سوال ۱ –

بخش آ) میدانیم در الگوریتم AdaBoost در هر مرحله یک weak learner بر اساس وزن داده ها آموزش داده میشود و تمرکز آن بر دسته بندی درست داده های با وزن بیشتر است(در مرحله اول تمام داده ها وزن یکسان دارند لذا تعداد داده هایی که misclassified میشوند مهم است و وزن داده ها مطرح نیست). سپس هر مدل یک ضریب(وزن) میگیرد(مبنی بر اینکه چقدر خوب داده هارا دسته بندی کرده که منور از داده ها، داده ها با احتساب وزن هایشان است یعنی داده های وزن بالا مهم تر هستند) و در هر مرحله مدل نهایی میشود میانگین وزن دار weak learner های بدست آمده تا آن مرحله بوسیله تابع sign . اگر رابطه آپدیت وزن های داده هارا بنویسیم:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} e^{-\alpha_t} & if\ y_i = h_t(i) \\ e^{\alpha_t} & if\ y_i \neq h_t(i) \end{cases}$$

همانطور که میبینیم، در هر iteration ، وزن داده ها مبنی بر اینکه توسط weak learner فعلی درست دسته بندی شده اند یا نه، آپدیت میشود و به این ترتیب، weak learner مرحله بعدی وظیفه دارد جوری آموزش داده شود که ضعف های weak learner قبلی را پوشش دهد. بنابراین بدلیل آپدیت شدن وزن های داده ها، weak learner بعدی نمیتواند دقیقا مانند weak learner قبلی عمل کند(یکسان باشند) زیرا وزن داده ها برایش متفاوت است و مجبور است تغییر کند تا بتواند ضعف های قبلی را پوشش دهد. بنابراین همیشه $h_t \neq h_{t+1}$.

بخش ب) یک weak learner در دوره(iteration) t سعی میکند h_t ای را پیدا کند که دقت وزن دار زیر را ماکسیمم کند:

$$P_{i\sim D_t}[y_i = h_t(x_i)] = \sum_{i=1}^{m} D_t(i) y_i h_t(x_i)$$

در الگوریتم *AdaBoost* سعی میشود یک توزیع D_t ساخته شود که به وسیله آن این دقت وزن دار(معادل همبستگی وزن دار) به ازای تمام *weak learner* ها کمینه شود.بنابراین در هر راندt ، یک توزیع جدید D_{t+1} تولید میشود که همبستگی با h_t صفر شود زیرا:

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

$$\rightarrow \sum_{i=1}^{m} D_{t+1}(i) y_i h_t(x_i) = \frac{1}{Z_t} \sum_{i=1}^{m} D_t(i)\, e^{-\alpha_t y_i h_t(x_i)} y_i h_t(x_i)$$

$$= -\frac{1}{Z_t} \frac{dZ_t}{d\alpha_t} = 0 \rightarrow \sum_{i=1}^{m} D_{t+1}(i) y_i h_t(x_i) = 0$$

منبع: MITPress

سوال ۲ –

بخش آ) میتوانیم به کمک روش های MAP/MLE بیاییم μ_i, σ_i را تخمین بزنیم با فرض اینکه

$$P(x_i|y_i) \sim N(\mu_i, \sigma_i) \qquad for\ i \in \{1,2,\dots,n\}$$

مزایا: تعداد کمی training sample مورد نیاز است زیرا تعداد feature ها محدود است.

عیب: فرض محکمی روی توزیع داده ها گذاشتیم که ممکن است صادق نباشد و از طرفی اگر واقعا شکل توزیع برایمان مهم باشد و بخواهیم آن را بدست آوریم و فرض اولیه ای نداشته باشیم از داده ها، این روش ها به ما نمیتوانند کمک کنند.

بخش ب) داریم:

$$E[\hat{p}_n(x)] = P(X_i \in I_l) * H = H \int_{\frac{l-1}{H}}^{\frac{l}{H}} p(v)dv = M\left(F\left(\frac{l}{H}\right) - F\left(\frac{l-1}{H}\right)\right)$$

$$= \frac{F\left(\frac{l}{H}\right) - F\left(\frac{l-1}{H}\right)}{\frac{1}{H}} = \frac{F\left(\frac{l}{H}\right) - F\left(\frac{l-1}{H}\right)}{\frac{l}{H} - \frac{l-1}{H}} = F'(x^*) = p(x^*), \qquad x^* \in \left[\frac{l-1}{H}, \frac{l}{H}\right]$$

طبق تئوری مقدار میانگین، یک نقطه دیگر به نام x^{**} وجود دارد که بین x و x^* قرار دارد که:

$$\frac{p(x^*) - p(x)}{x^* - x} = p'(x^{**})$$

بنابراین بایاس برابر است با:

$$bias\left(\hat{p}_n(x)\right) = E[\hat{p}_n(x)] - p(x) = p(x^*) - p(x)$$

$$= p'(x^{**}) * (x^* - x) \leq |p'(x^{**})||(x^* - x)| \leq \frac{\beta}{H}$$

توجه شود نامساوی وسطی نامساوی کوشی شوارتز است.در نامساوی آخر هم از این حقیقت استفاده کردیم که هم x هم x^* بین I_l هستند که طول اش 1/H است.بنابراین اندازه تفریق آن ها کمتر مساوی 1/H است.

منبع: اسلاید های دانشگاه واشنگتن از yen chi-chen

بخش پ) نتیجه میگیریم هرچه تعداد تعداد bin بیشتری داشته باشیم، هیستوگرام، بایاس به همان نسبت کمتر خواهد داشت زیرا طبق گفته دکتر رهبان سر کلاس، bins بیشتر یعنی رزولوشن بهتر در تخمین یعنی تخمین دقیق تر توزیع داده ها.

بخش ت) روش هیستوگرام یک روش non-parametric است که بر خلاف اسمش، تعداد خیلی خیلی زیادی پارامتر دارد اما بر خلاف دانسته های قبلی مان، این تعداد پارامتر زیاد باعث overfitting نمیشود زیرا به کمک روش هایی مثل رویکرد بیزین در روش های non-parametric ، جلوی overfitting را میگیریم.

البته در روش هیستوگرام که ساده ترین روش non-parametric است، رخدادی مانند overfitting رخ میدهد و آن هم زمانی است که تعداد bins زیادی داشته باشیم که در این حالت بایاس کم و واریانس زیاد میشود.

یکی دیگر از مشکلات این روش، این است که بخاطر تعداد پارامتر زیاد، اینگونه روش ها نیازمند داده های آموزشی زیادی هستند. مشکل دیگر این است که اگر عرض یک bin کوچک باشد، تخمین نویزی میشود و حساسیت به مقدار هر داده آموزشی در توزیع داده ها، زیاد میشود. مشکل دیگر این روش، این است که اگر عرض bins کوچک نباشد، شکل توزیع پله ای میشود که این چیز خوبی نیست.

سوال ۳ –

بخش آ) همانطور که میبینیم، مدل نهایی برابر است با علامتِ مجموع برچسب های پیش بینی شده توسط weak learner ها. حال سوال شده است چه موقعی این مدل دچار خطا میشود. به نظر من در دو حالت این مدل اشتباه میکند:

۱- وقتی تعداد weak learner هایی که برچسب مثبت را پیش بینی میکنند با آن هایی که برچسب منفی پیش بینی میکنند برابر شود، مجموع آن ها صفر شده و مدل نهایی، علامتِ عدد صفر یعنی برچسب مثبت را پیش بینی میکند. در صورتی که ممکن است برچسب واقعی داده، منفی باشد.

۲- وقتی تعداد weak learner هایی که پیش بینی اشتباه میکنند بیشتر از آن هایی باشد که پیش بینی درست میکنند که در این حالت مدل نهایی برچسب را اشتباه انتخاب میکند.

بخش ب) چون در این حالت ما یک رای گیری ساده بین weak learner ها داریم،احتمال اینکه مدل نهایی اشتباه بکند، برابر است با تمام حالت هایی که تعداد weak learner هایی که اشتباه میکنند، بزرگتر تعداد weak learner هایی باشد که درست تشخیص میدهند، بنابراین اگر فرض کنیم احتمال تشخیص درست یک weak learner را p در نظر بگیریم و $H(T)$ یعنی تعداد weak learner هایی که درست تشخیص میدهند بین T تا weak learner ، ،مسئله تبدیل به یک مسئله برنولی میشود و داریم:

$$P\left(H(T) \leq \frac{1}{2}T\right) = \sum_{i=0}^{\frac{1}{2}T} p^i(1-p)^i$$

طبق نامساوی هافدینگ برای توزیع برنولی داریم:

$$P\left(H(T) \leq \left(\frac{1}{2} - \varepsilon\right)T\right) \leq \exp\left(-2\varepsilon^2 T\right)$$

یعنی احتمال اینکه خطایی بوجود بیاید، یعنی کمتر از نیمی از weak learner ها درست تشخیص بدهند، حد بالای آن $\exp(-2\varepsilon^2 T)$ است.برای مثال اگر ۵ تا weak learner داشته باشیم، حد بالای بوجود آمدن خطا در مدل نهایی برابر است با :

$$\exp\left(-10\varepsilon^2\right)$$

حال اگر تعداد weak learner ها به سمت بینهایت میل کند، داریم:

$$P(H(T) \leq \infty) \leq \exp\left(-2\varepsilon^2 * \infty\right) \approx 0$$

یعنی احتمال بوجود آمدن خطا با داشتن بینهایت weak learner ، تقریبا صفر است و مدل نهایی قطعا درست کار میکند.