

Сравнение результатов вычислений моего кода и известных мне статей (анализ тестов)

Владимир Ивашкин

9 апреля 2018 г.

Введение

Захотелось написать большой развернутый отчет о всех тестах, которые я провожу. Возможно, это поможет найти наши ошибки и в будущем быть уверенными в результатах.

1 Chebotarev: Studying new classes of graph metrics

Ссылка: <https://arxiv.org/abs/1305.7514>

Здесь нам интересен Fig. 1. На графе "цепочка" можно прогнать такие же метрики при тех же параметрах. В обозначениях ниже я имею в виду, что вершины графа названы слева направо цифрами от 1 до 4. Расстояния здесь нормированы на то, чтобы сумма $D_{12} + D_{23} + D_{34} = 3$. Достаточно будет сравнивать расстояния $D_{12}, D_{23}, D_{13}, D_{14}$. Будем считать, что расстояния не соответствуют друг другу, если хотя бы одна соответствующая пара расстояний различается на 0.04 в абсолютной величине.

Вначале результаты не сходились, но потом выяснилось следующее:

- В моем коде из всех ядер, перед тем, как превращать их в расстояния, брался корень. Кажется, мы обсуждали, что это нужно для Communicability, но в итоге это было включено везде. В этом причина, почему тесты не проходили
- Для Communicability это все-таки нужно, в этом случае все результаты совпадают

		D_{12}	D_{23}	D_{13}	D_{14}
SP	true	1,000	1,000	2,000	3,000
	test	1,000	1,000	2,000	3,000
R	true	1,000	1,000	2,000	3,000
	test	1,000	1,000	2,000	3,000
Walk	true	1,025	0,950	1,975	3,000
	test	1,025	0,950	1,975	3,000
logFor	true	0,959	1,081	2,040	3,000
	test	0,959	1,081	2,041	3,000
For	true	1,026	0,947	1,500	1,895
	test	1,026	0,947	1,500	1,895
SqResistance	true	1,000	1,000	1,414	1,732
	test	1,000	1,000	1,414	1,732
Comm	true	0,964	1,072	1,492	1,564
	test	0,964	1,072	1,492	1,564
pWalk 4.5	true	1,025	0,950	1,541	1,466
	test	1,025	0,950	1,541	1,466
pWalk 1.0	true	0,988	1,025	1,379	1,416
	test	0,988	1,025	1,379	1,416

Таблица 1: Тест Studying new classes of graph metrics, Figure 1

Также я воспроизвел результаты из Table 1 в Chebotarev: The Walk Distances in Graphs (ссылка: <https://arxiv.org/abs/1103.2059>). Скорее всего, они основаны на тех же результатах, что уже были в таблице выше, но почему бы нет.

		D_{12}/D_{23}	$(D_{12} + D_{23})/D_{13}$	D_{14}/D_{12}
SP	true	1,000	1,000	1,500
	test	1,000	1,000	1,500
R	true	1,000	1,000	1,500
	test	1,000	1,000	1,500
Walk	true	1,080	1,000	1,520
	test	1,080	1,000	1,519
logFor	true	0,890	1,000	1,470
	test	0,887	1,000	1,470
For	true	1,080	1,320	1,260
	test	1,083	1,316	1,263
pWalk 4.5	true	1,080	1,280	0,950
	test	1,079	1,281	0,951
pWalk 1.0	true	0,960	1,460	1,030
	test	0,964	1,459	1,027

Таблица 2: The Walk Distances in Graphs, Table 1

Видим, что с этими тестами все ок. В последнем разделе я привожу сводную таблицу, где показываю, что именно было покрыто повторением результатов каждой статьи.

2 Kivimaki: Developments in the theory of randomized shortest paths with a article comparison of graph node distances

Ссылка: <https://arxiv.org/abs/1212.1666>

Здесь мы можем использовать два источника: это Figure 2, а также Table 2 с оптимальными значениями из Table 1.

2.1 Figure 2

Здесь исследуется поведение метрик RSP, FE, pRes, logFor, SP-CT при изменении их параметров в заданном интервале для графа "треугольник с хвостом". Можно исследовать всю кривую, но проще всего взять только крайние точки: слева отношение Δ_{12}/Δ_{23} равно 1.5, справа — 1.0.

После того, как я убрал взятие корня для logFor, все результаты сошлись:

border	measure	param	D_{12}/D_{23}		
			test	true	diff
left	CT		1,5	1,5	0
	logFor	500.0	1,4975	1,5	0,0025
	RSP	0.0001	1,4992	1,5	0,0008
	FE	0.0001	1,4996	1,5	0,0004
right	SP		1	1	0
	logFor	0.01	1,0011	1	0,0011
	RSP	20.0	1	1	0
	FE	20.0	0,9834	1	0,0166

Таблица 3: Kivimaki, Figure 2

2.2 Table 2 с оптимальными значениями из Table 1

Здесь проверяется качество (метрика качества — NMI) кластеризации методом kMeans графов из датасета Newsgroups. Кernels: RSP, FE, logFor, SP-CT, SCT. Результаты получаются похожими для всех метрик, кроме SP-CT. Результат очень плох: в статье ожидается качество порядка 70-80 NMI*100, по факту что SP, что CT дают 0.2-3 NMI*100. SP-CT применяется с параметром 1, то есть чистый SP. Видим проблему с SP.

		n2cl1	n2cl2	n2cl3	n3cl1	n3cl2	n3cl3
RSP	test	79,443	57,914	81,070	77,092	76,797	75,520
	true	84,500	58,700	81,000	76,600	77,000	76,500
	diff	5,057	0,786	0,070	0,492	0,203	0,980
FE	test	79,443	57,917	81,070	76,619	77,980	75,131
	true	80,700	58,700	81,100	76,200	78,300	77,000
	diff	1,257	0,783	0,030	0,419	0,320	1,869
logFor H	test	81,846	60,952	76,988	78,376	75,010	75,121
	true	83,100	58,800	75,000	75,400	75,500	74,400
	diff	1,254	2,152	1,988	2,976	0,490	0,721
SP-CT K	test	0,219	0,147	0,201	0,315	0,334	0,295
	true	65,200	51,200	85,900	74,200	62,600	71,500
	diff	64,981	51,053	85,699	73,885	62,266	71,205
SCT H	test	81,105	54,616	78,440	77,922	72,276	75,409
	true	81,600	56,800	79,600	77,300	73,000	75,900
	diff	0,495	2,184	1,160	0,622	0,724	0,491

Таблица 4: Kivimaki, Table 2

Помимо статей я искал другие реализации мер для того, чтобы расширить количество тестов. Я наткнулся на вот эту реализацию: <https://github.com/jmmcd/GPDistance>. Здесь я увидел суть более сложные реализации RSP и FE. Насколько я понял, они защищены от случаев вроде тех, когда граф не связный. Я реализовал тесты из этого репозитория и увидел, что RSP и FE из этого репозитория работают стабильнее, чем мои варианты, сделанные строго по формулам из статей. Я заменил свои версии версиями из репозитория и они проходят тесты из статей. В частности, таблица выше содержит результаты с обновленными мерами.

3 Sommer: Comparison of Graph Node Distances on Clustering Tasks

Ссылка: (не находил в открытых источниках)

Здесь нас интересует Table 3 с оптимальными значениями из Table 2. Метрики: CCT, FE, logFor, RSP, SCT, SP Датасеты: football, newsgroups, polblogs, zachary Проблемы: CCT не работает для football, на polblogs не работает ничего, видимо из-за большого размера. Для SP не проходят никакие тесты.

		n2cl1	n2cl2	n2cl3	n3cl1	n3cl2	n3cl3	zachary	football
SCCT	test	0,794	0,598	0,758	0,784	0,758	0,746	1,000	error
	true	0,794	0,582	0,758	0,778	0,762	0,746	1,000	
	diff	0,000	0,016	0,000	0,006	0,004	0,000	0,000	
FE	test	0,797	0,645	0,811	0,781	0,763	0,764	1,000	0,862
	true	0,805	0,591	0,811	0,781	0,797	0,771	1,000	0,906
	diff	0,008	0,054	0,000	0,000	0,034	0,006	0,000	0,045
logFor	test	0,831	0,622	0,769	0,746	0,745	0,752	1,000	0,895
	true	0,838	0,584	0,748	0,753	0,758	0,749	1,000	0,903
	diff	0,007	0,038	0,021	0,007	0,014	0,003	0,000	0,008
RSP	test	0,797	0,635	0,785	0,781	0,786	0,725	1,000	0,895
	true	0,797	0,580	0,796	0,781	0,776	0,730	1,000	0,909
	diff	0,000	0,055	0,011	0,000	0,010	0,005	0,000	0,014
SCT	test	0,820	0,625	0,824	0,753	0,723	0,765	1,000	0,845
	true	0,817	0,552	0,786	0,773	0,728	0,763	1,000	0,811
	diff	0,002	0,073	0,039	0,020	0,005	0,002	0,000	0,033
SP	test	0,003	0,003	0,009	0,003	0,021	0,006	0,677	0,861
	true	0,654	0,516	0,859	0,743	0,625	0,720	1,000	0,858
	diff	0,651	0,513	0,850	0,740	0,603	0,714	0,323	0,004

Таблица 5: Sommer, Table 3

Похоже, все-таки у нас есть проблема.
Как работает SP:

- Вызываю `shortest_path` из `scipy` (визуально выдает правильные результаты)
- Применяется нормализация, чтобы параметр адекватно смешивал SP и CT
- Применяю $D \rightarrow K$ преобразование

Больше ничего тут нет. В оправдание функции `shortest_path` скажу, что мы проверяли ее тестами выше. Но проблемы с $D \rightarrow K$ тоже быть не может, ведь RSP и FE преобразуются этой же формулой. Виновата нормализация? Без нормализации наблюдаем ту же проблему. Что еще интересно, с уменьшением размеров графа качество кластеризации растет. Может, здесь даже проблема, как у Commute Time, описанная в Getting lost in space?

Я попробовал найти другие реализации shortest path — не помогло. Попробовал найти сразу shortest path kernel и нашел здесь: <https://github.com/gmum/pykernels>, но результат все такой же плохой.

Общий результат

Здесь я постараюсь записать в единый список всё, что мы используем в коде и наличие тестов.

	Subject	Chebotarev	Kivimaki	Sommer	Avrachenkov	Result
Measure	Shortest path	+	+/-	+/-		-
	Resistance	+	+			?
	SP-CT		-			-
	plain Walk	+				+
	Walk	+			+	+
	Forest	+				+
	logForest	+	+	+	+	+
	Comm	+				+
	logComm				+	+
	Heat					?
	logHeat				+	+
	SCT		+	+		+
	SCCT			+		+
	RSP		+	+		+
	FE		+	+		+
	Normalized Heat				+	+
	P. PageRank				+	+
	Modified P. PageRank				+	+
	Heat P. PageRank				+	+
Transformation	a -> t	+	+			+
	H0 -> H	+	+	+		+
	H -> D	+	+	+		+
Dataset	News		+	+		+
	Football			+		+
	Zachary			+		+
Graph Generator	Stochastic Block Model				+	+
	Clustering		+	+	+	+

Таблица 6: Overall result

Я не могу понять, что не так с простейшей метрикой. Может они используют не просто SP? В конце концов Kivimaki и Sommer могут иметь одну и ту же кодовую базу.