell the emergence of PAC-Bayes bound, the reason why we need PAC-Bayes bound, and the way to utilize PAC-Bayes bound. However, PAC-Bayes bound can only be used to analyze the generalization gap of *randomized* classifier, thus it doesn't match what machine learning practitioners use. To utilize PAC-Bayes bonud to analyze a *deterministic* classifier, we need some skills to relate the training and testing loss of stochastic classifier to deterministic one. Here we introduce *noise-resilient* approach, one of common skills in recent research, and show that how can this kind of approach be used to analyze the generalization of deep neural network.

3.1 Noise-Resilient approach

The high level concept of noise-resilient approach is below:

(1) Inject the randomness by adding noise on the model parameter w

(2) Define function ρ to measure model's property

(3) The property of the model measured by ρ shouldn't change too much when added noise

(4) Because the noise doesn't affect the model too much, one can bound the testing loss of the deterministic classifier, $L_{0,deter}$, by the margin loss of th noised one, $L_{\gamma,noised}$

(5) Now the $L_{\gamma,noised}$ is the testing loss of a randomized classifier, then one can link it with its training loss, $\hat{L}_{\gamma,noised}$, by using PAC-Bayes bound

(6) Again, because the perturbation doesn't affect the model too much, the training loss of the noised classifier can further be bounded by the another margin training loss of the deterministic classifier, $\hat{L}_{\gamma',deter}$. And we successfully change the generalization bound of a randomized classifier to a deterministic one.

To use this kind of concept, we should construct a proper function ρ to measure the property of the model w, and carefully choose a proper magnitude of the noise to make the model able to reach the noise-resilient condition. Different function ρ we choose, different property of model we observe, and different generalization bound we will get. But if the function ρ is more complicated, it's more difficult to show that the noise-resilient condition is satisfied. In next sub-section, we show two different choice of ρ function, and the difference of them.

3.2 Choice of ρ function

In *A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks*, the ρ function is the output margin of the model, here we show some proof in the paper (the only difference is that we use the PAC-Bayes bound mentioned in Section 3) to explain how the noise-resilient approach really works:

Lemma 1. *Let $f_w(x) : \mathcal{X} \to R^k$ be any predictor (not necessarily a neural network) with parameters $\boldsymbol{w}$, $\mathcal{X}$ is feature space, and $\boldsymbol{P}$ be any distribution on the parameters that is independent of the training data. Then, for any $\gamma, \delta > 0$, with probability $\geq 1 - \delta$ over the training set of size m, for any $\boldsymbol{w}$, and any random perturbation $\boldsymbol{u}$ s.t. $P_u[max_{x\in\mathcal{X}}|f_{w+u}(x) - f_w(x)| < \frac{\gamma}{4}] \geq \frac{1}{2}$, we have:*

$$L_0(f_w) \leq \hat{L}_\gamma(f_w) + \sqrt{\frac{2KL(w+u||P) + ln\frac{8m}{\delta}}{2(m-1)}}$$

Proof. Let $w' = w + u,$ w is the parameter learned by our algorithm, and u is the noise. Let S_w be the of perturbation that satisfy:

$$S_w \subseteq \left\{ w' \mid \max_{x\in\mathcal{X}} |f_{w'}(x) - f_w(x)| < \frac{\gamma}{4} \right\}.$$

Let q be a probability density function(PDF) over all w', and $\tilde{q}$ be the PDF over all $\tilde{w}$ that:

$$\tilde{q}(\tilde{w}) = \frac{1}{Z}\left\{ \begin{array}{cc} q(\tilde{w}) & if\ \tilde{w} \in S_w \\ 0 & otherwise \end{array} \right.$$

Here Z is normalization constant, which is choose to satisfy $\mathbb{P}(\tilde{w} \in S_w) \geq a$, a is some constant we will decide later. Due to the definition of $\tilde{q}$, we have:

$$\max_{x\in\mathcal{X}} |f_{\tilde{w}}(x) - f_w(x)| < \frac{\gamma}{4},\ \forall \tilde{w} \in S_w$$

$$\Longrightarrow \underset{x\in\mathcal{X},i,j\in[1,k]}{max} \; |(|f_{\tilde{w}}(x)[i] - f_{\tilde{w}}(x)[j]| - |f_w(x)[i] - f_w(x)[j]|)| \tag{1}$$

$$\leq \underset{x\in\mathcal{X},i,j\in[1,k]}{max} \; |(|f_{\tilde{w}}(x)[i] - f_{\tilde{w}}(x)[j] - (f_w(x)[i] - f_w(x)[j])|)| \tag{2}$$

$$\leq \underset{x\in\mathcal{X},i,j\in[1,k]}{max} \; |(|f_{\tilde{w}}(x)[i] - f_w(x)[i] + f_w(x)[j] - f_{\tilde{w}}(x)[j]|)| < \frac{\gamma}{2} \tag{3}$$

Notice that the equation 1 above is maximum difference of the output margin between the noised classifier $\tilde{w}$ and the original classifier w. Thus we have:

$$\begin{gathered} L_0(f_w) \leq L_{\frac{\gamma}{2}}(f_{\tilde{w}}), \\ \hat{L}_{\frac{\gamma}{2}}(f_{\tilde{w}}) \leq \hat{L}_{\gamma}(f_w) \end{gathered}$$

which has done the step 4 and 6 in section 4.1. Now we can combine step 4, 5, and 6 by using PAC-Bayes bound, with a = $\frac{1}{2}$:

$$\begin{align} L_0(f_w) &\leq \mathbb{E}_{\tilde{q}}[L_{\frac{\gamma}{2}}(f_{\tilde{w}})] \tag{4} \\ &\leq \mathbb{E}_{\tilde{q}}[\hat{L}_{\frac{\gamma}{2}}(f_{\tilde{w}})] + \sqrt{\frac{KL(\tilde{q}||P) + \ln\frac{2m}{\delta}}{2(m-1)}} \tag{5} \\ &\leq \hat{L}_{\gamma}(f_w) + \sqrt{\frac{KL(\tilde{q}||P) + \ln\frac{2m}{\delta}}{2(m-1)}} \tag{6} \\ &\leq \hat{L}_{\gamma}(f_w) + \sqrt{\frac{2KL(q||P) + 2\ln 2 + \ln\frac{2m}{\delta}}{2(m-1)}} \tag{7} \\ &\leq \hat{L}_{\gamma}(f_w) + \sqrt{\frac{2KL(q||P) + \ln\frac{8m}{\delta}}{2(m-1)}} \tag{8} \end{align}$$

The inequality in equation 7 is due to the following: Let S_w^c be the complement set of S_w, and $\tilde{q}^c$ is the PDF over S_w^c. The definition of $\tilde{q}^c$ is like $\tilde{q}$, but the normalization constant of $\tilde{q}^c$ is $1 - Z$, and the set is S_w^c. Then,

$$\begin{aligned} KL(q||p) &= ZKL(\tilde{q}||p) + (1-Z)KL(\tilde{q}^c||p) - H(Z)) \\ KL(\tilde{q}||p) &= \frac{1}{Z}[KL(q||p) + H(Z) - (1-Z)KL(\tilde{q}^c||p)] \\ &\leq \frac{1}{a}(KL(q||p) + \ln 2) \end{aligned}$$

the last inequality is because: $H(Z) = -ZlnZ - (1-Z)ln(1-Z) \leq \ln 2, and\ KL(\tilde{q}^c||p) \geq 0.$ □

Notice that here the author of the paper choose the constant **a** to be $\frac{1}{2}$, but actually it's can be any constant in (0,1) , and it will affect the constant multiplied by the KL-term and the magnitude of noise to satisfied the *noise-resilient* condition: $\mathbb{P}(\tilde{w} \in S_w) \geq a$. In the later of [1], it use Gaussian distribution with mean $w + u$, variance σ^2 as posterior, and Gaussian distribution with mean 0, variance σ^2 as prior, so $KL(q||p) \leq \frac{|w|^2}{2\sigma^2}$ is inversely proportional to the variance (kind of magnitude mentioned before). The less **a** one chooses, the constant multiplied by the KL-term will be larger, but the variance of noise can be larger, too, thus make the KL-term smaller:

$$u \in \mathbb{R}^{h\times h},\ u \sim \mathcal{N}(o, \sigma^2 I),\ \mathbb{P}_u(\|u\|_2 > t) \leq 2he^{\frac{-t^2}{2h\sigma^2}}$$

and the perturbation of output margin scales with the spectral norm of noise in each layer. For more detail, please refer to Section 2 of [1].

After all, the choice of different **a** will just affect the constant multiplied by the KL-term in the final result, so the constant $\frac{1}{2}$ isn't very important. But it's worth noticing that the skill in [1] only observe the output margin of the model (the ρ function it defines.), so the detail of each layer in DNN doesn't matter, and the final result scales with the Frobenius norm of parameter of each layer ($|w|^2$ term), which may not be favorable. To avoid this, the question to ask is that if there're other ways to choose the ρ function to observe the detail of model structure.

3.3 Another choice of ρ function

Answer to this question is yes. In [2], the ρ function is a set of property of model: the L2-norm of the output of each layer, the pre-activation value of each layer, the L2-norm of row of the Jacobian matrix of any two layers d and d' (d' $\leq$ d), and the spectral norm of Jacobian matrix of each two layers d and d' (d' $\leq$ d). ρ function measures the difference between them and some layer-dependent constant. We skip the detail of proof here but summary the high level idea of utilizing this kind of ρ function:

(a) If the perturbation in layer 1 to layer d can be bounded by a set of constant $\mathcal{C}_d$, and the activation state of each neuron with ReLU activation (greater than 0 or not) is unchanged after added noise(so the propagation of perturbation, or say Jacobian, is unchanged due to the property of ReLU), then the perturbation in d+1 layer, $\mathcal{C}_{d+1}$, can be expressed by $\mathcal{C}_d$, the variance of noise, and the confidence δ. (because we will use high probability result in the noise-resilient condition)

(b) In layer d+1, we first bound the Jacobian terms, and then use them to further bound the output and the pre-activation terms, which means that the process to calculate $\mathcal{C}_{d+1}$ is not orderless.

(c) To reach the condition in (a), the key step is that we wish if $\mathcal{C}_d$ is satisfied, and ρ of layer d+1 is greater than 0, than with high probability $\mathcal{C}_{d+1}$ is satisfied. To make this work, it turns out that the minimum of the pre-activation of layer d shouldn't be too small, and the other three property in layer d shouldn't be too large.

(d) If (c) and $\mathcal{C}_d$ are satisfied, then the state-unchanged condition will be satisfied naturally (the perturbation is relative small in layer d), then the conditions in (a) are satisfied, which means $\mathcal{C}_{d+1}$ will be satisfied with high probability after doing (b) in layer d+1.

(e) Sequentially doing step (a) to (d) on each layer, we can prove that the noise-resilient conditions are satisfied when properly choosing the variance of the gaussian noise. Now we use training samples to decide the most proper variance that makes (a) to (d) work.

(f) Because we use high probability result in the noise-resilient conditions, there is still some probability that some training or testing data that makes the classifier not noise-resilient (the loss on these data is relative large). The last thing to do is bounding these terms with the probability of not noise-resilient (because we use 0-1 loss, the maximum of loss on one data is 1), then we can use the same skill (step 4. to step 6.) mentioned in Section 4.2, but with a little different because now we observe not only the output margin.

The final result in [2] successfully avoid scaling with the Frobenius norm or spectral norm of each layer. But due to the more complicated choice of ρ function, step to prove that the model satisfies these conditions becomes more complex, and the result is less intuitive. Besides, the result in [2] scales inversely with the minimum of the pre-activation value in the model, which may make this bound worse. Here we just want to show that the noise-resilient approach is quite flexible. It gives us a natural way to use PAC-Bayes bound, and a different way to observe the model structure.

4 Conclusion

In this report, we have introduce the PAC-Bayes bound and noise-resilient approach that can help us bound the generalization gap of a deterministic classifier. The main obstacle of this approach is constructing the noise-resilient properties and proving that it can be found in our model under practical conditions. Comparing to other approach such as VC-dimension, it changes the problem and offers us a different way to analyze the generalization gap. In our opinion, PAC-Bayes bound is a quite flexible tool and is worth thinking new approaches to utilize it in the future.