Notes 18: Random Classification Noise model

1. Statistical Query and Random Classification Noise

If $\mathcal{C}$ is efficiently learnable from SQ's, then $\mathcal{C}$ is efficiently PAC-learnable with RCN

Theorem 1. *If some efficient algorithm A learns $\mathcal{C}$ to error ε from M statistical queries of tolerance τ, then some efficient algorithm PAC-learns $\mathcal{C}$ with Random Classification Noise of rate η using*

$$O\left(\frac{M}{\tau^2(1-2\eta)^2}\ln\frac{M}{\delta}\right) \quad \textit{samples}$$

Proof. Suppose A makes a statistical query with predicate $\varphi : X \times \{+1,-1\} \to \{0,1\}$
Any such φ can be decomposed (uniquely) as $\varphi(x,y) = \underbrace{f(x)}_{\text{indep. of } y} + \underbrace{g(x)\cdot y}_{\text{linear in } y}$

$$\text{since} \quad \varphi(x,y) = \varphi(x,1)\mathbb{1}(y=1) + \varphi(x,-1)\mathbb{1}(y=-1) = \varphi(x,1)\frac{1+y}{2} + \varphi(x,-1)\frac{1-y}{2}$$
$$= \frac{\varphi(x,1)+\varphi(x,-1)}{2} + \frac{\varphi(x,1)-\varphi(x,-1)}{2}\cdot y$$

Estimating $\mathbb{E}_{\mathrm{EX}(c,\mathcal{D})}[\varphi(x,y)]$ within τ amounts to estimating expectations of both terms within $\tau/2$

1st term (independent of y) has the same expectation under $\mathrm{EX}(c,\mathcal{D})$ and under $\mathrm{EX}^\eta(c,\mathcal{D})$
Since $f(x) = (\varphi(x,1)+\varphi(x,-1))/2$ takes a value between 0 and 1
With prob $\geqslant 1-\delta/2M$, can estimate $\mathbb{E}_{\mathrm{EX}(c,\mathcal{D})}[f(x)]$ within $\frac{\tau}{2}$ using $O\left(\frac{1}{\tau^2}\ln\frac{M}{\delta}\right)$ samples

2nd term (linear in y) has expectation

$$\mathop{\mathbb{E}}_{\mathrm{EX}^\eta(c,\mathcal{D})}[g(x)\cdot y] = (1-\eta)\mathop{\mathbb{E}}_{\mathrm{EX}(c,\mathcal{D})}[g(x)\cdot y] + \eta \mathop{\mathbb{E}}_{\mathrm{EX}(c,\mathcal{D})}[g(x)\cdot -y] = (1-2\eta)\mathop{\mathbb{E}}_{\mathrm{EX}(c,\mathcal{D})}[g(x)\cdot y]$$

i.e. expectation under $\mathrm{EX}^\eta(c,\mathcal{D}) = (1-2\eta)$ times expectation under $\mathrm{EX}(c,\mathcal{D})$
To estimate expectation of 2nd term under $\mathrm{EX}(c,\mathcal{D})$ within $\frac{\tau}{2}$
Suffices to estimate its expectation under $\mathrm{EX}^\eta(c,\mathcal{D})$ within $\frac{\tau}{2}(1-2\eta)$
and dividing this latter estimate by $1-2\eta$
Since $g(x)y = (\varphi(x,1)-\varphi(x,-1))y/2$ takes a value between $-1/2$ and $1/2$
With prob $\geqslant 1-\delta/2M$, can estimate $\mathbb{E}_{\mathrm{EX}^\eta(c,\mathcal{D})}[g(x)y]$ within $\frac{\tau}{2}(1-2\eta)$
using $O\left(\frac{1}{\tau^2(1-2\eta)^2}\ln\frac{M}{\delta}\right)$ samples (Hoeffding)
A makes M queries, by union bound, with prob $\geqslant 1-\delta$, all estimates $\hat{P}_\varphi$ are within $\pm\tau$ of P_φ □

---

2. Guessing noise rate

So far we assumed learning algorithm knows true noise rate η exactly (unrealistic assumption)
Above proof suggests that knowing an approximate value η' of η is enough

Algorithm pretends noise rate is η' (and suppose $1-\frac{\tau}{2} \leqslant \frac{1-2\eta}{1-2\eta'} \leqslant 1+\frac{\tau}{2}$)
It wants to estimate $\mathbb{E}_{\mathrm{EX}(c,\mathcal{D})}[g(x)y]$, but cannot do so directly
It will first estimate $\mathbb{E}_{\mathrm{EX}^\eta(c,\mathcal{D})}[g(x)y]$ (call this expectation P_η) within $\frac{\tau}{4}(1-2\eta')$
Denote algorithm's estimate by $\hat{P}_\eta$
Algorithm then divides $\hat{P}_\eta$ by $1-2\eta'$ to get an estimate for $\mathbb{E}_{\mathrm{EX}(c,\mathcal{D})}[g(x)y] = \frac{1}{1-2\eta}P_\eta$

$$\left|\frac{1}{1-2\eta'}\hat{P}_\eta - \mathop{\mathbb{E}}_{\mathrm{EX}(c,\mathcal{D})}[g(x)y]\right| = \left|\frac{1}{1-2\eta'}\hat{P}_\eta - \frac{1}{1-2\eta'}P_\eta + \frac{1}{1-2\eta'}P_\eta - \frac{1}{1-2\eta}P_\eta\right|$$
$$\leqslant \frac{1}{1-2\eta'}\left|\hat{P}_\eta - P_\eta\right| + |P_\eta|\left|\frac{1}{1-2\eta'} - \frac{1}{1-2\eta}\right|$$

1st term is at most $\frac{1}{1-2\eta'}\frac{\tau}{4}(1-2\eta') = \frac{\tau}{4}$

2nd term is at most

$$|P_\eta| \left| \frac{1}{1-2\eta'} - \frac{1}{1-2\eta} \right| = \left| \frac{1}{1-2\eta} P_\eta \right| \left| \frac{1-2\eta}{1-2\eta'} - 1 \right| \leqslant \left| \mathop{\mathbb{E}}_{\mathrm{EX}(c,\mathcal{D})}[g(x)y] \right| \frac{\tau}{2} \leqslant \frac{1}{2}\frac{\tau}{2} = \frac{\tau}{4}$$

Last inequality due to $g(x)y = (\varphi(x,1) - \varphi(x,-1))y/2$ taking a value between $-1/2$ and $1/2$
So algorithm's actual estimate will be within $\frac{\tau}{2}$ of $\mathbb{E}_{\mathrm{EX}(c,\mathcal{D})}[g(x)y]$ with high prob

---

What if only an upper bound η_* to the true noise rate η is known? $(0 \leqslant \eta \leqslant \eta_* < 1/2)$
Algorithm can try noise rates $\eta_1, \eta_2, \ldots, \eta_k$ such that
$1 - 2\eta_j = \left(1 - \frac{\tau}{2}\right)^{j-1} \left(1 + \frac{\tau}{2}\right)^{-j}$ for $1 \leqslant j < k$ and $\eta_k \geqslant \eta_*$
One of these noise rates, say η_ℓ, will satisfy $1 - \frac{\tau}{2} \leqslant \frac{1-2\eta}{1-2\eta_\ell} \leqslant 1 + \frac{\tau}{2}$ (Exercise)
Algorithm gets hypotheses $h_1, \ldots, h_k$ from different noise rates $\eta_1, \ldots, \eta_k$
Hypothesis h_ℓ corresponding to η_ℓ (that is close to η) will have $\mathrm{err}_{\mathcal{D}}(h_\ell, c) \leqslant \varepsilon$ with high prob

How can algorithm find out which h_j is good?
Ideally, feed samples to h_j and estimate $\mathrm{err}_{\mathcal{D}}(h_j, c)$
But algorithm can only access noisy samples from $\mathrm{EX}^\eta(c,\mathcal{D})$, not clean samples from $\mathrm{EX}(c,\mathcal{D})$
Observation: $\mathbb{P}_{\mathrm{EX}^\eta(c,\mathcal{D})}[h(x) \neq y] = \mathrm{err}_{\mathcal{D}}(h,c)(1-2\eta) + \eta$
Reason: If $\varepsilon = \mathrm{err}_{\mathcal{D}}(h,c) = \mathbb{P}_{\mathrm{EX}(c,\mathcal{D})}[h(x) \neq y]$, then
$\mathbb{P}_{\mathrm{EX}^\eta(c,\mathcal{D})}[h(x) \neq y] = (1-\eta)\varepsilon + \eta(1-\varepsilon) = \varepsilon(1-2\eta) + \eta$
Transformation $\varepsilon \mapsto \varepsilon(1-2\eta) + \eta$ mapping $\mathrm{err}_{\mathcal{D}}(h,c)$ to $\mathbb{P}_{\mathrm{EX}^\eta(c,\mathcal{D})}[h(x) \neq y]$ is monotone
Thus hypothesis h_j minimizing $\mathbb{P}_{\mathrm{EX}^\eta(c,\mathcal{D})}[h(x) \neq y]$ will also minimize $\mathrm{err}_{\mathcal{D}}(h_j, c)$

How many noise rates (and hypotheses) to try?
Since $1 - 2\eta_k = \left(1 - \frac{\tau}{2}\right)^{k-1} \left(1 + \frac{\tau}{2}\right)^{-k}$, we want $\left(1 - \frac{\tau}{2}\right)^{k} \left(1 + \frac{\tau}{2}\right)^{-k} \leqslant \left(1 - \frac{\tau}{2}\right)(1 - 2\eta_*)$
so $k = \left(\ln \frac{1}{1-2\eta_*} - \ln\left(1 - \frac{\tau}{2}\right)\right) / \ln\left(\left(1 + \frac{\tau}{2}\right) / \left(1 - \frac{\tau}{2}\right)\right) = O\left(\frac{1}{\tau} \ln \frac{1}{1-2\eta_*}\right)$
because $\left(1 + \frac{\tau}{2}\right) / \left(1 - \frac{\tau}{2}\right) = 1 + \Theta(\tau)$ for small $\tau > 0$ and $\ln(1+y) = \Theta(y)$ for small $y > 0$