Воспроизведение результатов статьи в py_graphs.

Владимир Ивашкин

5 июля 2018 г.

2 Logarithmic vs. plain measures

Здесь RI, а не ARI

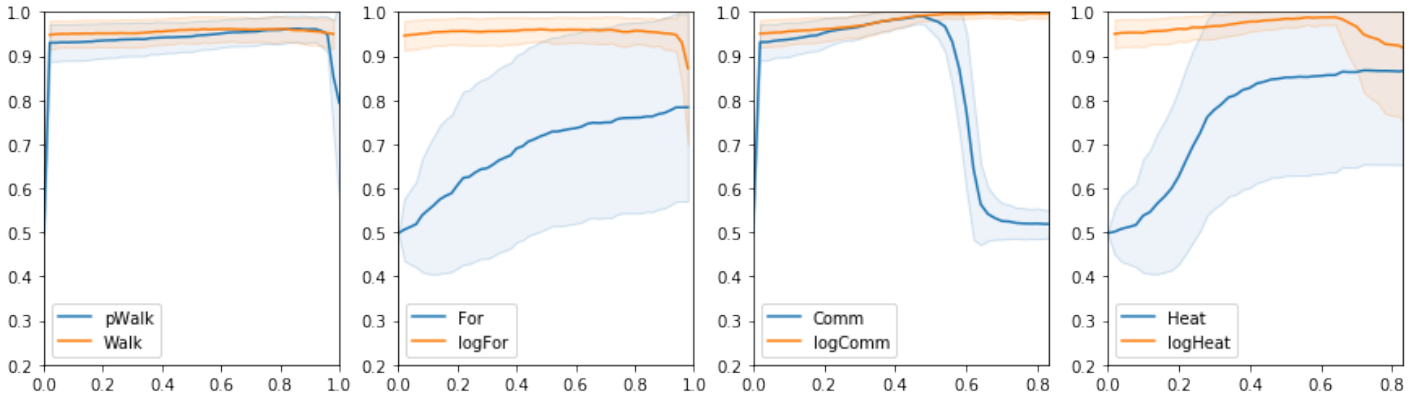


Рис. 1: Logarithmic vs. plain measures for $G(100, (2)0.2, 0.05)$

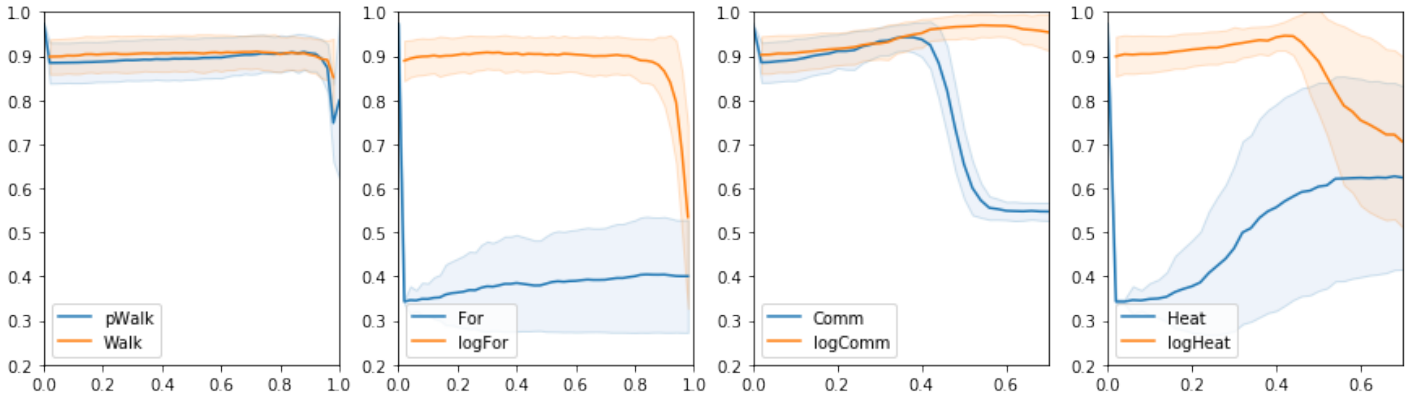


Рис. 2: Logarithmic vs. plain measures for $G(100, (3)0.3, 0.1)$

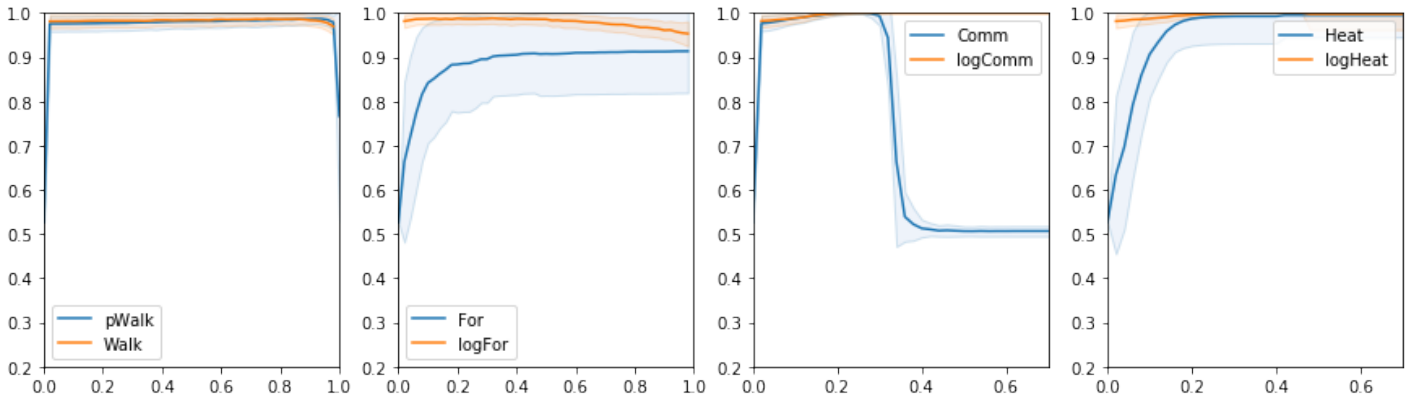


Рис. 3: Logarithmic vs. plain measures for $G(200, (2)0.3, 0.1)$

3 Competition by Copeland's score

Nodes	100	100	100	100	200	200	200	200	**Sum**
Classes	2	2	4	4	2	2	4	4	**of**
p_{out}	0.1	0.15	0.1	0.15	0.1	0.15	0.1	0.15	**scores**
logComm	383	547	476	-66	301	565	592		**2798**
Comm	249	150	308	418	291	212	325		**1953**
SCCT	316	299	166	44	293	392	412		**1922**
logHeat	308	314	180	-264	301	321	343		**1503**
pWalk	-81	26	56	418	-105	-155	6		**165**
SCT	-74	36	78	44	47	-24	44		**151**
Heat	221	-342	-456	418	295	205	-478		**−137**
RSP	-96	4	62	-272	-32	-85	-30		**−449**
Walk	-90	-26	78	-222	-149	-125	-26		**−560**
logFor	-92	-44	-24	-264	-63	-92	-32		**−611**
FE	-202	-64	-44	-224	-135	-169	-134		**−972**
For	-387	-566	-456	418	-525	-574	-478		**−2568**
SP-CT	-455	-334	-424	-448	-519	-471	-544		**−3195**

(a) optimal parameters

Nodes	100	100	100	100	200	200	200	200	**Sum**
Classes	2	2	4	4	2	2	4	4	**of**
p_{out}	0.1	0.15	0.1	0.15	0.1	0.15	0.1	0.15	**scores**
logComm	413	568	448	356	332	568	598	598	**3881**
SCCT	269	274	136	78	340	391	423	360	**2271**
logHeat	318	183	290	142	340	273	202	98	**1846**
Comm	168	151	222	172	286	258	333	178	**1768**
SCT	58	92	46	90	26	45	38	104	**499**
logFor	-114	60	56	110	-55	-115	4	88	**34**
Walk	-84	-10	132	86	-140	-85	30	66	**−5**
pWalk	-125	-40	54	74	-163	-79	-2	-14	**−295**
FE	-198	-27	-27	120	-120	-186	-66	32	**−472**
RSP	-151	-1	-8	78	-138	-179	-106	-16	**−521**
Heat	299	-341	-502	-490	340	154	-417	-515	**−1472**
SP-CT	-463	-345	-320	-228	-558	-462	-446	-396	**−3218**
For	-390	-564	-588	-588	-490	-583	-591	-583	**−4377**

(b) 90th percentiles

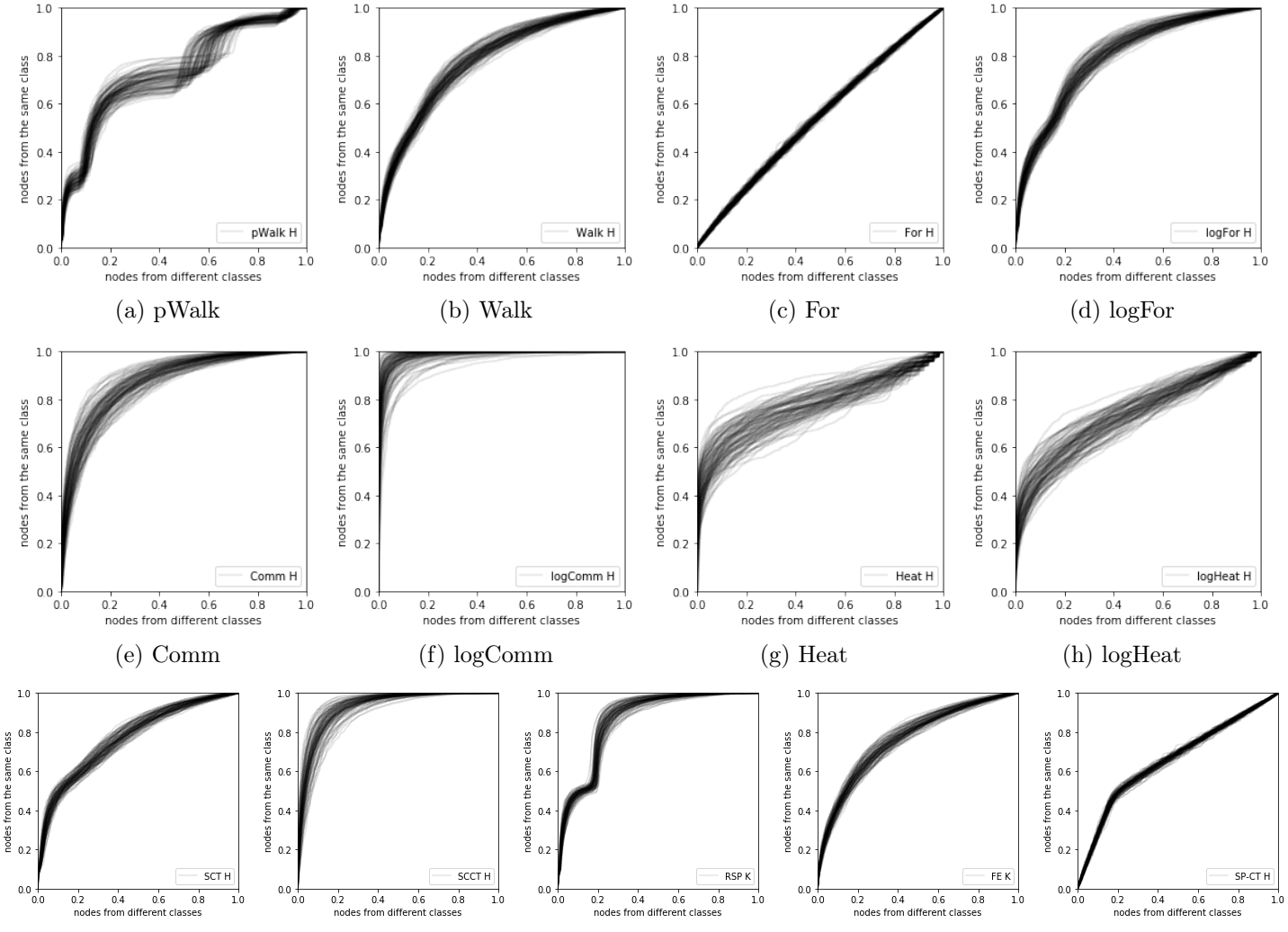
Таблица 1: Copeland's scores of the measure families on random graphs

4 Reject curves

Measure (kernel)	$G(100,(2)0.3,0.05)$ Opt. parameter, ARI	$G(100,(2)0.3,0.1)$ Opt. parameter, ARI	$G(100,(2)0.3,0.15)$ Opt. parameter, ARI
pWalk	0.93, 1.00	0.87, 0.91	0.73, 0.66
Walk	0.93, 1.00	0.67, 0.91	0.70, 0.65
For	0.60, 0.99	0.97, 0.51	0.40, 0.01
logFor	0.70, 1.00	0.40, 0.93	0.10, 0.68
Comm	0.33, 1.00	0.33, 0.98	0.30, 0.77
logComm	0.33, 1.00	0.47, **1.00**	0.57, **0.91**
Heat	0.37, 1.00	0.60, 0.87	0.73, 0.15
logHeat	0.37, 1.00	0.53, 0.99	0.37, 0.80
SCT	0.40, 1.00	0.57, 0.94	0.43, 0.72
SCCT	0.03, 1.00	0.57, 0.98	0.63, 0.80
RSP	0.97, 1.00	0.97, 0.93	0.97, 0.67
FE	0.90, 1.00	0.90, 0.91	0.87, 0.68
SP-CT	0.00, 0.99	0.03, 0.78	0.07, 0.49

Таблица 2: Optimal family parameters and the corresponding ARI's

Ошибка была в том, что подобранные параметры из таблицы выше принадлежат к диапазону $[0,1]$, а значит их нужно преобразовывать к диапазону, специфичному для конкретной метрики. Я же этого не делал. Вторая ошибка состояла в том, что я использовал тут близости вместо расстояний. Еще тогда, когда я строил их в прошлый раз, я заметил, что по близостям logComm совсем не обгоняет остальные меры, но по расстояниям эффект выраженный. Тут его тоже видно:

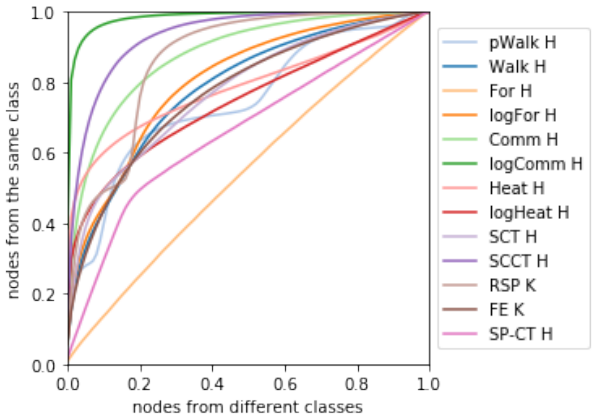


(a) pWalk (b) Walk (c) For (d) logFor

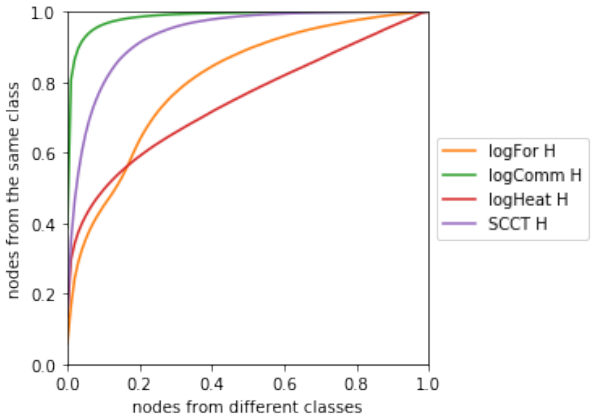
(e) Comm (f) logComm (g) Heat (h) logHeat

(i) SCT (j) SCCT (k) RSP (l) FE (m) SP-CT

Рис. 4: Reject curves for the graph measures under study



(a) All families



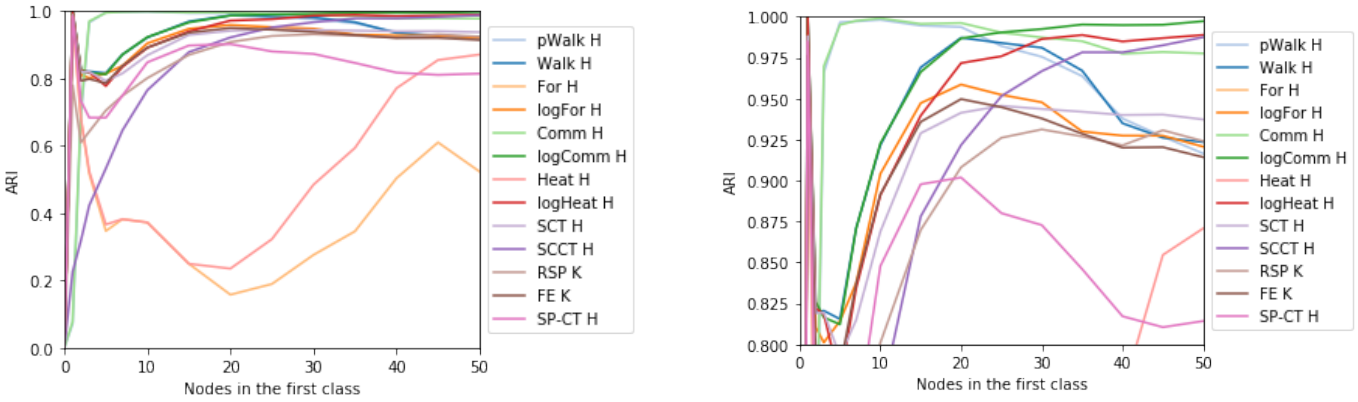
(a) All families

Рис. 5: Average reject curves

Здесь была проблема со взятием корня из Comm/logComm. А проблема была такая: если некоторые значения матрицы D при взятии корня превращаются в nan, то стандартная сортировка оставляет их на тех же позициях и отдельно сортирует массив слева и справа от них. Получается кусочно-возрастающая функция, из которой потом получаются несколько маленьких reject curve вместо одной большой. Решение – фильтровать эти nan и сортировать без них. Раз такой эффект вообще возник, значит в матрице D иногда появляются отрицательные значения.

Можно подозревать внешний вид графика pWalk. Может быть, это связано с тем, как мы фиксируем параметр. Параметром считаем отскалированое в [0, 1] число, для каждого графа преобразуем его в зависимости от спектрального радиуса матрицы A ($param = t/\rho(A)), t \in [0, 1]$).

5 Graphs with classes of different sizes



(a) All families

(b) Leading families

Рис. 6: Graphs with two classes of different sizes: clustering with optimal parameter values

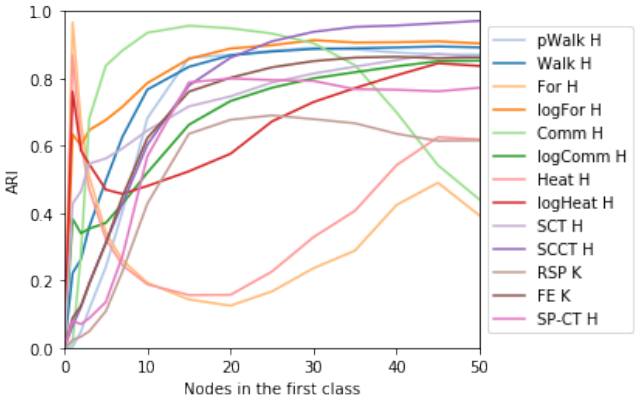
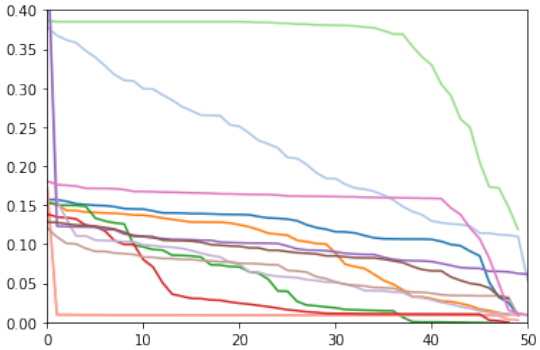


Рис. 7: Graphs with two classes of different sizes: random parameter values

$$P = \begin{pmatrix} 0.30 & 0.20 & 0.10 & 0.15 & 0.07 & 0.25 \\ 0.20 & 0.24 & 0.08 & 0.13 & 0.05 & 0.17 \\ 0.10 & 0.08 & 0.16 & 0.09 & 0.04 & 0.12 \\ 0.15 & 0.13 & 0.09 & 0.20 & 0.02 & 0.14 \\ 0.07 & 0.05 & 0.04 & 0.02 & 0.12 & 0.04 \\ 0.25 & 0.17 & 0.12 & 0.14 & 0.04 & 0.40 \end{pmatrix}.$$

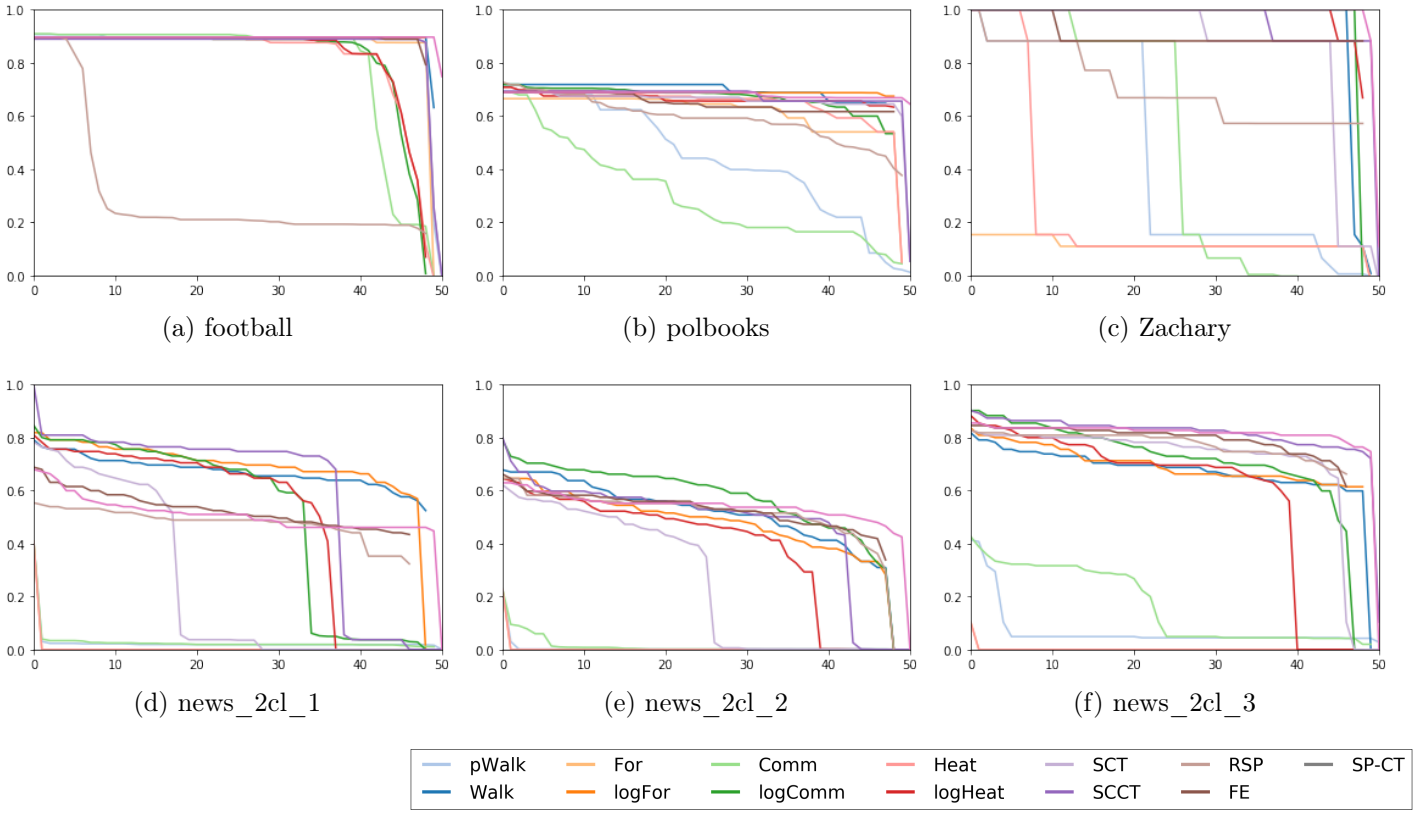


.45.45ARI

of various measure families on a structure with 6 classes

6 Cluster analysis on several classical datasets

Здесь ошибка была в том, что я зафиксировал число классов – 2, хотя в датасете football их 12. Теперь все похоже на статью:



(a) football (b) polbooks (c) Zachary

(d) news_2cl_1 (e) news_2cl_2 (f) news_2cl_3

Рис. 8: ARI of various measure families on classical datasets