

ظرفیت کانال های کلاسیک

۱ مقدمه

در درس قبل دیدیم که می توان پیام های یک منبع را که از کلمات $X = \{x_1, x_2, \dots, x_{2^n}\}$ تشکیل شده است را می توان با کدگذاری مناسب یعنی کدگذاری بلوکی چنان مخابره کرد که بجای n بیت به ازای هر کلمه، $nH(X) \leq n$ بیت مخابره کنیم. این نتایج البته در حد n های بزرگ دقیق هستند. تمام این حرف ها در صورتی بود که از نوفه داخل کانال صرف نظر کنیم و فرض کنیم که پیام بدون هیچ نوع تغییری به مقصد می رسد. مطالب بالا محتوی قضیه شانون را درباره کدگذاری بدون نوفه تشکیل می دهند. در این درس می خواهیم درباره کدگذاری در حضور نوفه بحث کنیم و نهایتاً قضیه شانون درباره کدگذاری در حضور نوفه را بیان کنیم. در این درس هم چنین با مفهوم مهم ظرفیت کانال آشنا خواهیم شد و در حالت های خیلی ساده ای مقدار آن را محاسبه می کنیم.

۲ موازنه بین پایین آوردن خطا و بالا بردن نرخ انتقال اطلاعات

از این به بعد فرض می کنیم که کلمات منبع خود را به طریق بهینه ای کدگذاری کرده ایم و تنها می خواهیم کد کلمه ها را که رشته هایی از 0 و 1 هستند مخابره کنیم. فرض می کنیم که کانالی که پیام های خود را از طریق آن ارسال می کنیم تحت تاثیر نوفه قرار می گیرد و هر علامت 0 یا 1 که ارسال می کنیم با احتمال q تبدیل به 1 یا 0 شده و با احتمال $1 - q$ دست نخورده باقی می ماند. می خواهیم کاری کنیم که گیرنده پیام ها همچنان بتواند پیام های صحیح را از پیام های دریافت شده استخراج کند.

۱.۲ یک تذکره مهم

آنچه که در مثال بالا گفته شد یک نوع کد موسوم به کد تکرار سه تایی برای تشخیص و تصحیح خطاست و با کدگذاری کلمات فرق می کند. از این به بعد در این درس هر وقت که ما از کلمه کدگذاری استفاده می کنیم منظورمان این نوع کدگذاری است. به عبارت دیگر فرض می کنیم که کلمات منبع X که می توانند حروف یک زبان مثل فارسی یا انگلیسی یا علائم ریاضی باشند به نحو بهینه ای تبدیل به رشته های 0 و 1 شده اند و هدف ما مخابره این رشته ها از درون کانال است. در انتهای کانال رشته های دریافت شده نخست برای تعیین و تصحیح خطاها بررسی شده و پس از اصلاح به کلمات زبان مورد نظر بازگشایی می شوند.

راهی که برای تعیین و تصحیح خطاها باید پیش گرفت آن است عنصر تکرار را به نوعی وارد پیام های خود کنیم. به عنوان مثال می توانیم بجای مخابره یک 0 سه تا 0 و بجای یک 1 سه تا 1 مخابره کنیم و از گیرنده بخواهیم که با استفاده از قانون اکثریت تصمیم بگیرد که یک رشته سه تایی که دریافت کرده است در واقع چه رشته ای بوده است (جدول ??).

0	0 0 0
1	1 1 1

کد تکرار سه تایی: 1:

در واقع احتمال خطا که قبلاً برابر بود با q اینک کمتر شده است، زیرا احتمال وقوع خطا اینک برابر است با احتمال وقوع دو برگردان و یا سه برگردان در کد سه تایی که اولی برابر است با $3q^2(1-q)$ و دومی برابر است با q^3 . در نتیجه احتمال وقوع خطا برابر خواهد بود با $q^3 + 3q^2(1-q)$ که برای q های کوچک از مرتبه q^2 است. البته بهایی برای این کاهش خطا پرداخت کرده ایم و آن این است که نرخ مخابره اطلاعات را پایین آورده ایم و از سه بیت برای مخابره یک بیت استفاده کرده ایم. در این حالت نرخ مخابره اطلاعات یعنی R برابر است با $\frac{1}{3}$. در حالت کلی اگر از n بیت برای مخابره 2^k پیام استفاده کنیم (که در نبود نوفه می توانست برای مخابره 2^n پیام مورد استفاده قرار گیرد) می گوئیم که نرخ مخابره اطلاعات برابر است

$$R := \frac{k}{n}. \quad (1)$$

مسلم است که در این نوع کدگذاری می توان باز هم احتمال خطا را کاهش داد و در حد ایده آل آن را به صفر نزدیک کرد ولی در این صورت نرخ R نیز به سمت صفر میل خواهد کرد. حال سوالی که با آن مواجه هستیم آن است که آیا اصولاً می توان از یک کانال نوفه دار با نرخ محدود و غیر صفر اطلاعاتی را عبور داد به نحوی که احتمال وقوع خطا لا اقل در حد مجانبی (برای رشته های بزرگ) به سمت صفر میل کند؟ آیا یک نوع کدگذاری وجود دارد که این امکان را برای ما فراهم آورد؟ آیا نرخ بیشینه ای وجود دارد که هر نوع مخابره اطلاعات با بیش از آن نرخ بدون خطا غیر ممکن باشد؟ این ها سوالاتی است که در قضیه دوم شانون پاسخ داده شده است. قبل از بیان قضیه کلی سعی می کنیم که دو مثال ساده را بررسی کنیم.

مثال ۱: حالت ساده ای را در نظر می گیریم که منبع X پیام هایی را ارسال می کند که در آنها احتمال وقوع 0 و 1 یکسان باشد. به عبارت دیگر آنتروپی منبع در این مثال برابر است با 1. فرض کنید که پیام خود را در رشته های n بیتی ارسال کنیم. مجموعه تمام رشته های n بیتی دارای 2^n رشته است. نقاط این فضا نقاط یک شبکه ابرمکعبی n بعدی را پر کرده اند. اگر همه این نقاط را به عنوان کد کلمه های خود به کاربریم به آسانی در طول عبور از کانال یک کد کلمه به یکی از کد کلمه های همسایه اش یا کد کلمه های نزدیکش تبدیل می شود و گیرنده واقعاً نمی فهمد که چه کد کلمه ای برایش ارسال شده است. بنابراین راه مقابله با خطا آن است که سعی کنیم که کد کلمه ها را با فاصله کافی از یکدیگر انتخاب کنیم تا احتمال وقوع خطا پایین بیاید. در واقع وقتی که یک کد کلمه را ارسال می کنیم حول آن یک کره به اندازه کافی بزرگ رسم می کنیم و هر وقت نقطه ای درون این کره دریافت کنیم آن را به عنوان کد کلمه ای که در مرکز آن قرار دارد تعبیر و خطاهای بوجود آمده را تصحیح می کنیم. این کار به طور طبیعی نرخ را پایین می آورد زیرا همه نقاط شبکه استفاده نمی کنیم. حال می پرسیم که کد کلمه ها را با چه فاصله ای انتخاب کنیم. در این جا می بایست بین دو عامل متضاد موازنه ایجاد کنیم. اگر بخواهیم خطا را پایین بیاوریم می بایست فاصله کد کلمه ها را از یکدیگر زیاد کنیم. از طرفی این کار نرخ را پایین می آورد. خوب بیایید ببینیم به طور متوسط یک کد کلمه به چند

کلمه نزدیک دیگر ممکن است تبدیل شود؟ در یک کلمه n حرفی هر حرف با احتمال q ممکن است که برگردانده شود. بنابراین تعداد حرف های برگردانده شده به طور متوسط برابر است با nq . این تعداد حرف می تواند به $2^{nH(q)} \approx \binom{n}{nq}$ طریق در رشته n حرفی قرار گیرند. در نتیجه تعداد رشته های نزدیک به این کلمه که ممکن است از خطاهای بوجود آمده در این کلمه ایجاد شوند برابر است با $2^{nH(q)}$. در نتیجه یک نحوه کد گذاری خوب آن است که حول هر کلمه ناحیه ای در نظر بگیریم که به طور متوسط تعداد کلمات داخل آن در حدود $2^{nH(q)}$ باشد. حال اگر نرخ مبادله برابر با $R = \frac{k}{n}$ باشد معنایش این است که تعداد کل کد کلمه هابرابر است با $2^k = 2^{nR}$. چون تعداد کل کلمات ممکن برابر است با 2^n به این نتیجه می رسیم که می بایست شرط زیر برقرار باشد:

$$2^{nR} 2^{nH} \leq 2^n, \quad (2)$$

یا

$$R \leq 1 - H(q) =: C. \quad (3)$$

کمیت $C := 1 - H(q)$ ظرفیت این کانال نامیده می شود و این رابطه بیان می کند که برای پرهیز از خطا نرخ مبادله اطلاعات حتما باید از ظرفیت کانال کمتر باشد. حال بایک سوال اساسی مواجه می شویم و آن اینکه آیا واقعا نوعی کد گذاری وجود دارد که بتوان با استفاده از آن نرخ مبادله اطلاعات را در حد مجانبی $n \rightarrow \infty$ به حداکثر ممکن یعنی به C رساند؟ پاسخ این سوال مثبت است. در واقع شانون نشان داده است که کد تصادفی این کار را می تواند انجام دهد. البته کد تصادفی هوشمندانه ترین نحوه کد کردن کلمات نیست ولی نکته جالب آن است که این کد می تواند حد بالای نرخ مبادله را اشباع کند. تعریف کد تصادفی به این صورت است که در فضای تمام کد کلمه های ممکن که شامل 2^n تا نقطه است، $2^k \equiv 2^{nR}$ نقطه به طور تصادفی انتخاب می کنیم. حول هر کدام از این نقاط کره ای رسم می کنیم که تعداد $2^{n(H(q)+\delta)}$ تا نقطه داشته باشد. حال همان نحو کد گشایی را که در بالا توضیح دادیم بکار می بریم.

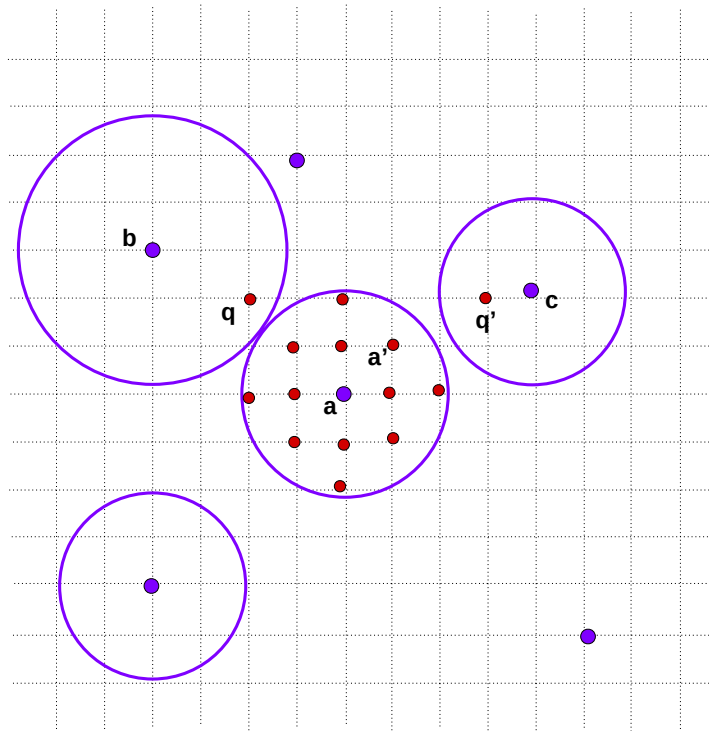
یک کد کلمه و کره اطراف آن را در نظر بگیرید. این کره کسر

$$f = \frac{2^{n(H(q)+\delta)}}{2^n} = 2^{-n(C-\delta)} \quad (4)$$

از نقاط را در بر دارد. اگر کد کلمه واقع در مرکز این کره در طول انتقال به نقطه ای درون همین کره تبدیل شده باشد، خطا محسوب نمی شود. خطا وقتی بوجود می آید که کد کلمه های کرات دیگر درون این کره بیفتند. احتمال این اتفاق برابر است با f . چون در کل 2^{nR} کلمه داریم احتمال وقوع خطا برابر می شود با:

$$P = f 2^{nR} = 2^{-n(C-R-\delta)} \quad (5)$$

حال چون مجازیم δ را هر مقدار مثبتی بگیریم به این نتیجه می رسیم که می توانیم R را هر چقدر که می خواهیم به C نزدیک کنیم و P همچنان در حد $n \rightarrow \infty$ به سمت صفر میل می کند.



شکل ۱: هرگاه کد کلمه a دچار خطای کوچک شود و تبدیل به a' شود خطای بوجود آمده قابل تصحیح است. خطا وقتی بوجود می آید که a تبدیل به نقاط درون دیگر کره ها شود.

۳ تعریف ظرفیت در حالت کلی

در بخش قبل خود را به منبع پیامی محدود کردیم که دارای آنتروپی بیشینه بود زیرا در این منبع حرف 0 و حرف 1 هر دو با احتمال $\frac{1}{2}$ رخ می دادند و بنابراین داشتیم $H(X) = H(\frac{1}{2}) = 1$. در این بخش حالت کلی را در نظر می گیریم که در آن آنتروپی منبع دلخواه است. فرض می کنیم که منبع حرف 0 را با احتمال p و حرف 1 را با احتمال $1-p$ تولید کند. در نتیجه آنتروپی منبع برابر است با

$$H(X) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}. \quad (6)$$

حال می خواهیم ببینیم که پیام های این منبع را حداکثر با چه نرخ می توانیم مخابره کنیم. فرض کنید که رشته $y = (y_1, y_2, \dots, y_n)$ را دریافت کنیم. فرض اساسی ما آن است که حروف کلمه های ارسالی و هم چنین خطاهای ایجاد شده در آنها هیچ نوع همبستگی بایکدیگر ندارند. بنابراین احتمال وقوع 0 در همه مکان ها برابر با p است. تعداد رشته های متعارف x ای که می توانسته اند در اثر خطا منجر به این رشته y شوند برابر است با $2^{nH(X|y)}$. بنابراین حول این y می توانیم کره ای رسم کنیم که تعداد $2^{nH(X|y)}$ نقطه داشته باشد. به طور متوسط کره هایی که حول نقاط مختلف رسم می کنیم دارای $2^{nH(X|Y)}$ نقطه

هستند. اگر تعداد کل نقاط را $2^k = 2^{nR}$ بگیریم می بایست داشته باشیم

$$2^{nH(X|Y)} \times 2^{nR} \leq 2^{nH(X)} \quad (7)$$

و در نتیجه

$$R \leq H(X) - H(X|Y). \quad (8)$$

بنابراین ظرفیت کانال رامی توانیم به شکل زیر تعریف کنیم:

$$C = \max_X I(X; Y) = \max_X (H(X) - H(X|Y)). \quad (9)$$

و در نتیجه نشان دادیم که امکان مخابره اطلاعات با نرخ R که کمتر از ظرفیت کانال باشد امکان پذیر است. به همان ترتیب که در مورد مثال اول عمل کردیم در این حالت کلی نیز می توان استدلال کرد که نرخ خطا را می توان در حد n های بزرگ به سمت صفر میل داد. یک رشته معین یعنی یک نقطه معین در شبکه را در نظر بگیریم. تعداد نقاط درون کره ای که دور این نقطه رسم کرده ایم برابر با $2^{n(H(X|Y)+\delta)}$. این نقاط کسر f از کل نقاط را تشکیل می دهد که در آن

$$f = \frac{2^{n(H(X|Y)+\delta)}}{2^{nH(X)}} = 2^{-n(H(X)-H(X|Y)-\delta)} \quad (10)$$

حال احتمال اینکه یکی از نقاط دیگر به اشتباه درون این کره بیفتد و مادر بازگشایی این رشته مرتکب خطا شویم برابر است با f . احتمال وقوع خطا در بازگشایی کل رشته ها برابر است با f ضرب در تعداد کل رشته ها یعنی

$$P = f \times 2^{nR} = 2^{-n(H(X)-H(X|Y)-\delta)} \times 2^{nR} = 2^{-n(C-R-\delta)}. \quad (11)$$

حال می توانیم هر قدر که R را به C نزدیک کنیم و δ را چنان اختیار کنیم که همچنان $C - R - \delta > 0$ باشد و در نتیجه P در حد n های بزرگ به سمت صفر میل کند. به عبارت بهتر به ازای هر $\epsilon > 0$ و هر $\delta > 0$ می توان n را آنقدر بزرگ گرفت که هر دو شرط زیر برقرار باشند:

$$R < C - \delta, \quad P < \epsilon, \quad (12)$$

و این نشان دهنده این است که می توانیم هر قدر که بخواهیم نرخ مخابره را به C نزدیک کرده و در عین حال خطا را هر قدر که بخواهیم کوچک کنیم. در این جانکته ای مطرح است و آن اینکه خطای محاسبه شده در واقع یک خطای متوسط است یعنی خطای بوجود آمده برای تمام کد کلمه ها متوسط گیری شده است. در واقع اگر خطای ایجاد شده برای کد کلمه i را با P_i نشان دهیم آنچه که نشان داده ایم آن است که

$$\frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} P_i < \epsilon. \quad (13)$$

می توان با حذف کردن بعضی از کد کلمه ها و در واقع خلوت کردن فضای شبکه ابرمکعبی از نقاط فاصله بین کد کلمه ها را زیادتر کرد طوری که خطای ایجاد شده برای هر کد کلمه از یک مقداری نهایت کوچک تر شود. می خواهیم ببینیم که چه مقدار از کلمات را باید دور بریزیم و این کار چقدر نرخ را پایین خواهد آورد. فرض کنید که تعداد کل کد کلمه هایی که خطای آنها از 2ϵ بیشتر باشد را با $N_{2\epsilon}$ نشان دهیم. در این صورت نامساوی بالا نشان می دهد که

$$\frac{1}{2^{nR}} N_{2\epsilon} 2\epsilon < \epsilon, \quad (14)$$

که از آن نتیجه می گیریم

$$N_{2\epsilon} < 2^{nR-1}. \quad (15)$$

بنابراین حداکثر می بایست به تعداد 2^{nR-1} یعنی نصف کد کلمه ها را دور ریخت تا فاصله بین کد کلمه ها آنقدر زیاد شود که خطای هر کد کلمه به زیر 2ϵ رسد. در این صورت نرخ تبدیل می شود به:

$$R' = \frac{k'}{n} = \frac{nR-1}{n} = R - \frac{1}{n}, \quad (16)$$

که در حد n های بزرگ دوباره با R برابر است.

آنچه که در این فصل نشان دادیم محتوی قضیه دوم شانون بود یعنی می توان از یک کانال نوفه دار برای مخابره بدون خطای اطلاعات استفاده کرد مشروط بر اینکه نرخ مخابره را زیر یک حد که ظرفیت کانال نامیده می شود نگاه داشت. می توان هر قدر که بخواهیم نرخ را به ظرفیت کانال نزدیک کرد و همچنان خطا را کوچک نگاه داشت و برای این کار می بایست از کدهایی با طول بلندتر استفاده کرد.