الگوریتم EM برای داده های از دست رفته
(EXPECTATION–MAXIMIZATION)

استاد راهنما : دکتر وحید رضایی تبار

ارائه دهنده : حسام افشار

فهرست مطالب

- مقدمه
- انواع داده های از دست رفته
- الگوریتم EM و نحوه عملکرد آن
- مثال

مقدمه

بر اساس یک نظرسنجی علم داده که در سال 2020 توسط Anaconda صورت گرفت، متخصصان داده 45 درصد از زمان خود را صرف آماده سازی داده ها می کنند. یکی از مشکلات رایج که همه دانشمندان داده در این 45 درصد از زمان در دسترس خود با آن مقابله می کنند، رسیدگی به داده های از دست رفته در مجموعه داده خود است. و با این حال، هیچ راه حل یکسانی برای رسیدگی به این مشکل وجود ندارد.

داده‌های از دست رفته یا مقادیر از دست رفته زمانی رخ می‌دهد که داده‌ای برای متغیرها یا شرکت‌کنندگان خاصی ذخیره نکنید. داده ها ممکن است به دلیل ورود ناقص داده ها، خرابی تجهیزات، فایل های از دست رفته و بسیاری دلایل دیگر از بین بروند و معمولا در مجموعه داده های بزرگ داده های از دست رفته وجود دارند.

انواع داده های از دست رفته

دلیل وجود داده های از دست رفته باید در نظر گرفته شود، زیرا به شما کمک می کند تا نوع داده های از دست رفته و اقداماتی که باید در مورد آن انجام دهید را تعیین کنید. در حالت کلی سه نوع اصلی از داده های از دست رفته وجود دارد که در جدول زیر آمده است:

انواع داده از دست رفته	مفهوم
به طور کاملاً تصادفی (MCAR)	داده های از دست رفته به طور تصادفی در بین متغیرها توزیع می شوند و با سایر متغیرها ارتباطی ندارند و هیچ ساختار خاصی برای توصیف این کمبود وجود ندارد.
به طور تصادفی (MAR)	داده های از دست رفته به طور تصادفی توزیع نمی شوند، اما آنها توسط سایر متغیرهای مشاهده شده محاسبه می شوند.
به طور غیر تصادفی (MNAR)	داده های از دست رفته به طور سیستماتیک با مقادیر مشاهده شده متفاوت است.

انواع داده های از دست رفته

حذف، انتساب میانه یا میانگین، انتساب چندگانه یا حداکثر درستنمایی برخی از رایج ترین روش ها برای رسیدگی به داده های از دست رفته هستند.اما آن‌ها فقط برای مجموعه‌های داده‌ای قابل اجرا هستند که MCAR هستند و کاملاً MNAR نیستند.

الگوریتم EM و نحوه عملکرد آن

در موارد خاص، ما بیشتر علاقه مند خواهیم بود که بدانیم آیا الگو یا دلیلی برای داده های از دست رفته وجود دارد یا خیر و با مراقبت از آن، آن را نسبت دهیم.

در چنین مواردی، روش هایی مانند الگوریتم EM مناسب تر است.

با این فرض که توزیع مشترک داده های گمشده و داده های مشاهده شده معلوم است، هدف الگوریتم EM یافتن تخمین پارامتری است که لگاریتم احتمال داده های مشاهده شده را به حداکثر می رساند، یعنی تابع چگالی احتمال مشاهدات و از آنجا مقادیر از دست رفته را تخمین می زند.

این الگوریتم دارای دو مرحله است: مرحله expectation (E-step) و مرحله maximization (M-step) که به صورت تکرار شونده انجام می شوند.

الگوریتم EM و نحوه عملکرد آن

مراحل اجرای این الگوریتم عبارتند از:

1) با توجه به مجموعه داده ناقص یک سری پارامتر اولیه در نظر میگیریم.
2) (E-step): با استفاده از داده های موجود، مشاهدات گمشده را برآورد میکنیم.
3) (M-step): با استفاده از داده های کامل تولید شده در مرحله 2، پارامترهای مدل را به حداکثر میرسانیم.
4) مرحله 2 و 3 را تا زمان همگرایی تکرار میکنیم، یعنی برآورد پارامتر بین تکرارها تغییر زیادی نکند.

یکی از مزایای اصلی الگوریتم EM این است که بایاس پارامترهای برآورد شده بسیار کمتر است.

الگوریتم EM و نحوه عملکرد آن

فرض کنید که مشاهدات $x_1, x_2,..., x_n$ را با d نمایش دهیم ومتغیرهای پنهان $h_1, h_2,..., h_n$ را با h و همهٔ پارامترهای توزیع را نیز با θ نمایش دهیم. در این صورت لگاریتم درست نمایی کل داده ها (پنهان و نمایان = مشاهدات) برابر خواهد بود با:

$$l(\theta)=\log p(d|\theta)= \log \sum_h p(d,h|\theta)$$

از آنجا که لگاریتم تابع اکیداً صعودی است، می‌توان لگاریتم درست نمایی کل داده‌ها را نسبت به θ بیشینه کرد. ولی آرگومان لگاریتم یک مجموع است و نمی‌توان به سادگی پاسخ تحلیلی برای θ افت. از این رو، الگوریتم ب-ا ترفندی را برای بیشینه کردن حد *پایین* لگاریتم درست نمایی بکار می‌برد. این حد پایین از نابرابری ینسن بدست می‌آید.

الگوریتم EM و نحوه عملکرد آن

بر اساس نابرابری ینسن برای هر دسته k تایی از t_i ها و w_i ها اگر $\sum w_i = 1,\ t_i>0$ خواهیم داشت:

$$\sum_{i=1}^{k} w_i \log\ t_i \leq \log \sum_{i=1}^{k} w_i\ t_i$$

اکنون $J(\theta)$ را به صورت زیر باز نویسی میکنیم:

$$\log\ \sum_h q(h) \frac{p(d,h|\theta)}{q(h)} \geq \sum_h q(h) \log \frac{p(d,h|\theta)}{q(h)} = J(q,\theta)$$

که با گزینش $q(h)=p(h|d,\theta)$ داریم:

الگوریتم EM و نحوه عملکرد آن

$$\log p(d|\theta) = \sum_h p(h|d,\theta^{(t)}) \log p(d,h|\theta) - \sum_h p(h|d,\theta^{(t)}) \log p(h|d,\theta)$$

$$= Q(\theta|\theta^{(t)}) + H(\theta|\theta^{(t)})$$

$$\log p(d|\theta^{(t)}) = \sum_h p(h|d,\theta^{(t)}) \log p\left(d,h\middle|\theta^{(t)}\right) - \sum_h p(h|d,\theta^{(t)}) \log p(h|d,\theta^{(t)})$$

$$= Q(\theta^{(t)}|\theta^{(t)}) + H(\theta^{(t)}|\theta^{(t)})$$

$$\log p(d|\theta) - \log p(d|\theta^{(t)}) = Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)}) + H(\theta|\theta^{(t)}) - H(\theta^{(t)}|\theta^{(t)})$$

الگوریتم EM و نحوه عملکرد آن

که با توجه به نابرابری گیبز($-\sum_{i=1}^{n} p_i \log p_i \leq -\sum_{i=1}^{n} p_i \log q_i$) داریم $H(\theta|\theta^{(t)}) \geq H(\theta^{(t)}|\theta^{(t)})$ در نتیجه میتوان نوشت:

$$\log p(d|\theta) - \log p(d|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)})$$

در واقع یعنی انتخاب θ به گونه ای که $Q(\theta|\theta^{(t)})$ را بهبود دهد باعث میشود $\log p(d|\theta)$ نیز حداقل همان اندازه بهبود یابد

$$Q(\theta|\theta^{(t)}) = E_{h|d,\theta}[\log L(\theta|d,h)] = \sum_h p(h|d,\theta^{(t)}) \log p(d,h|\theta)$$

$$= \sum_h q(h) \log p(d,h|\theta)$$

الگوریتم EM و نحوه عملکرد آن

در نتیجه روش کار الگوریتم امید ریاضی-بیشینه کردن به صورت زیر است:

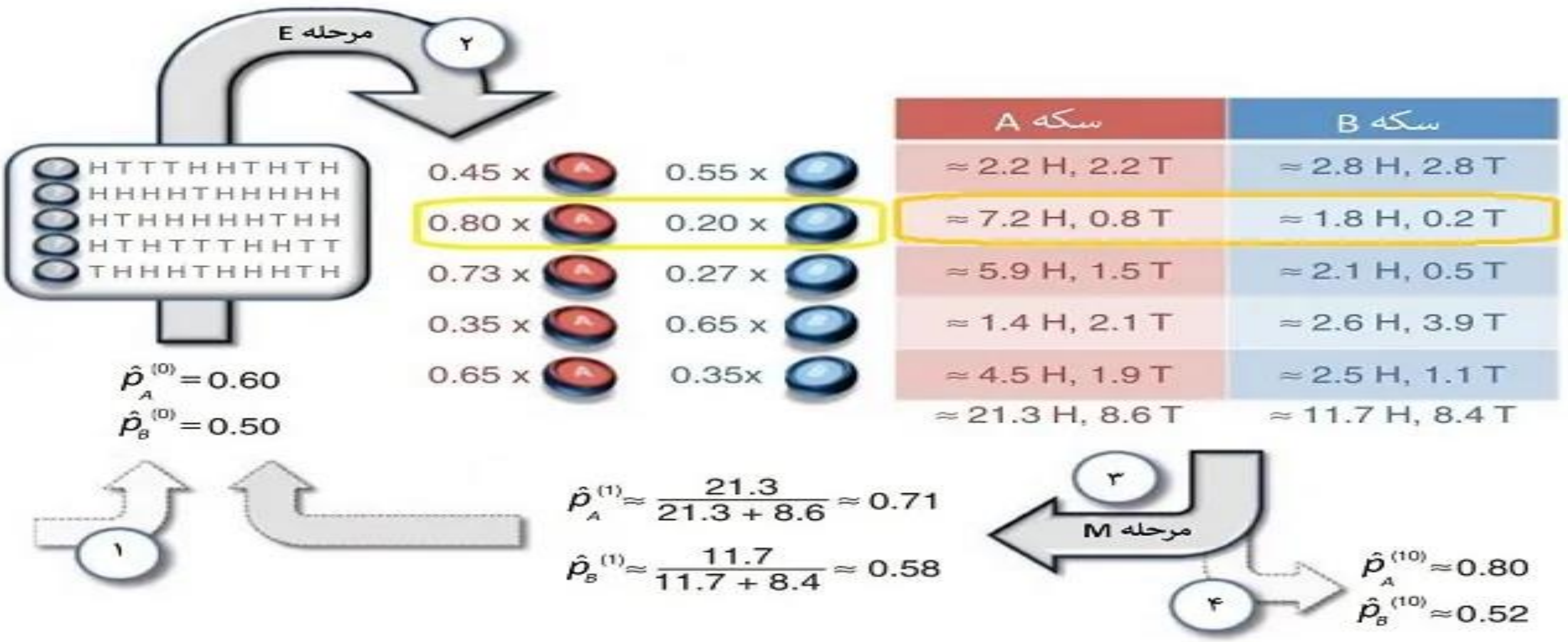
1- پارامترها را مقدار آغازین $\theta^{(0)}$ میدهیم.

2-تا رسیدن به همگرایی دو گام زیر را انجام میدهیم:

- گام امید ریاضی(E-step): $q^{(t)} = \underset{q}{armax}\, Q(\theta|\theta^{(t)})$

- گام بیشینه کردن (M-step): $\theta^{(t)} = \underset{\theta}{armax}\, Q(\theta|\theta^{(t)})$

مثال



مرحله E
۲
H T T T H H T H T H
H H H H T H H H H H
H T H H H H H T H H
H T H T T T H H T T
T H H H T H H H T H
0.45 x A 0.55 x B
0.80 x A 0.20 x B
0.73 x A 0.27 x B
0.35 x A 0.65 x B
0.65 x A 0.35x B
سکه A
سکه B
≈ 2.2 H, 2.2 T
≈ 2.8 H, 2.8 T
≈ 7.2 H, 0.8 T
≈ 1.8 H, 0.2 T
≈ 5.9 H, 1.5 T
≈ 2.1 H, 0.5 T
≈ 1.4 H, 2.1 T
≈ 2.6 H, 3.9 T
≈ 4.5 H, 1.9 T
≈ 2.5 H, 1.1 T
≈ 21.3 H, 8.6 T
≈ 11.7 H, 8.4 T
$\hat{p}_A^{(0)} = 0.60$
$\hat{p}_B^{(0)} = 0.50$
۱
۳
مرحله M
$\hat{p}_A^{(1)} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71$
$\hat{p}_B^{(1)} \approx \frac{11.7}{11.7 + 8.4} \approx 0.58$
۴
$\hat{p}_A^{(10)} \approx 0.80$
$\hat{p}_B^{(10)} \approx 0.52$