PROCESS MINING

intro

Контакты

Данил Сметанев

Data Science Executive Director

+7 915 470 42 42

smetanev.danil@gmail.com

Александр Кореков

Data Science Executive Director

https://www.linkedin.com/in/aleksandr-korekov-7225a49b/

+7 999 926 46 00

av.korekov@gmail.com

План

1. Зачем всё это нужно
2. Какие данные и алгоритмы используем
3. Как оценить качество реконструкции процесса?
4. О некоторых алгоритмах векторного представления процессов
5. Use-cases из проанализированных процессов
6. О библиотеке sberPM

Зачем?

Process Mining - это технология анализа и моделирования бизнес-процессов и клиентских путей на основе «цифровых следов» (логов)

Условия

- >80% процентов процесса **реализуется в информационных системах**
- Действия пользователей в ИС логируются на достаточном уровне
- В **логах** присутствуют id, событие, дата-время события

Цель

выявление инсайтов по процессу для повышения операционной эффективности и улучшения клиентского опыта

Суть анализа

Получаем данные из систем-источников

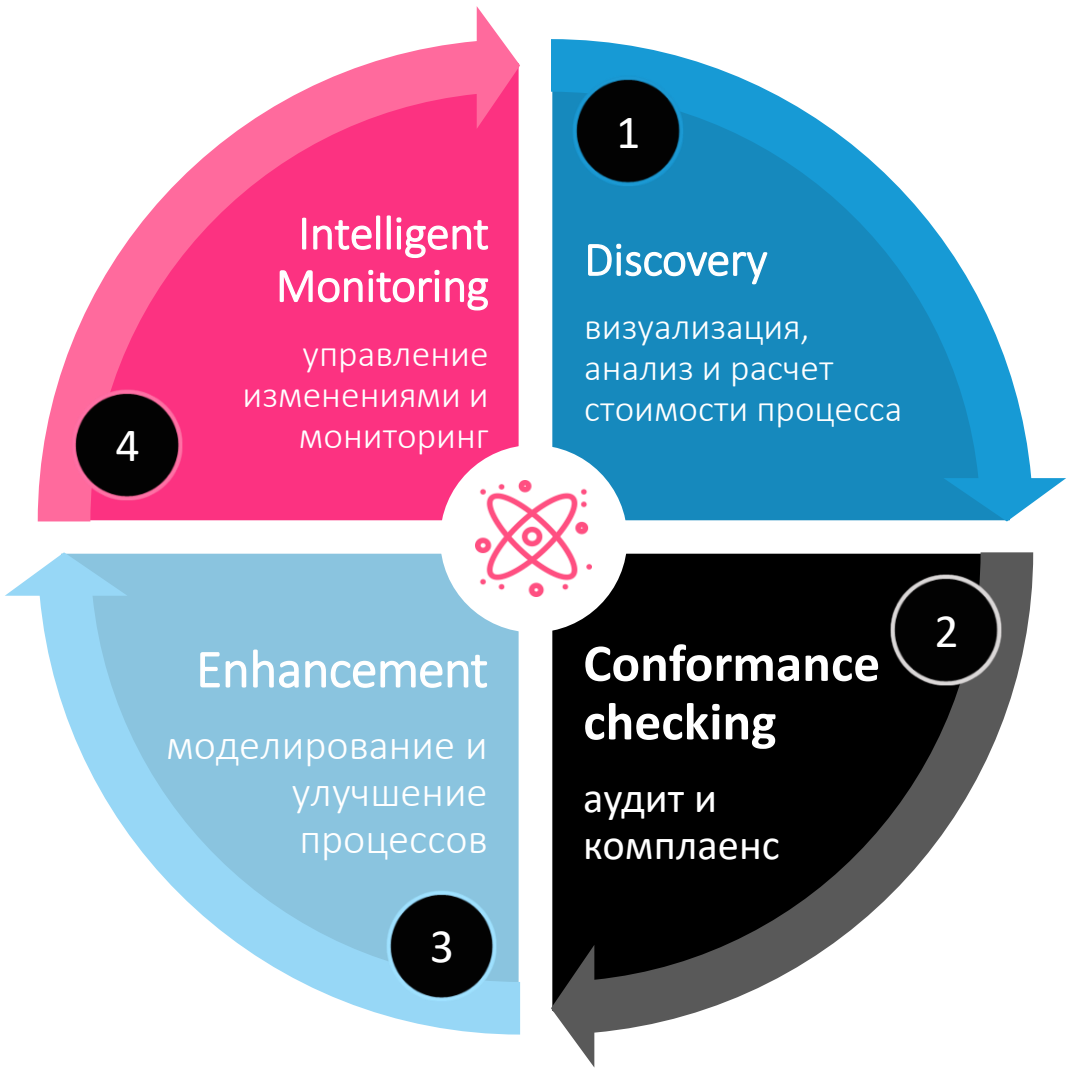
Восстанавливаем фактический процесс

Обрабатываем инструментами ML

Выявляем инсайты/ неэффективности

Применение Process Mining

- Оцифровка процесса «до» и «после» изменений
- Мониторинг изменений процесса/показателей эффективности

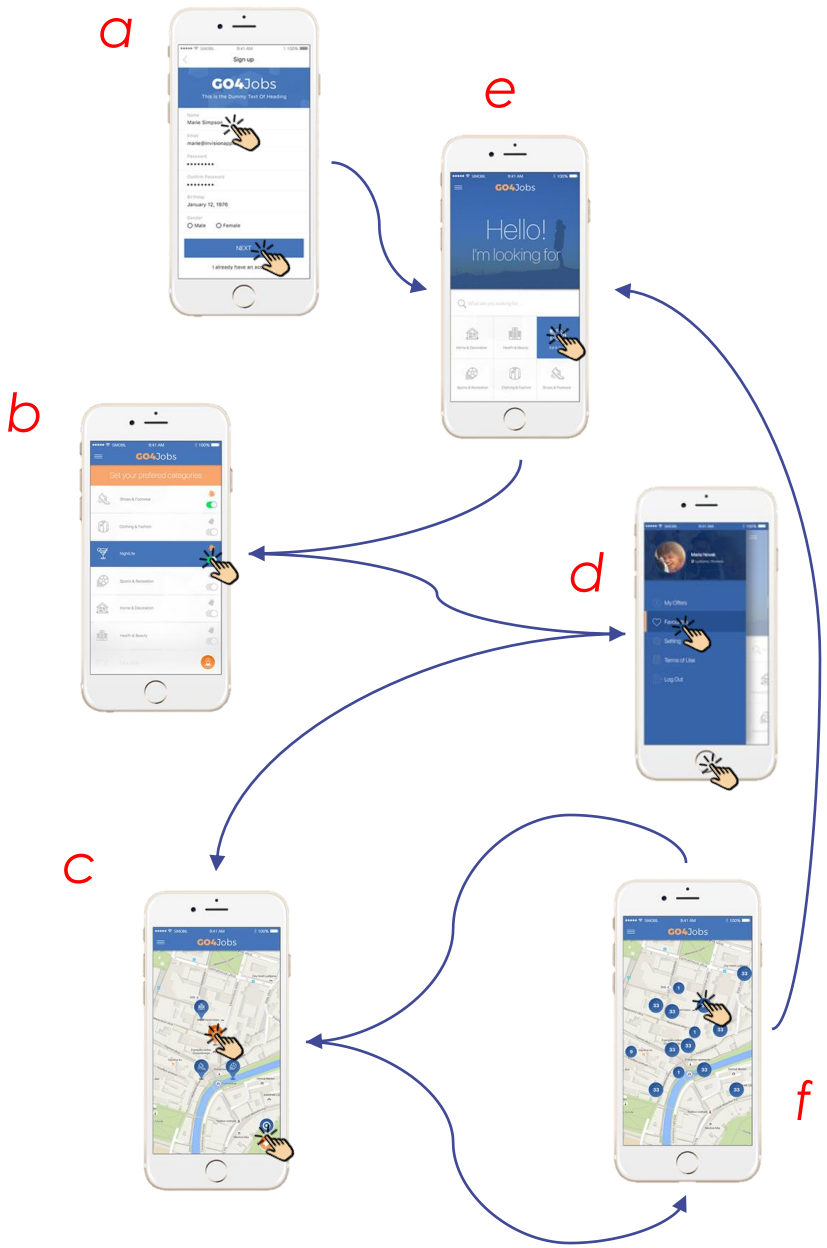


- Расчет производительности, затрат и стоимости процесса /этапа/конкретных исполнителей, ошибки, узкие места
- Анализ клиентских путей
- Автоматический хронометраж
- Анализ отклонений

- Авто-рекомендации по улучшению процесса
- Моделирование «Что если?»
- Прогноз отклонений и сбоев

- Проверка соответствия требованиям ВНД и регулятора
- Benchmark процесса (анализ лучших практик, GAP-анализ)
- Поиск мошеннических схем

Какая природа данных?



$$user\ session \qquad activities$$

$$\begin{bmatrix} id, \\ activity\_id, \\ attributes \end{bmatrix} \qquad \begin{bmatrix} id, \\ names, \\ attributes \end{bmatrix}$$

$$\text{event log } L = \begin{bmatrix} \langle a, e \rangle^5, \langle a, b, c, e \rangle^{10}, \\ \langle a, b, e, f \rangle, \langle a, c, e \rangle, \\ \langle a, d, d, d, e \rangle, \langle a, d, d, e \rangle^2, \\ \langle a, d, e \rangle^{10}, \langle a, c, b, e \rangle^{10} \end{bmatrix}$$

Данные

Event-log

Файл с записями о событиях в информационной системе в хронологическом порядке
На данный момент в большинстве АС ведется логирование действий сотрудников

Минимальный набор атрибутов лог-файла для проведения исследования

Идентификатор
ID экземпляра процесса, например, номер заявки

Название операции
Например: клик пользователя на кнопку

Временная метка
указывает дату и время выполнения операции

ID	Операция	Дата события
34781	Создание заявки	15.05.2020 11:1
34781	Отправка заявки	15.05.2020 11:2
34781	Поступление заявки н	15.05.2020 11:2
34781	Назначение КС и испо	15.05.2020 12:4
34781	Отправка внутренней	15.05.2020 13:1
34781	Формирование реестр	18.05.2020 23:1
34781	Прием реестра	19.05.2020 10:5
34781	Сканирование	19.05.2020 10:4
34781	Закрытие заявки	19.05.2020 15:19

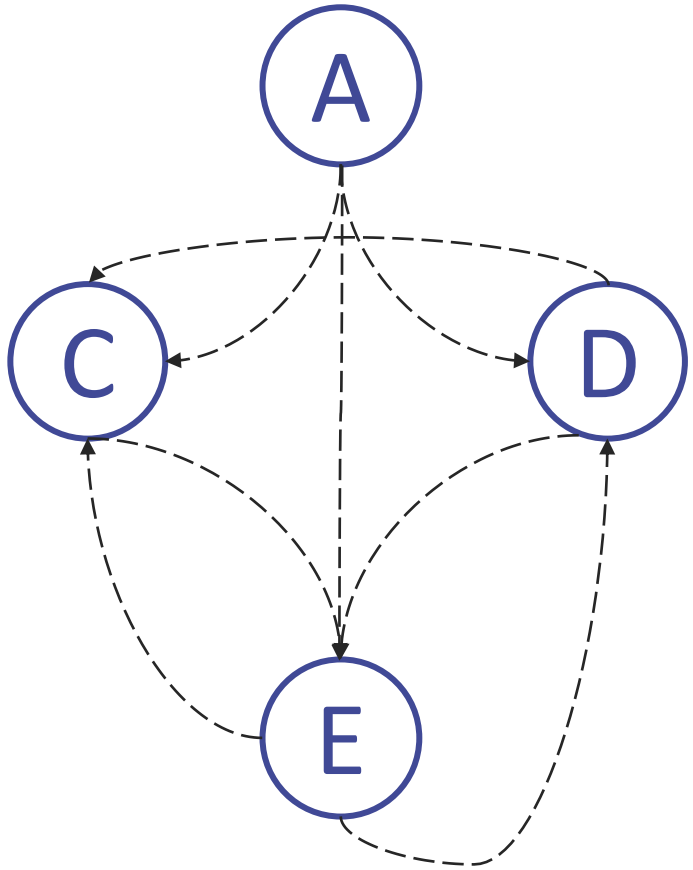
+ любые доп. атрибуты:

исполнитель, роль пользователя, продукт, локация, текстовый комментарий

Проблема: как определить истинную структуру процесса?

event log $L = \begin{bmatrix} \langle a, e \rangle^5, \langle a, b, c, e \rangle^{10}, \\ \langle a, b, e, f \rangle, \langle a, c, e \rangle, \\ \langle a, d, d, d, e \rangle, \langle a, d, d, e \rangle^2, \\ \langle a, d, e \rangle^{10}, \langle a, c, b, e \rangle^{10} \end{bmatrix}$

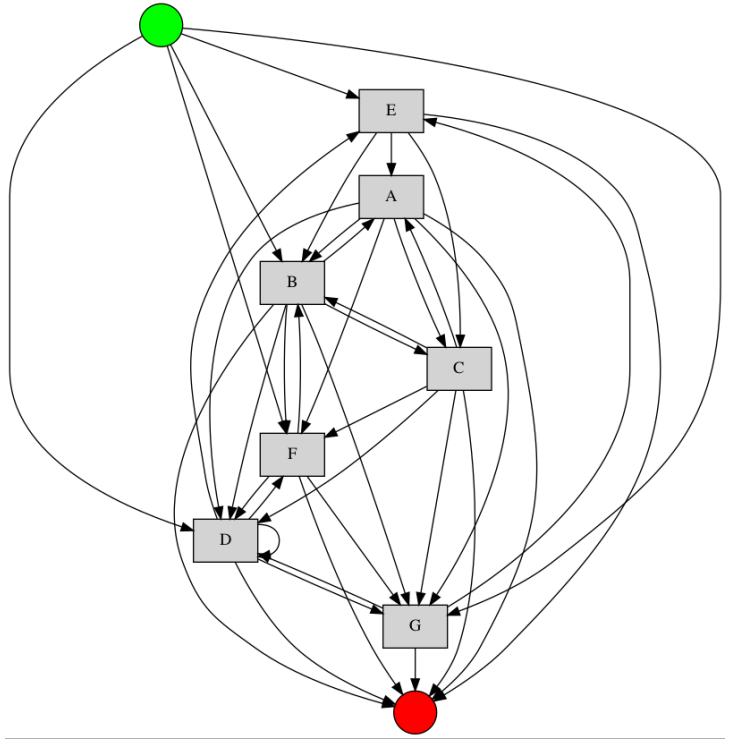
?



Как может помочь Process Mining?

1

Реконструкция графа процесса



2

Расчет метрик качества реконструкции графа

Quality Criteria:

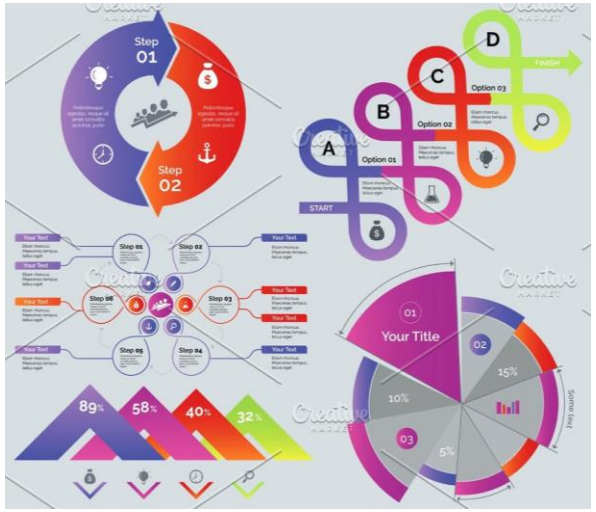
1. Fitness
2. Simplicity
3. Precision
4. Generalization

Metrics:

1. Parsing Measure
2. Partial Fitness Complete
3. Behavioral precision B_p and recall
4. Extended Cardoso Metric
5. Extended Cyclomatic Metric
6. Advanced behavioral appropriateness
7. Structural similarity
8. Behavioral recall r_B^p, specificity s_B^n and precision
9. И другое

3

Моделирование изменений в процессах



Абсолютно бессмысленная картинка про сравнение реализаций процесса

Basic algos: Alpha, Alpha+, Heuristics

Проблема Лог-файл представляет собой «плоскую» таблицу, истинная структура процесса неизвестна

Идея Представить процесс в виде символьной / числовой матрицы и эвристически описать процесс

Семейство альфа-алгоритмов

Direct succession: $X > Y$
- Если за какой-нибудь X непосредственно следует за Y

Causality: $X \rightarrow Y$
- Если X > Y и нет отношений Y > X

Parallel: $X \,||\, Y$
- Если X > Y и есть отношения Y > X

Unrelated: $X \# Y$
- Если нет отношений X > Y и нет отношений Y > X

Эвристический майнер

$$a \Rightarrow b = \frac{|a > b| - |b > a|}{|a > b| + |b > a| + 1}$$

$$d \Rightarrow d = \frac{|d > d|}{|d > d| + 1}$$

Альфа-алгоритм

$$\textit{event log}\ L = [\langle a, b, c, d\rangle^3, \langle a, c, b, d\rangle^2, \langle a, e, d\rangle^1]$$

⇒W	A	B	C	D	E
A	#	→	→	#	→
B	←	#	||	→	#
C	←	||	#	→	#
D	#	←	←	#	←
E	←	#	#	→	#

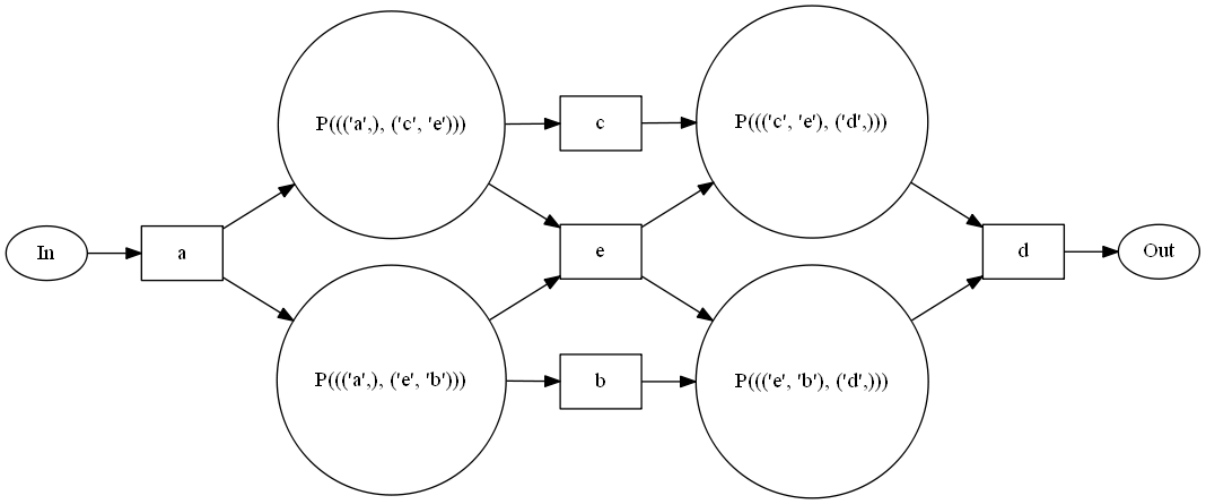
({a}, {b, e}), ({a}, {c, e}),

({b, e}, {d}), ({c, e}, {d})

Альфа-алгоритм

({a}, {b, e}), ({a}, {c, e}),

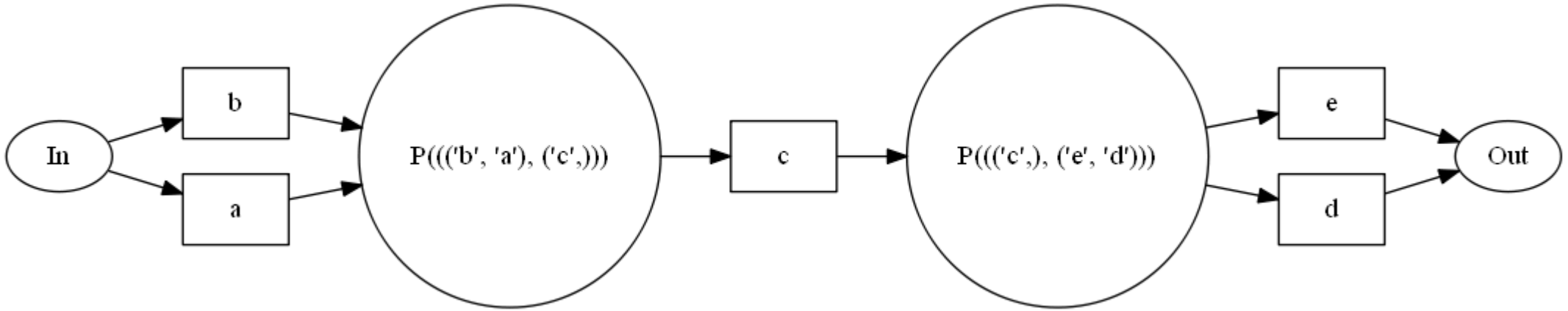
({b, e}, {d}), ({c, e}, {d})



Альфа-алгоритм: ограничения

Corner cases with non-existestent processes

$$\text{event log } L = [\langle a, c, d \rangle, \langle b, c, e \rangle, \langle a, c, e \rangle]$$

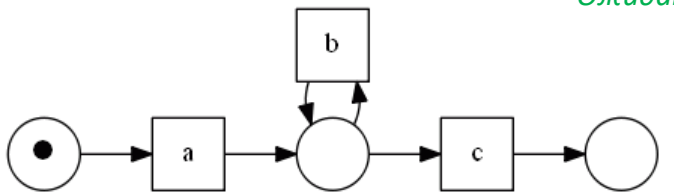


Альфа-алгоритм: ограничения

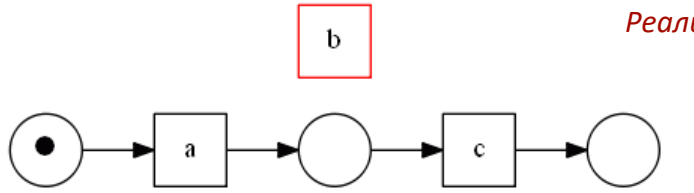
Corner cases with loops

1. Циклы длиной «1» event log $L = [\langle a, c\rangle, \langle a, b, c\rangle, \langle a, b, b, c\rangle, \langle a, b, b, b, c\rangle]$

Ожидание

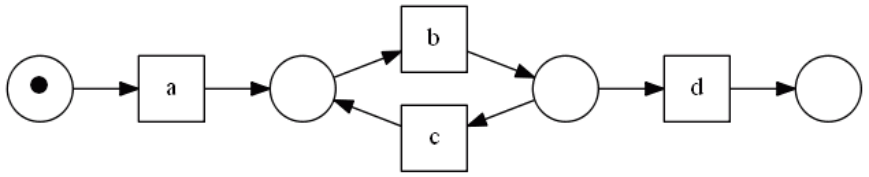


Реальность

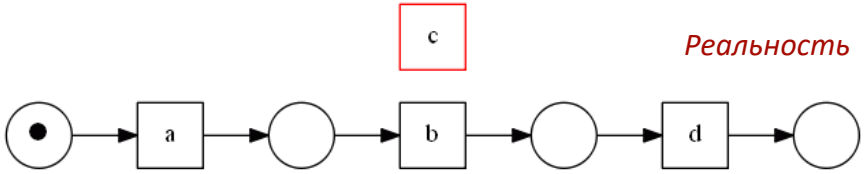


2. Циклы длиной «2» event log $L = [\langle a, b, d\rangle, \langle a, b, c, b, d\rangle, \langle a, b, c, b, c, b, d\rangle]$

Ожидание



Реальность



Эвристический майнер

$$\textit{event log}\ L = \left[\begin{matrix}\langle a,e\rangle^{5},\langle a,b,c,e\rangle^{10},\langle a,b,e\rangle,\langle a,c,e\rangle,\\ \langle a,d,d,d,e\rangle,\langle a,d,d,e\rangle^{2},\langle a,d,e\rangle^{10},\langle a,c,b,e\rangle^{10}\end{matrix}\right]$$

⇒W	A	B	C	D	E
A	0	0.92	0.92	0.93	0.83
B	-0.92	0	0	0	0.92
C	-0.92	0	0	0	0.92
D	-0.93	0	0	0.8	0.93
E	-0.83	-0.92	-0.92	-0.93	0

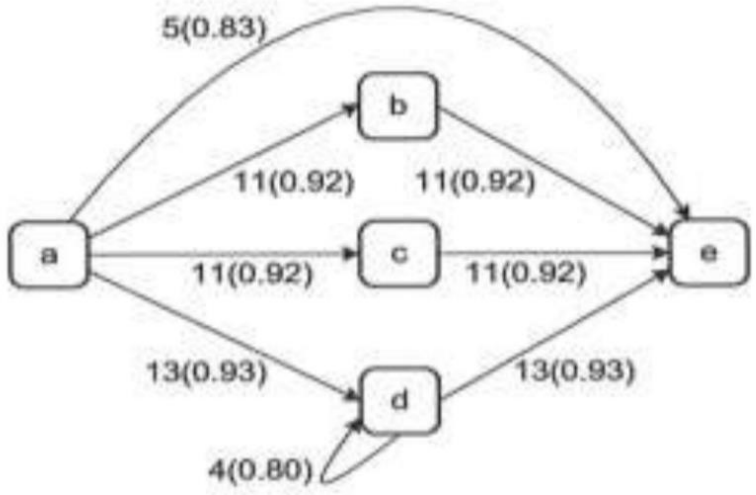
$$a \Rightarrow b = \frac{|a > b| - |b > a|}{|a > b| + |b > a| + 1}$$

$$d \Rightarrow d = \frac{|d > d|}{|d > d| + 1}$$

Эвристический майнер

$$\text{event log } L = \left[\begin{array}{c} \langle a,e\rangle^5, \langle a,b,c,e\rangle^{10}, \langle a,b,e\rangle, \langle a,c,e\rangle, \\ \langle a,d,d,d,e\rangle, \langle a,d,d,e\rangle^2, \langle a,d,e\rangle^{10}, \langle a,c,b,e\rangle^{10} \end{array}\right]$$

⇒W	A	B	C	D	E
A	0	0.92	0.92	0.93	0.83
B	-0.92	0	0	0	0.92
C	-0.92	0	0	0	0.92
D	-0.93	0	0	0.8	0.93
E	-0.83	-0.92	-0.92	-0.93	0



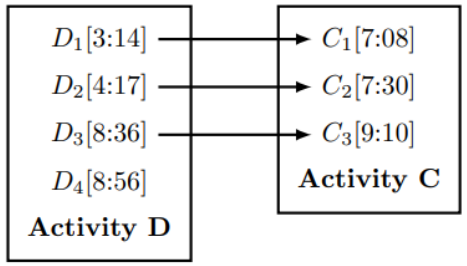
Алгоритмы «без разметки»: Correlation miner

Проблема Неизвестен истинный id процесса – это может быть, например, как id сессии, так и id юзера. Или процесс проходит сквозь несколько систем без единого id

Идея У событий, связанных прямым переходом будет высокая частота a->b и низкое время перехода

Концепция

$$P/S_{i,j} = \frac{\sum_{(e,f)\in\Omega_{i,j}} b(e,f)}{|\Omega_{i,j}|}$$



Результат

$$P/S = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0.00 & 0.47 & 0.97 & 0.73 & 0.88 \\ 0.53 & 0.00 & 1.00 & 0.67 & 0.90 \\ 0.03 & 0.00 & 0.00 & 0.33 & 0.57 \\ 0.26 & 0.33 & 0.67 & 0.00 & 0.70 \\ 0.12 & 0.10 & 0.43 & 0.30 & 0.00 \end{pmatrix} \end{array}$$

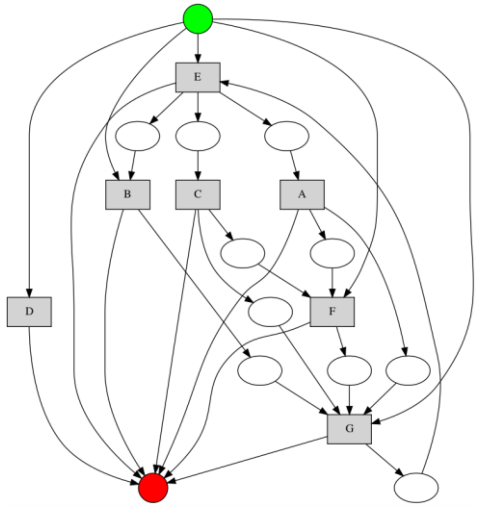
$$D = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 74 & 109 & 108 & 220 \\ 54 & 0 & 214 & 164 & 69 \\ 6 & 0 & 0 & 87 & 42 \\ 0 & 106 & 150 & 0 & 124 \\ 0 & 102 & 62 & 96 & 0 \end{pmatrix} \end{array}$$

Heurstics

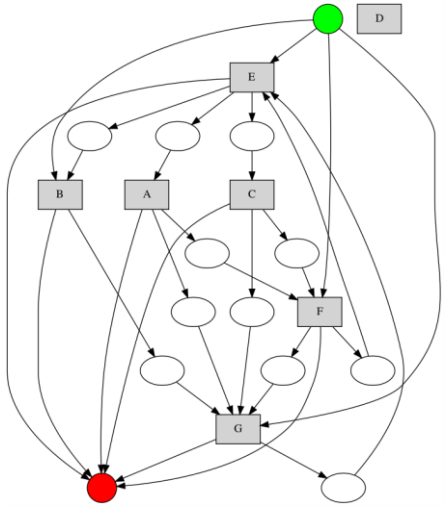
$$\mathcal{C}_{ij} = \frac{D_{i,j}}{P/S_{i,j}} \cdot \frac{1}{min(|E_i|, |E_j|)}$$

$$minimize \sum_{x_{ij}\in\mathcal{X}} \mathcal{C}_{ij} \cdot x_{ij}$$

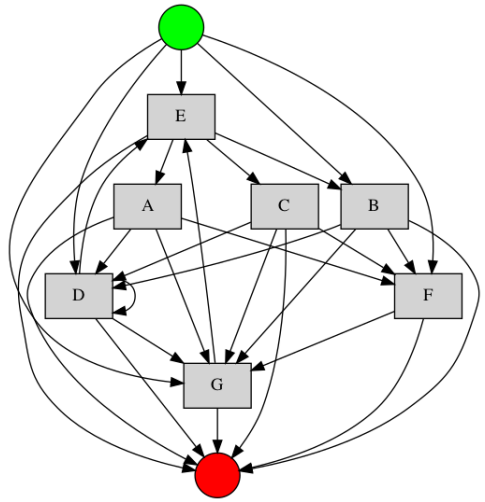
Сравнение майнеров



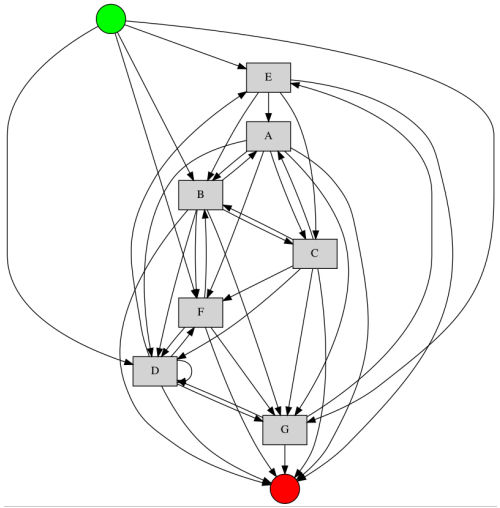
Alpha miner



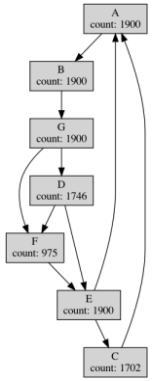
Alpha+



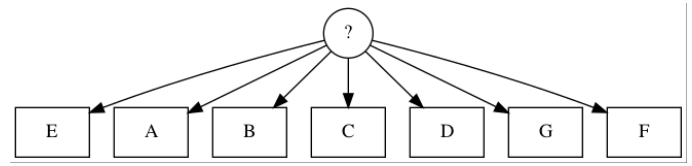
Heurstics



DFG



Correlation



Inductive

Критерии качества

Fitness

Насколько точно модель может воспроизвести все пути из лога

$$fitness = \frac{1}{2}\left(1 - \frac{missed}{consumed}\right) + \frac{1}{2}\left(1 - \frac{remaining}{produced}\right)$$

Simplicity

Оценивает «простоту» модели – количество «дублирующихся» и «лишних» задач

$$simplicity = \frac{N\ transitions\ - Duplicates\ - Redundant}{N\ transitions}$$

Generalization

Оценивает количество «возможных» переходов, которых в явном виде нет в исходных данных

$$generalization = 1 - \frac{1}{\sum_i \sqrt{node_i\ nb\ of\ occurences}}$$

Precision

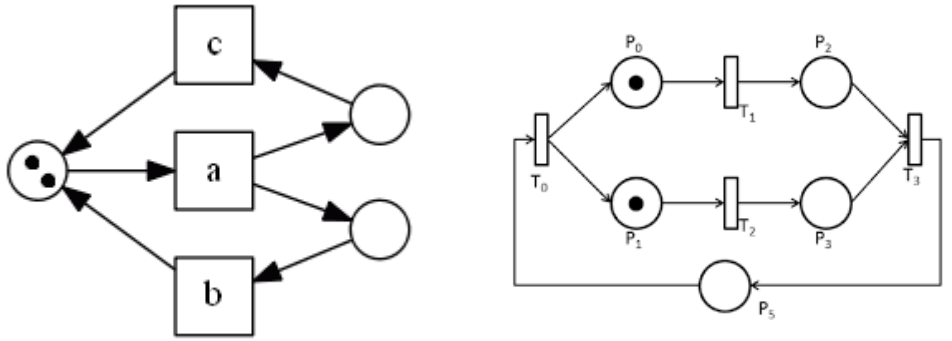
Оценивает все возможные переходы и встречались ли они в исходных данных

$$precision = 1 - \frac{total\ edges\ - used\ edged}{total\ edges}$$

See full reference: http://www.temjournal.com/content/84/TEMJournalNovember2019_1232_1241.pdf

Сети Петри

Сеть Петри – двудольный ориентированный граф, содержащий вершины двух типов:
— места/узлы/вершины (обозначаются кружками)
— переходы (обозначаются прямоугольниками)



Сеть Петри – это тройка $N = (P, T, F)$

P — узлы, T — переходы, F — дуги; $P \cap T = \emptyset$

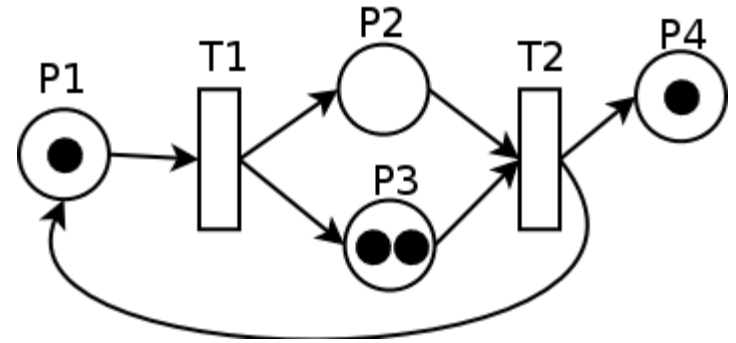


Рис.1 Рандомные картинки из Интернета по запросу «Petri Net»

Сеть Петри используется для двух вещей:

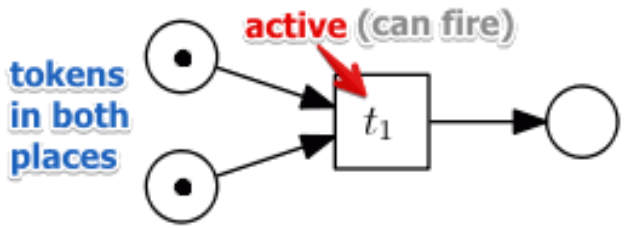
1) Описание процесса: допустимые переходы и условия ветвления
2) Реконструкция графа: убираем редкие и неважные переходы и оставляем суть

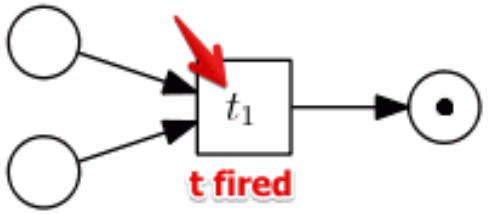
*Формально, в Process Mining используется не Petri Net, а Workflow Net**

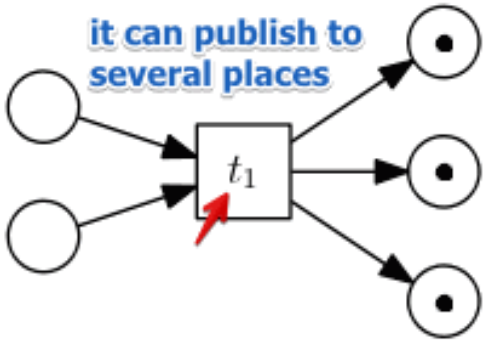
* http://mlwiki.org/index.php/Workflow_Nets

Примеры

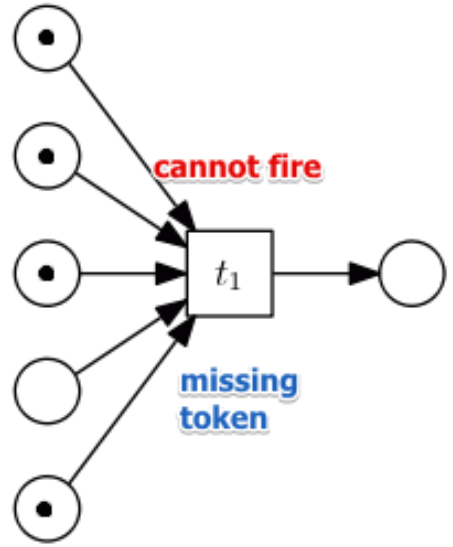
Marking

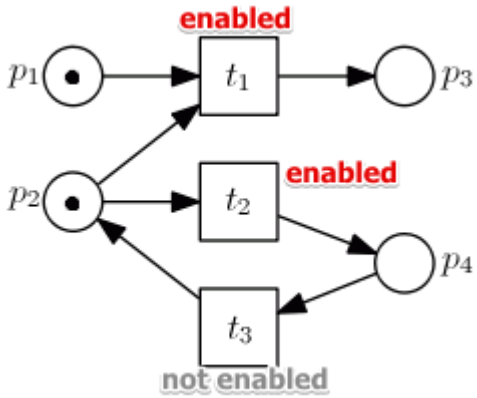






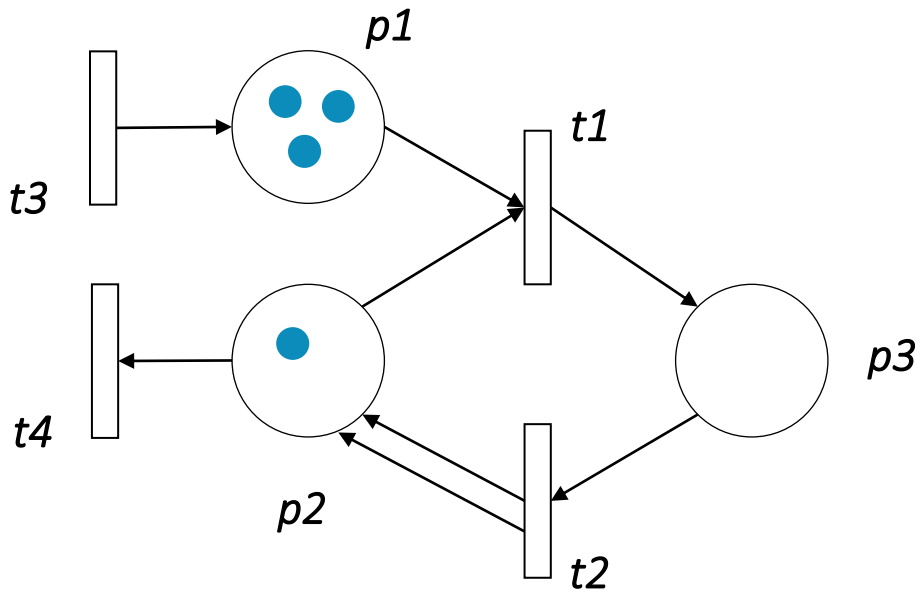
Enabled Transitions





Token firing algo

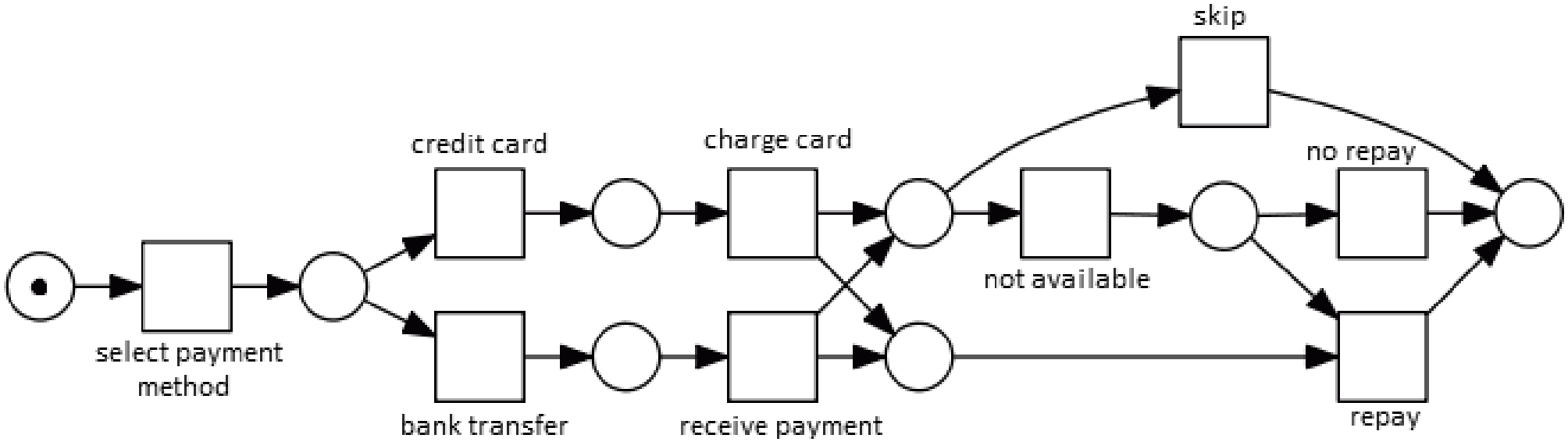
Token firing – динамика в сети Петри – это последовательность исполнения *(firing)* переходов *(transitions)*



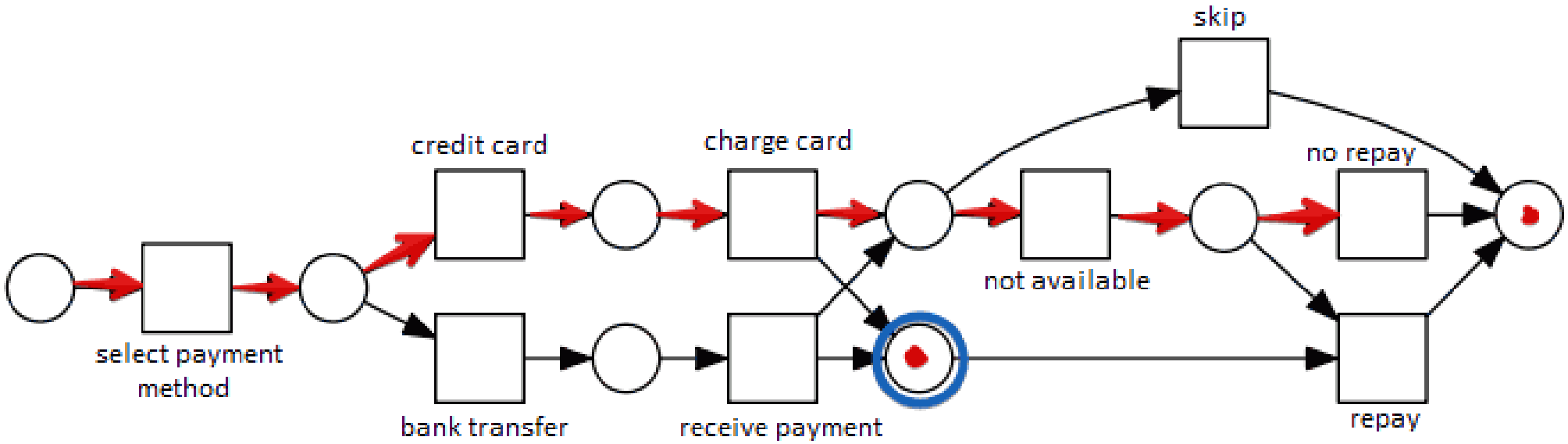
Пример:

1) Transition t1 firing: 1 токен убираем из places p1 и p2 и добавляем 1 токен в p3
2) Transition t2 firing: убираем 1 токен из p3 и добавляем 1 в p2

Fitness example



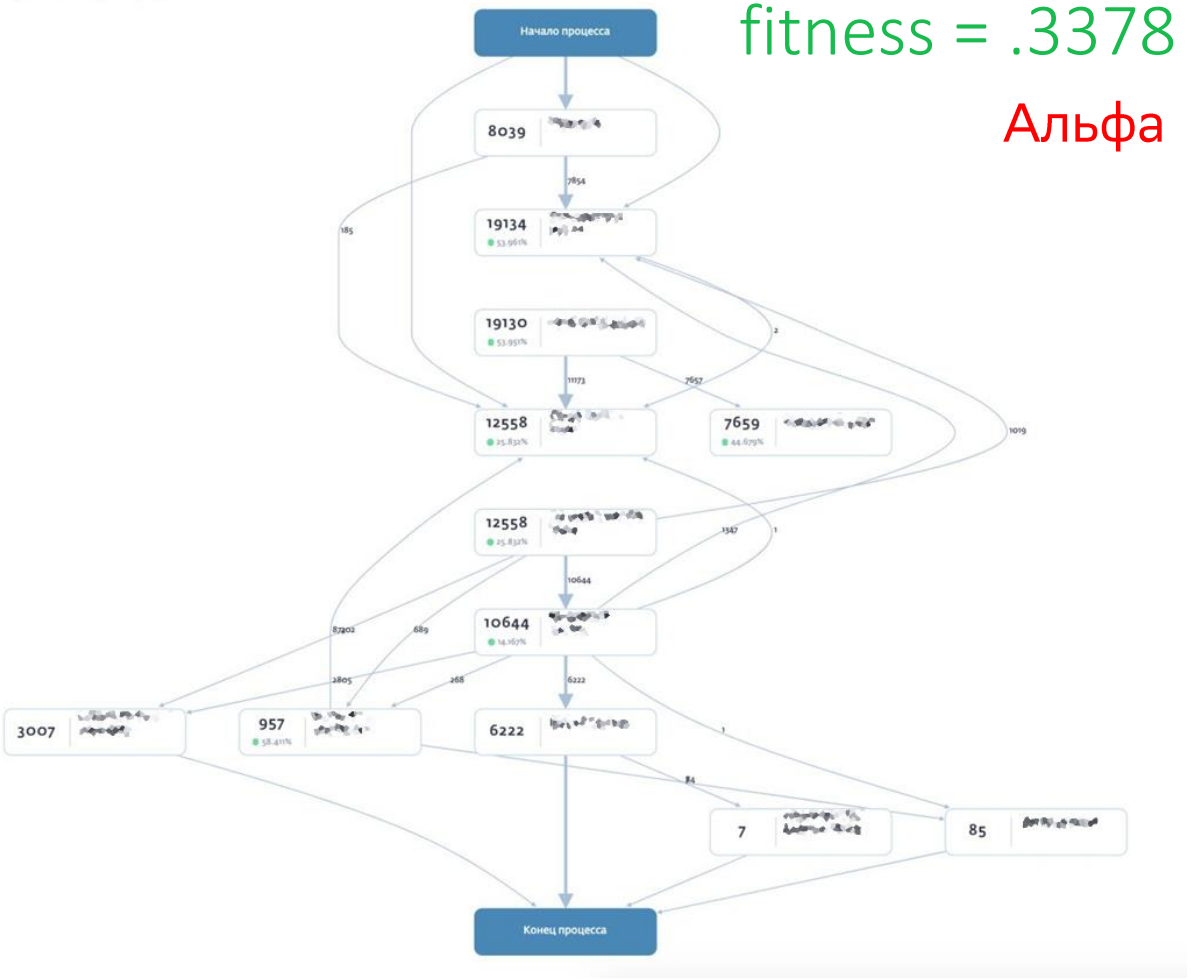
Fitness example



Сравнение алгоритмов

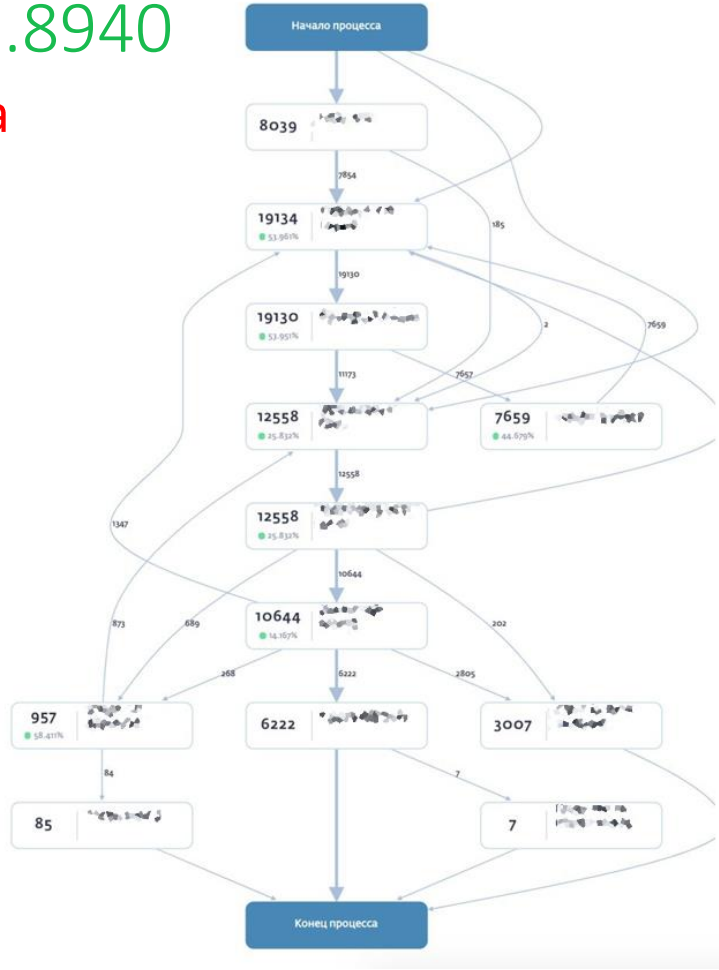
fitness = .3378

Альфа



fitness = .8940

Эвристика

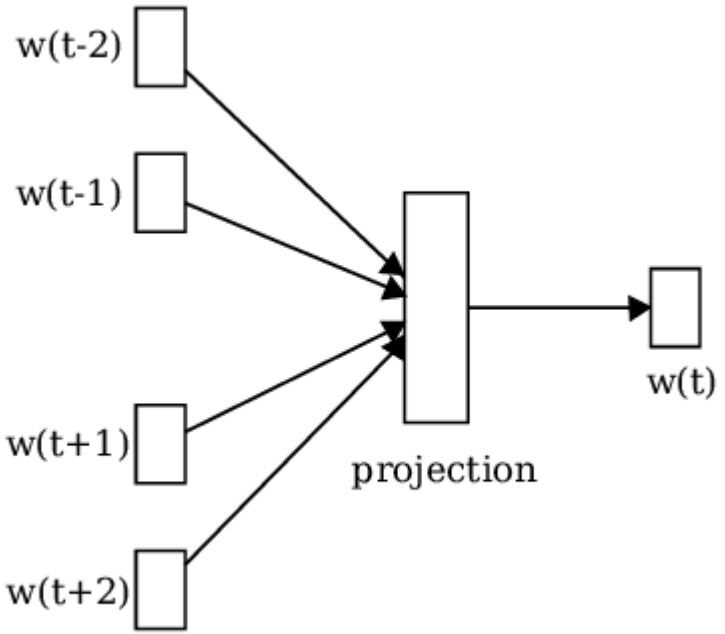


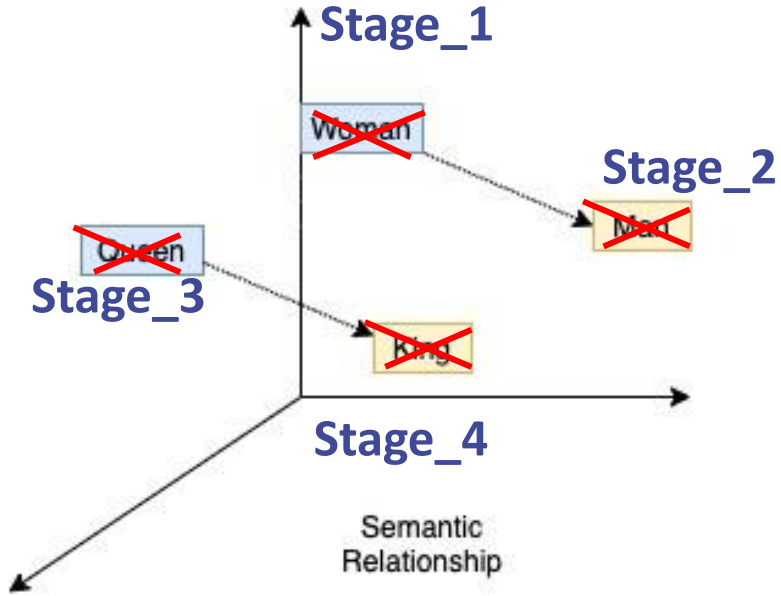
ML-задачи в Process Mining

1. «Сжатие» большого процесса
2. Идентификация Bottlenecks в процессах
3. Оценка влияния Bottlenecks на метрики процесса
4. Предсказание переходов по клиентскому пути
5. Поиск «оптимального пути» процесса

A very simple approach: processWord2Vec

$$\log L = \begin{bmatrix} \langle a, e \rangle^5, \\ \langle a, b, c, e \rangle^{10}, \\ \langle a, b, e, f \rangle, \\ \langle a, c, e \rangle, \\ \langle a, d, d, d, e \rangle, \\ \langle a, d, d, e \rangle^2, \\ \langle a, d, e \rangle^{10}, \\ \langle a, c, b, e \rangle^{10} \end{bmatrix}$$

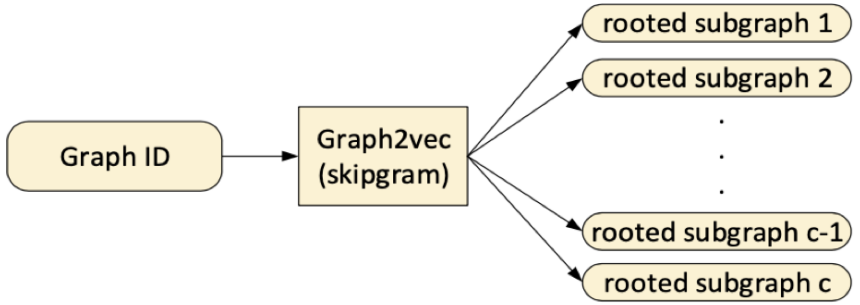




Graph2Vec (1/2)

Алгоритм работы:

1. Составляем пары – [(a,b), (b,c), (a,d) ….]
2. Представляем в виде графа с помощью – все возможные комбинации ребер и нод
3. Идем циклом по каждой ноде:
   3.1. Находим у ноды n всех соседей.
   3.2. Находим степень каждого соседа
   3.3. Объединяем множество всех степеней соседей и нашей ноды n
   3.4. Множество переводим в строку
   3.5. В виде словаря представляем ноде строку
4. Полученный строки мы объединяем в «предложение», которые будем подавать в Doc2Vec.
5. Получаем вектор для реализации процесса



http://jmlr.org/papers/volume12/shervashidze11a/shervashidze11a

Graph2Vec (2/2)

$$L = [\langle a, b, c \rangle^1, \langle a, d, c, b \rangle^1, \langle a, b, b, c, b \rangle^1]$$

Алгоритм работы:

1. Составляем пары:

<a,b,c> - (a,b) , (b,c)
<a,d,c,b> - (a,d) , (d,c), (c,b)
<a,b,b,c,b> - (a,b), (b,b), (b,c), (c,b)

2. Для каждой реализации процесса находим у соседей каждой ноды (1: A – 1, B-2, C-1, 2: A – 1, D-2, C – 2, B-1, 3: A-1, B-4, C-2)

3. Объединим множество всех степеней свободы и каждой ноды, в каждой реализации процесса:

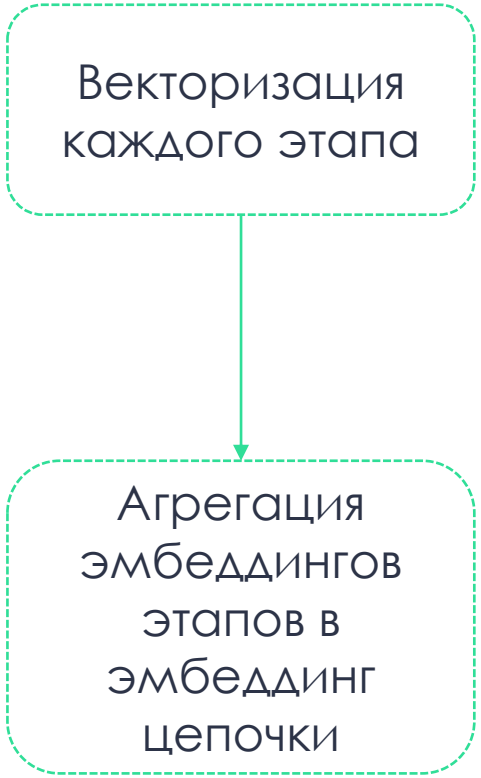
1, 2, 1 - > [1, 1_2, 2, 2_2_1, 1, 1_2]
1, 2, 2, 1 - > [1, 1_2, 2, 2_2_1, 2, 2_2_1, 1, 1_2]
1, 4, 4, 1, 4 - > [1, 1_4, 4, 1_4_4, 4, 4_4_1, 1, 4_1_4, 4, 1_4]

4. Используем лист степеней свободы в doc2vec и предсказываем последовательность этапов

[1, 1_2, 2, 2_2_1, 1, 1_2] **predict** <a,b,c>
[1, 1_2, 2, 2_2_1, 2, 2_2_1, 1, 1_2] **predict** <a,d,c,b>
[1, 1_4, 3, 1_4_4, 4, 4_4_1, 1, 4_1_4, 4, 1_4] **predict** <a,b,b,c,b>

5. Получаем вектора для каждой реализации процесса

Word2Mat(1/2)



Получение эмбеддинга каждого этапа процесса

Использована идея агрегации из Word2Mat[1]: эмбеддинг каждого этапа процесса преобразуется к квадратной матрице, далее они перемножаются, и полученная матрица «растягивается» к исходному размеру

[1]https://openreview.net/forum?id=H1MgjoR9tQ

Word2Mat (2/2)

High-Order Proximity Embedding (HOPE)[2]

$$\mathbf{S} = \mathbf{M}_g^{-1} \cdot \mathbf{M}_l$$

Table 1: General Formulation for High-order Proximity Measurements

Proximity Measurement	$\mathbf{M}_g$	$\mathbf{M}_l$
Katz	$\mathbf{I} - \beta \cdot \mathbf{A}$	$\beta \cdot \mathbf{A}$
Personalized Pagerank	$\mathbf{I} - \alpha\mathbf{P}$	$(1-\alpha) \cdot \mathbf{I}$
Common neighbors	$\mathbf{I}$	$\mathbf{A}^2$
Adamic-Adar	$\mathbf{I}$	$\mathbf{A} \cdot \mathbf{D} \cdot \mathbf{A}$

`source` эмбеддинги $U^s = \left[\sqrt{\sigma_1}\mathbf{v}_1^{\mathbf{s}}, \ldots, \sqrt{\sigma_k}\mathbf{v}_k^{\mathbf{s}}\right]$

`target` эмбеддинги $U^t = \left[\sqrt{\sigma_1}\mathbf{v}_1^{\mathbf{t}}, \ldots, \sqrt{\sigma_k}\mathbf{v}_k^{\mathbf{t}}\right]$

используя SVD разложение $S = V^s \Sigma V^{tT}$

Weighted Adjacency Matrix

$$S = A \odot QQ^T$$

$A_{pxp} = (a_{ij})$ - матрица смежности для графа процесса.

Q_{pxq} - взвешенная матрица метрик.

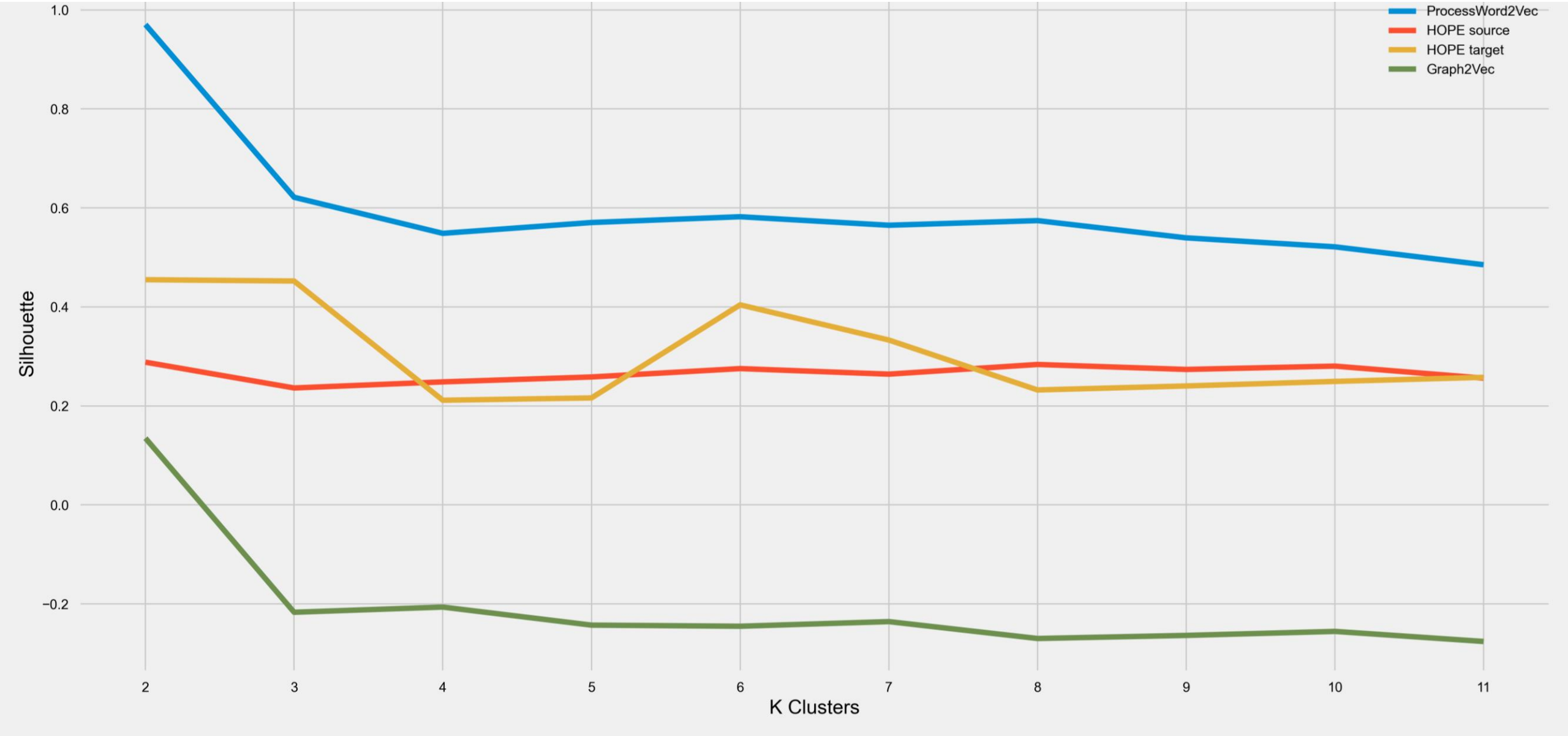
`source` эмбеддинги $U^s = \left[\sqrt{\sigma_1}\mathbf{v}_1^{\mathbf{s}}, \ldots, \sqrt{\sigma_k}\mathbf{v}_k^{\mathbf{s}}\right]$

`target` эмбеддинги $U^t = \left[\sqrt{\sigma_1}\mathbf{v}_1^{\mathbf{t}}, \ldots, \sqrt{\sigma_k}\mathbf{v}_k^{\mathbf{t}}\right]$

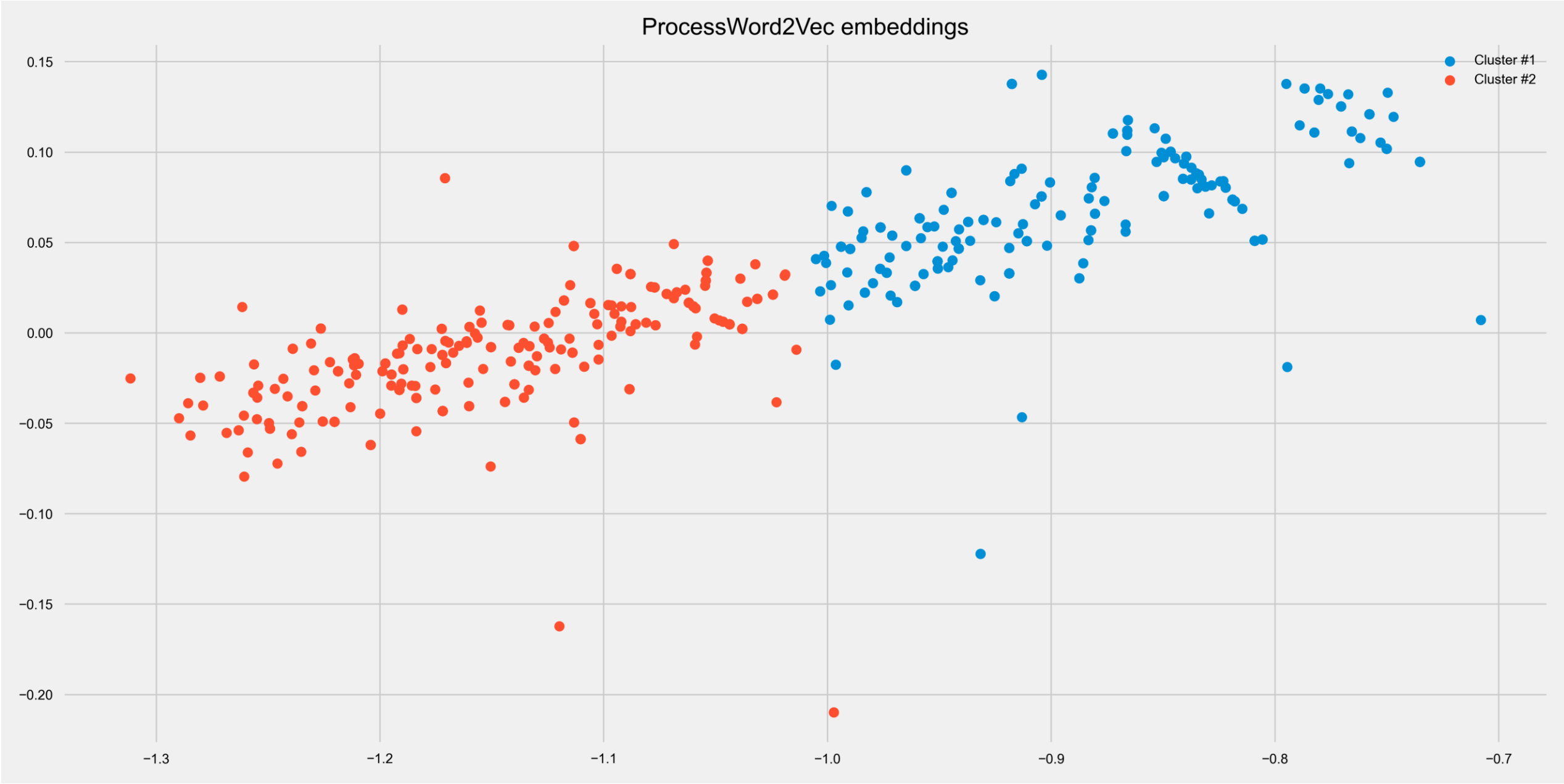
используя SVD разложение $S = V^s \Sigma V^{tT}$

[2]https://dl.acm.org/doi/pdf/10.1145/2939672.2939751

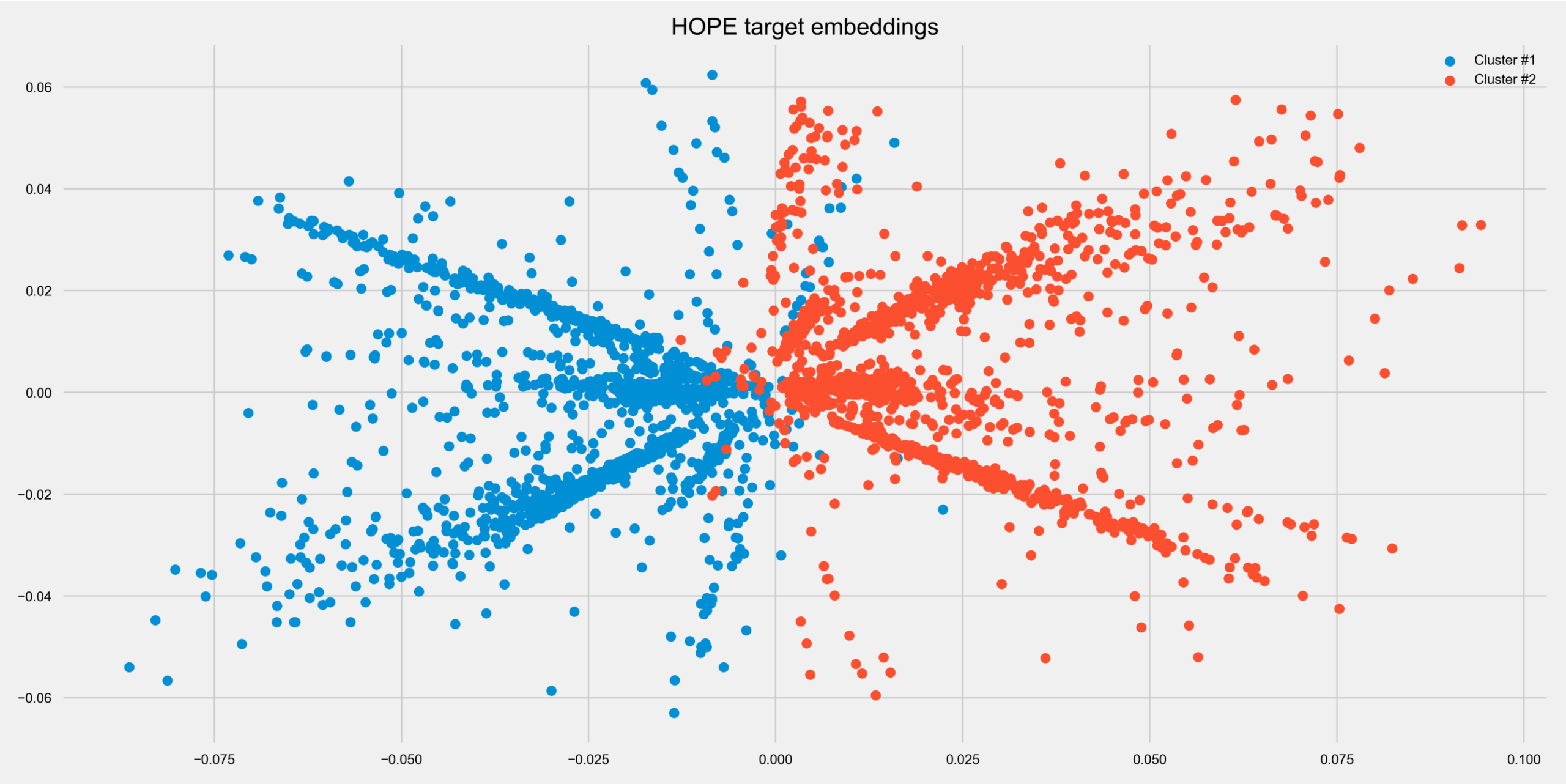
Сравнение алгоритмов



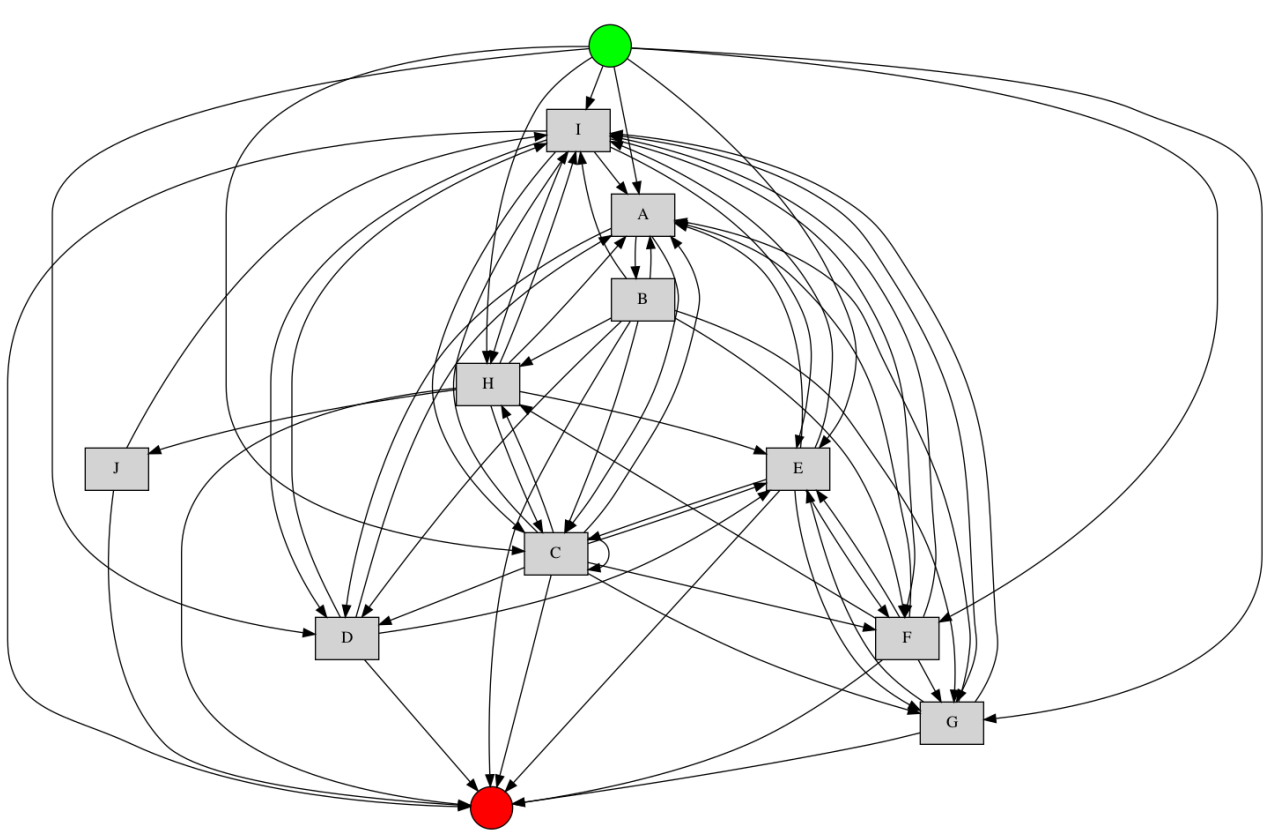
proc2vec

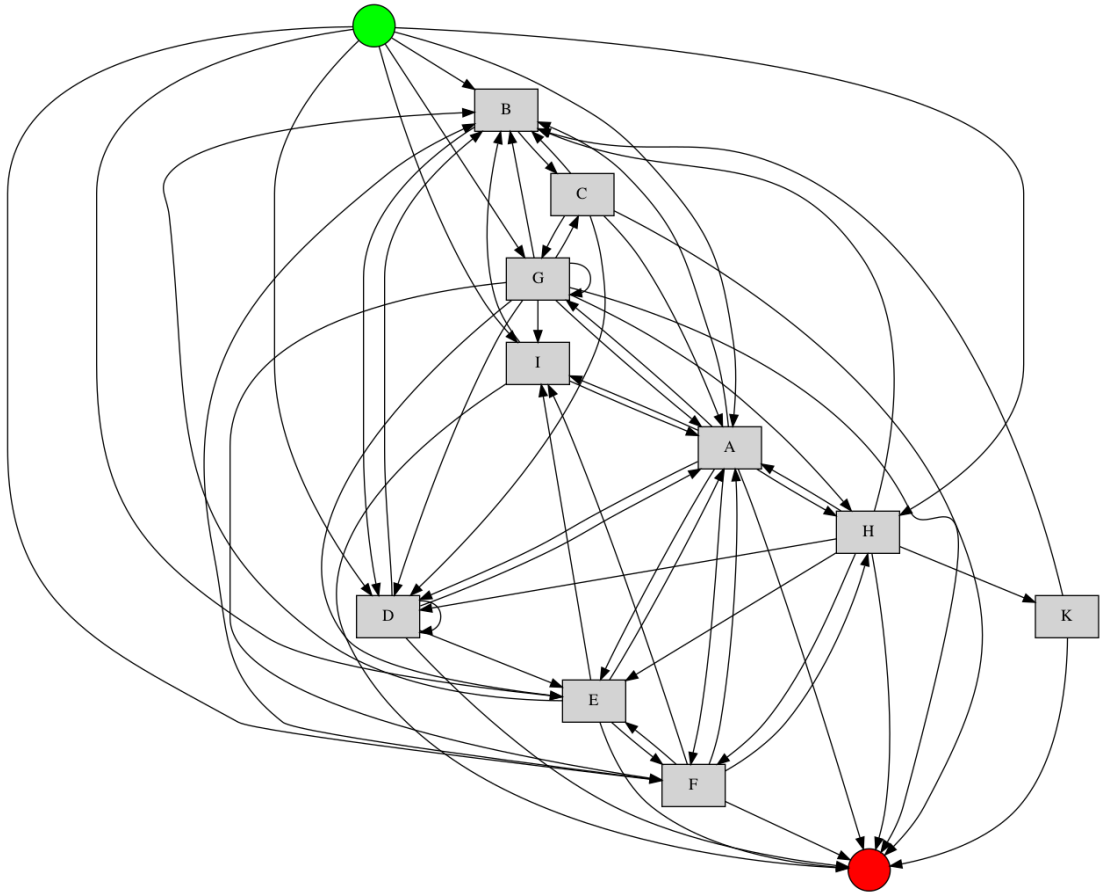

ProcessWord2Vec embeddings
Cluster #1
Cluster #2
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−1.3
−1.2
−1.1
−1.0
−0.9
−0.8
−0.7

HOPE

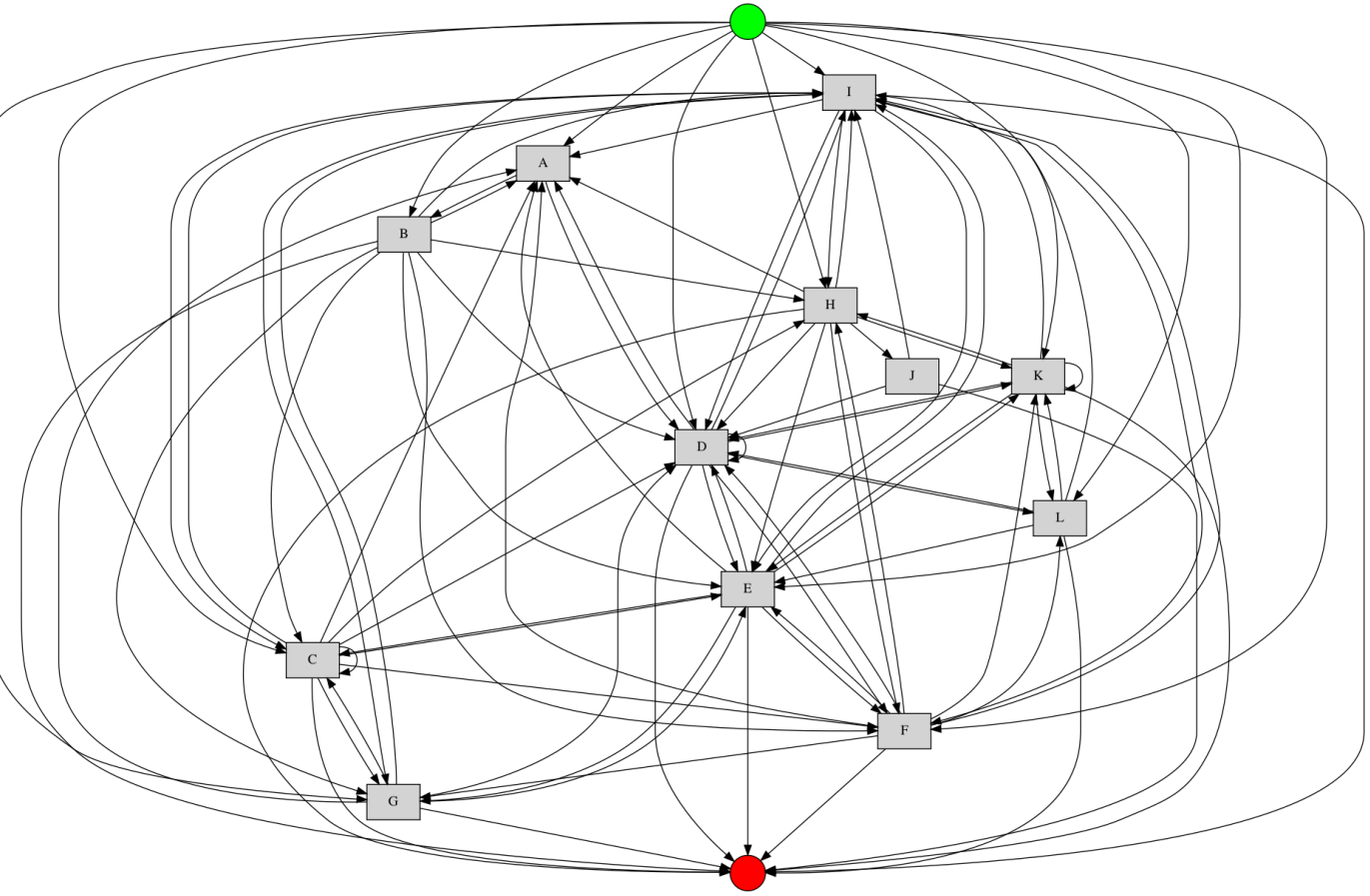

HOPE target embeddings
Cluster #1
Cluster #2
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100

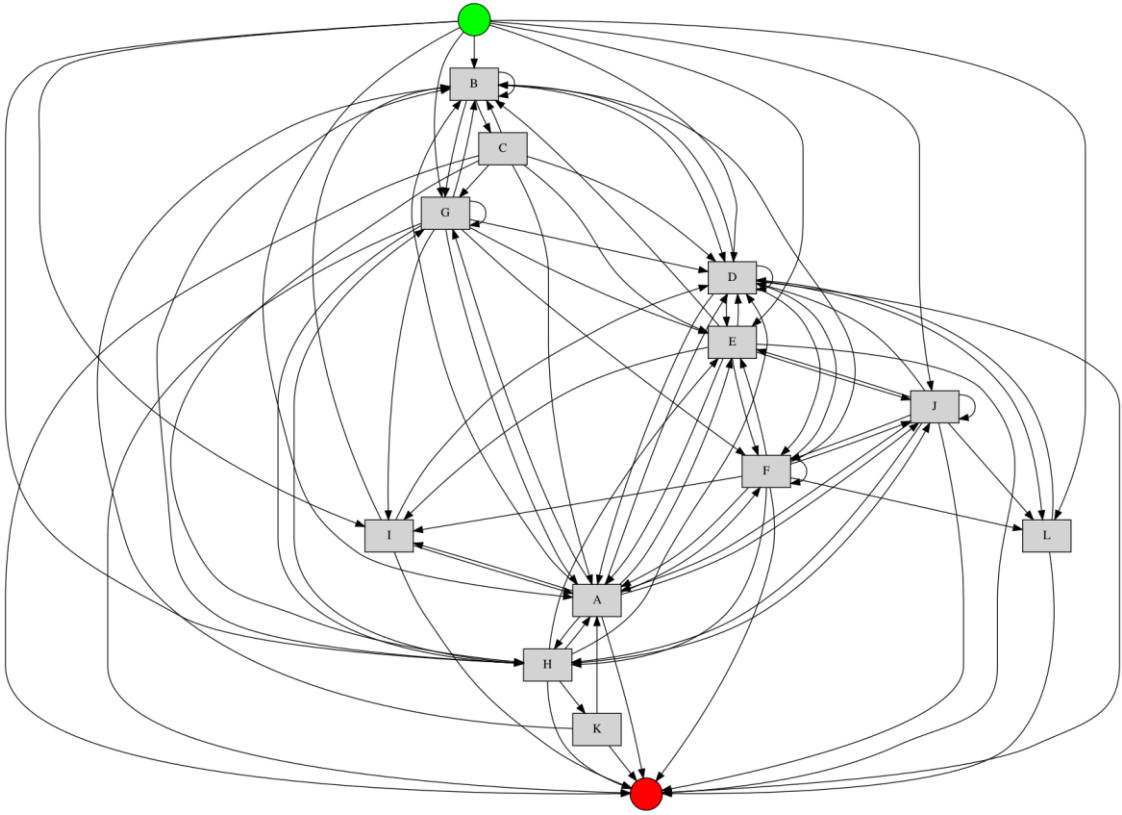
proc2vec



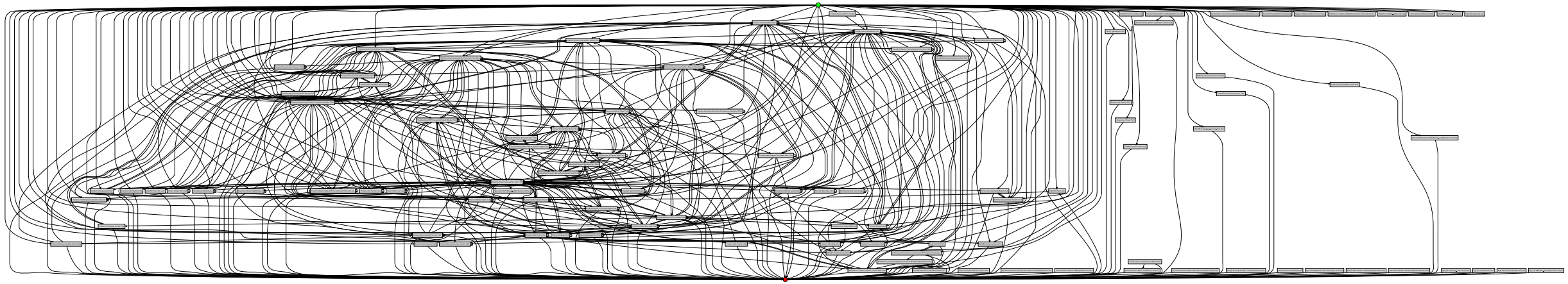


HOPE





Process Mining for Customer Journey (1/2)



- Большое количество переходов
- Нет основного пути, каждая цепочка уникальна
- Мало малоиспользуемых активностей

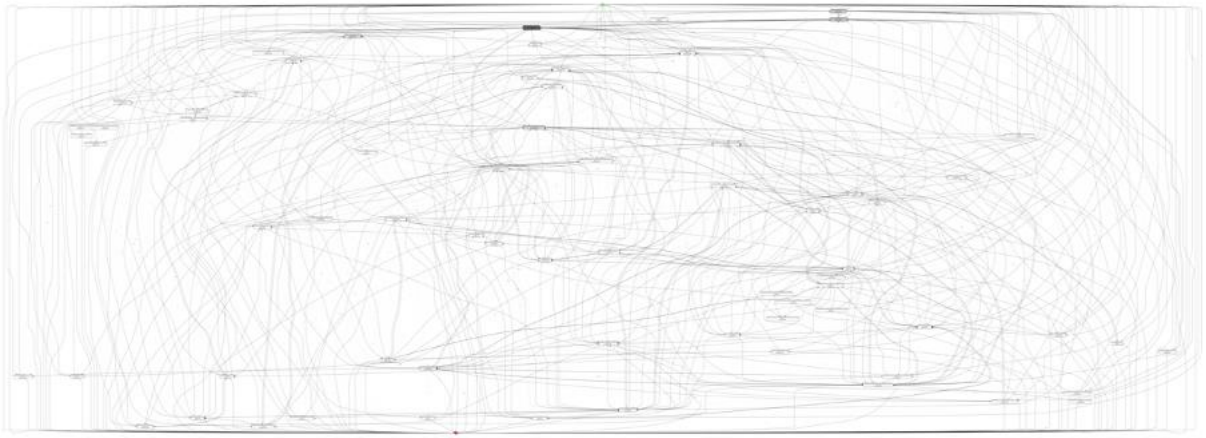
Process Mining for Customer Journey (2/2)

Процесс

Клиентский путь в мобильном приложении Сбера

AS IS

- 2351 активность
- невозможно анализировать
- не виден основной путь



Задача алгоритма

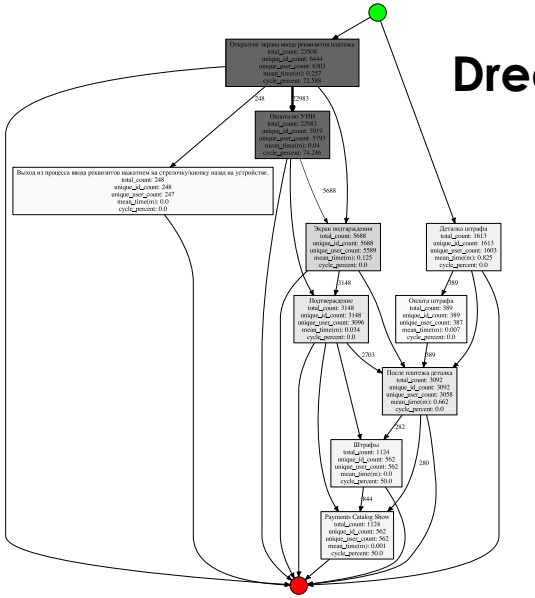
Реконструировать процесс

Минимизировать число шагов до успеха

Провести клиента до оплаты за минимальное время

Исключить циклы

Dream process



Доступные Open source решения

Python. Библиотека от Сбера с фокусом на использование ML для аналитики клиентских путей, в ней чуть меньше фичей по классическому process mining, больше аналитических тулзов (метрики, графики, ML)

Python. Библиотека on Fraunhofer Institute for Applied Information Technology (FIT) с фокусом на классический process mining – поддерживает больше методов PM и анализа качества реконструкции графа

pm4py для тех, кто ценит R

Java. Продукт от Eindhoven University of Technology – с интерфейсом и свистелками

(картинки кликабельные)

Python-решения: SberPM vs PM4PY

Объем лог-файла (ids / rows)	Library	Считывание лог-файла	Alpha	Alpha+	HeuMiner	HeuMiner + metrics	Token Replay
10к / 77к	SBER PROCESS MINING	0,8 sec	0,03 sec	0,06 sec	0,06 sec	0,1 sec	1,26 sec
	PM4PY	0,66 sec	0,17 sec	1,37 sec	1,15 sec	-	1,30 sec
250к / 1,6m	SBER PROCESS MINING	23,4 sec	0,84 sec	1,61 sec	1,68 sec	2,8 sec	34 sec
	PM4PY	15,1 sec	2,67 sec	10,5 sec	13,4 sec	-	6 sec
1m / 6,7m	SBER PROCESS MINING	95 sec	3,2 sec	6,2 sec	6,0 sec	10 sec	118 sec
	PM4PY	80 sec	13 sec	34 sec	32 sec	-	13 sec

Что ещё?

https://github.com/SberProcessMining/Sber_Process_Mining

SberPM is an open-source Python library for conducting a comprehensive analysis of business processes with the use of process mining and machine learning techniques.

https://habr.com/ru/company/sberbank/blog/554378/

https://sashakorekov.medium.com/

Обзорные статьи по использованию sberPM

https://icpmconference.org/2020/wp-content/uploads/sites/4/2020/10/ICPM_2020_paper_141.pdf

An In-depth Analysis of Reimbursement Processes Using Process Mining Techniques

http://www.processmining.org/

Process Mining Group, Math&CS department, Eindhoven University of Technology