بسم الله الرحمن الرحيم





دانشکده مهندسی برق و کامپیوتر

بازیابی هوشمند اطلاعات - تمرین دوم سید مهدی رضوی استاد: خانم دکتر شاکری

آبان ماه ۱۴۰۲



																						ب	طال	، م	ست	ہر،	فر
٣																						ی	، عمل	اول	مرين	ت	١
۶																						لى	م عم	ن د و	نمرين	j	۲
٧																					(يحى	، تشر	اول	مرين	ت	٣
٨																					(يحى	، تشر	دو•	مرين	ت	۴
١.																					ن	ريح	م تش	سو	مرين	ت	۵
																						رير	صاو	، ت	ست	ہر،	فر
٣																	A	dditi	ve,	JM	I S	mo	othi	ng		١	
۴													ی	بازي	وارس	همو	ی ه	وشها:	ش رو	ِ زمای	ی آ	ي برا	كريپت	اساً		٢	
۴																		I	Oric	hlet	t S	mo	othi	ng		٣	
۵																		Т	wos	Step	S	mo	othi	ng		۴	
۶																			كيبي	يتم تر	گوري	ي الأ	دەساز	پیاه		۵	
۶																										۶	



۱ تمرین اول عملی

							mahdi@l	MahdiRazavi:	~/Documer	ts/bazyabi/H	w2-RUNS/	'additiveSmoothing				-	. o 🔞
	w Search Terminal						004		000			0.004					
217 0.000		000		000	0.0		.001		.026		. 000	0.001	0.026	0.000	0.001	0.026	0 0
00	0.001 0.001	0.026	.026	0.000	.000	0.001 0.001	0.0	0.026	0.0	0.000		0.001	0.026	0.000	0.001	0.026	0.0
218		0.020 000		000	0.0		.000		.000		.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000		.000		.000	0.000	. 000	0.000		0.000	. 000	0.000	0.000	0.000	0.000	0.000	0.0
00	0.000	0.000		0.000		0.000	0.0		0.0			0.000	0.000	0.000	0.000	0.000	0.0
219		001		028	0.0		.000		.000		.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000		.000		.000	0.000		0.000		0.000		0.000	0.000	0.000	0.000	0.000	0.0
00	0.000	0.000		0.000		0.000	0.0		0.0								•.•
223	0.	000		000	0.0		.000		.000		.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000	0	.000	0	.000	0.000		0.000		0.000		0.000	0.000	0.000	0.000	0.000	0.0
00	0.000	0.000		0.000		0.000	0.0	00	0.0	90							
224		063		160	0.1		. 000		.000		.000	0.000	0.000	0.000	0.000	0.000	
0.000	0.000		.000		.000	0.000		0.000		0.000		0.000	0.000	0.000	0.000	0.000	0.0
00	0.000	0.000		0.000		0.001	0.0		0.0								
225		003		048	0.0		.001		.024		. 000	0.001	0.024	0.000	0.001	0.024	
0.000 00	0.001 0.001	0 0.024	.024	0 0.000	.000	0.001 0.001	0.0	0.024	0.0	0.000		0.001	0.024	0.000	0.001	0.024	0.0
run-id	ahdi/Docume	map		ndcg	UNC /- 4	p10		A				0.007	0.048	0.0	10		
	andi/Docume ahdi/Docume											008	0.048	0.017	12		
	ahdi/Docume											009	0.042	0.017			
	ahdi/Docume											009	0.042	0.015			
	ahdi/Docume											009	0.041	0.013			
	ahdi/Docume											009	0.041	0.016			
	ahdi/Docume											009	0.041	0.015			
	ahdi/Docume											009	0.040	0.013			
	ahdi/Docume										0.	009	0.040	0.012			
/home/m	ahdi/Docume	nts/ba	zyabi	/HW2-R	UNS/ad	ditiveSmo	othin	g/runJ	MS-9.	txt	Θ.	008	0.039	0.013			
	t: randomiz ahdiRazavi:						itive:		ing\$ [
(i)	🔭 🚺 📒 addītiveSm	oothing 🧐	، راهنمایی	درخواست	CA2.pdf	6-slm (files merg.	🖺 galag	o-3.16 – Diri.	🎯 additir	ve.sh — -/	D 🚞 galago-3.16	mahdi@Mahd	Raz 🞒 IR-HW2 – fbM	ixtur 🕜 EN en 🔄	🎑 🛕 🦃 🜒 Thu N	lov 9, 18:23

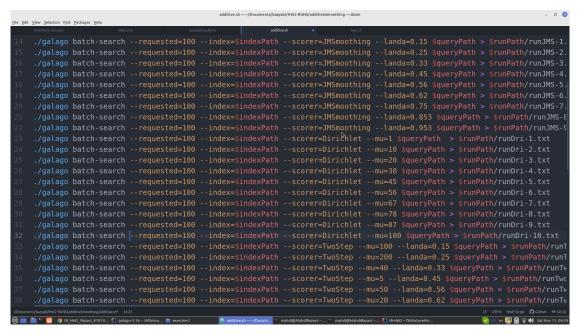
شكل ۱- Additive , JM Smoothing

از آنجایی که Additive Smoothing هیچ وابستگی به متغیرهای لاندا و مو ندارد ، آن را فقط یکبار آزمایش خواهیم کرد. درباره مقایسه روش هموارسازی Additive در مقایسه با JM Smoothing نمیتوان کاملا به قطعیت نظرداد چون دچار اختلاف بین پارامترها هستیم.

برای سهولت امر تست پارامترهای مختلف ، اسکریپت آن را نوشتهام. پارامترهای متناظر مو و لاندا را با توجه به بازههای آنها بررسی خواهیم کرد.

همانطور که از پارامترها مشاهده میشود ، در JM Smoothing تغییر محسوسی در میزان دقت و سایر پارامترهای متناظر ایجاد نشده است ، اما با نگاهی ریزبینانه تر متوجه خواهیم شد که با افزایش میزان متغیر لاندا ، میزان دقت تابع امتیازدهی به اسناد رو به کاهش میباشد.





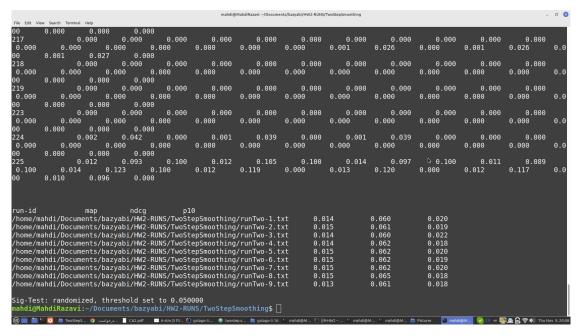
شکل ۲: اسکریپتی برای آزمایش روشهای هموارسازی



شکل ۳: Drichlet Smoothing

در این روش هموارسازی ، همانطور که مشخص است بهترین اجرای ما مربوط به اجرای ۶ می باشد. با توجه به این که مقدار متغیر مو در اینجا برابر ۵۶ می باشد و ما در این سری آزمایش ها کلا بازه انتخابی برای متغیر مو را بین ۱ تا ۱۰۰ در نظرگرفته ایم ، مو اینطور به نظر می رسد که در حالت وسط نتیجه متغیرهای مربوط به دقت بازیابی بهتر خواهد بود.





شکل ۴: TwoStep Smoothing

با توجه به تصاویر بالا بهترین حالت اجرا مربوط به اجرای هشتم می شود. متغیر لاندا برابر 0.853 و متغیر مو برابر ۳۰ می باشد. این نتایج با توجه به حالت های مختلف به دست آمده است.



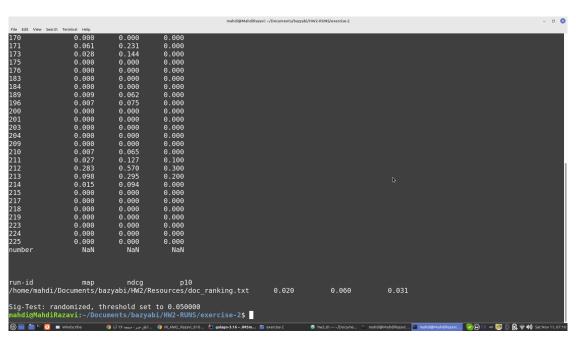
۲ تمرین دوم عملی

```
| The lett your gargest gots perfutur gold in n 100 Vot yindow party or make your gots perfuture and in no 100 perfuture and i
```

شكل ۵: پيادهسازي الگوريتم تركيبي

ما الگوریتم ذکرشده در اسلاید درس مبنی بر EM Computation را پیادهسازی کردهایم. از ساختمان داده HashMap برای ذخیره احتمال مورد انتظار و بیشینه احتمال به ازای هر کویری ترم میپردازیم. همانطور که در تصویر ارزیابی مشخص است ، نتایج این الگوریتم به طرز معناداری بهتر از روش های هموارسازی های تمرین اول است.

با اختلاف بیشتر از 0.008 این الگوریتم با حدود ۲۰ بار تکرار به این صورت عمل خواهد کرد. به نظر می رسد هر بار تکرار به بیشینه کردن احتمالات تعلق و یا عدم تعلق کلمات مرتبط و نامرتبط کمک خواهد کرد.



شكل 6: ارزيابي كد بالا



۳ تمرین اول تشریحی



۴ تمرین دوم تشریحی

$$OKAPI - TF = \sum_{t \in q} \frac{(k_1 + 1) * t f_{td}}{k + t f_{td}}$$

$$BM - 25 = \sum_{t \in q} \log \frac{(r_1 + 0.5)/(R - r_i + 0.5)}{(n - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} * \frac{(k_1 + 1)f_i}{K + f_i} * \frac{(k_2 + 1)qf_i}{K_2 + qf_i}$$

Relevance Documents to queries status

D3	$\mathbf{D2}$	D1	Relevance
NO	YES	YES	Q1
NO	YES	NO	$\mathbf{Q2}$
NO	YES	NO	$\mathbf{Q3}$

متغیر b تاثیر طول سند را کنترل میکند. متغیر k تاثیر تکرار عبارتها را کنترل میکند.

 $\mathrm{Okapi}(\mathrm{Q1}\ ,\,\mathrm{D1})=1.3422$

 $Okapi(Q1\ ,\, D2)=0.9174$

 $\mathrm{Okapi}(\mathrm{Q1}\;,\,\mathrm{D3})=0$

Okapi(Q2, D1) = 1.3422

Okapi(Q2 , D2) = 1.8265

Okapi(Q2, D3) = 0.88105

 $\mathrm{Okapi}(\mathrm{Q3}\;,\,\mathrm{D1}) = 2.22333$

Okapi(Q3, D2) = 2.74395

Okapi(Q3, D3) = 1.76211



BM-25(Q1 , D1) = -1.4746 BM-25(Q1 , D2) = -1.0079 BM-25(Q1 , D3) = 0 BM-25(Q2 , D1) = 2.9492 BM-25(Q2 , D2) = 2.006639 BM-25(Q2 , D3) = 0.967940 BM-25(Q3 , D1) = 1.474647 BM-25(Q3 , D2) = 3.01454 BM-25(Q3 , D3) = 1.9358

Rankings Based on BM-25 Scoring

Q1: (1) D3, (2) D2, (3) D1 Q2: (1) D1, (2) D2, (3) D3 Q3: (1) D2, (2) D3, (3) D1

Okapi-TF و EM-25 هر دو الگوریتمهای رتبهبندی و امتیازدهی هستند که در سیستمهای IR استفاده می شوند. الگوریتم Okapi-TF یک الگوریتم رتبهبندی است که فراوانی ترم ها را در سند بررسی می کند و با فرض این که هر چه یک کوئری ترم در سند بیشتر باشد ، آن ترم با سند مرتبطتر است.

ی می از رویکردی BM-25(Best Matching 25) همچنین یک الگوریتم رتبهبندی است که علاوه بر فراوانی کوئری ترمها ، از رویکردی احتمالی برای رتبهبندی اسناد استفاده می کند. (از جمله محاسبه IDF)

این امر باعث میشود که الگوریتم ${
m BM-25}$ در امتیازدهی موثرتر واقع شود.



۵ تمرین سوم تشریحی

بخش اول

$$P(neural|D1) = 0.7 * \frac{1}{16} + 0.3 * \frac{1}{1000} = 0.04405$$

Dirichlet Prior

$$P(W|d) = \frac{|d|}{|d| + \mu} \frac{c(w,d)}{|d|} + \frac{\mu}{|d| + \mu} P(W|C)$$
$$P(SVM|C) = \frac{1}{33}$$

بخش دوم

$$P(SVM|D2) = \frac{17}{17 + 1500} * \frac{1}{17} + \frac{1500}{17 + 1500} * \frac{1}{33} = 0.03062264$$

بخش سوم

JM Smoothing

P(machine learning models | D1) = P(machine | D1) * P(learning | D1) * P(models | D1) = 0.000475085

P(machine learning models | D2) = P(machine | D2) * P(learning | D2) * P(models | D2) = 0.000354225

Dirichlet Smoothing

P(machine learning models | D1) = P(machine | D1) * P(learning | D1) * P(models | D1) = 0.000445664

P(machine learning models | D2) = P(machine | D2) * P(learning | D2) * P(models | D2) = 0.000442364 + 0.0004424 + 0.0004424 + 0.00044 + 0.000444 + 0.00044 + 0.000444 + 0

برای هموار ساز Dirichlet سند شماره ۱ با احتمال بالاتری بازگردانده خواهد شد. (البته اختلاف میزان احتمال سند۱ خیلی قابل توجه به نسبت سند ۲ نیست.) برای هموار ساز JM سند شماره ۱ با احتمال بالاتری بازگردانده خواهد شد. (این بار با اختلاف محسوستری این سند بازگردانده خواهد شد.) خواهد شد.)



از برتریهای JM نسبت به Dirichlet در مرتبسازی اسناد ممکن است شامل موارد زیر باشد: (همواره این اتفاق نمیافتد و مورد به مورد ممکن است متفاوت باشد)

- ۱. تعمیمپذیری: یکی از برتریهای JM نسبت به Dirichlet این است که این روش قابلیت تعمیمپذیری بیشتری دارد. به عبارت دیگر JM قادر است بهتر با تغییرات در مجموعه اسناد و یا مجموعه کلمات موجود در اسناد ، سازگاری یابد. احساس میکنم بیشتر به خاطر این که متغیر لاندا بین صفر و یک نوسان میکند راحت تر میتوان آن را تغمیم داد.
- ۲. انعطاف پذیری: JM انعطاف بیشتری در تنظیم پارامترهای خود دارد که این امکان را به ما میدهد که بهتر بتوانیم آن را به وضعیت خاص دادهها و مساله مورد نظرمان تظبیق بدهیم.
 - ٣. استفاده از اطلاعات پیشین
- ۴. عملکرد بهتر در موارد پراکنده : در شرایطی که داده ها پراکنده باشد ، این روش ممکن است عملکرد بهتری داشته باشد و بهبود قابل توجهی نسبت به Dirichlet داشته باشد .