Использование сочетания различных типов данных для рекомендации каналов

Кощенко Екатерина Васильевна

Научный руководитель: Коваленко Владимир Владимирович

НИУ ВШЭ СПбШФМКН, 2021

Обзор области

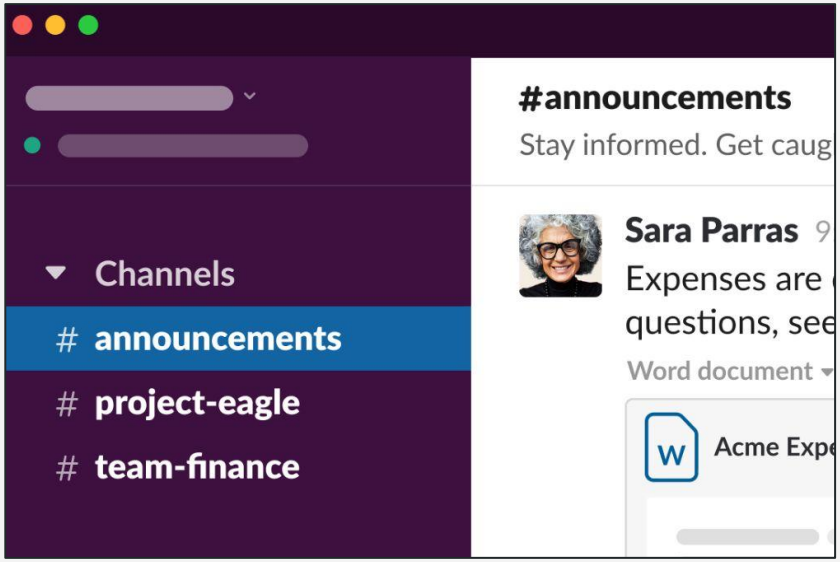
Введение: Коллаборация

- Работники IT компаний используют различные коллаборативные платформы для ежедневных задач
- **Алгоритмы социо-технической поддержки** помогают коллаборативной работе, используя данные о социальных взаимодействиях и технических репозиториях. Частный случай: рекомендательные системы [1]

1. Robillard M.P et.al., Recommendation Systems in Software Engineering, 2014

Рекомендации каналов

- Система каналов фокусирует общение пользователей на конкретных проектах, темах или командах
- Релевантные и разнообразные рекомендации каналов могут улучшить взаимодействие сотрудников организации [2]



2. M´ıvian Ferreira et.al., Algorithms for estimating truck factors: a comparative study, 2019

Существующие проблемы

Социальные и технические данные распределены по нескольким платформам (Slack, Telegram, GitHub, сайты компаний).

Поэтому алгоритмы социо-технической поддержки обычно сконцентрированы на одном типе данных [3, 4, 5].

3. H. Alperen Çetin et.al., A review of code reviewer recommendation studies: Challenges and future directions, 2021
4. Renaud Bourassa, Personalized channel recommendations in Slack, 2017
5. Patanamon Thongtanunam et.al., Who should review my code? A file location-based code-reviewer recommendation approach for modern code review, 2015

Space

Space — интегрированная среда для командной работы от компании JetBrains.

- Git-hosting
- Code review
- Tasks tracker
- Channels
- Calendars
- Blogs
- Team profiles

Цель:

Разработать систему рекомендации каналов, работающую на данных разных типов:

- **Каналы**: основная информация
- **Структура организации**: позиция пользователя
- **Технические репозитории**: профессиональные интересы и экспертиза

Задачи:

- Собрать данные: три модальности + пользователи
- Построить общеиспользуемые системы рекомендации и Slack-подобный метод
- Построить систему с использованием мультимодальных данных и сравнить с бейзлайнами

Системы рекомендаций

- По рекомендации каналов была найдена только работа от Slack [4]
- В области рекомендаций каждый год появляются нейронные модели [5], однако на практике часто используются более простые методы:
  - Основанная на пользователях коллаборативная фильтрация [6]
  - Матричная факторизация [7]
  - Векторизация Node2Vec [8]

5. Shuai Zhang et.al., Deep Learning Based Recommender System: A Survey and New Perspectives, 2018
6. Schafer J.B. et.al., Collaborative Filtering Recommender Systems, 2007
7. Steffen Rendle et.al., BPR: Bayesian personalized ranking from implicit feedback, 2009
8. Jinyin Chen et.al., N2VSCDNNR: A Local Recommender System Based on Node2vec and Rich Information Network, 2019

Сбор данных

Space данные

С платформы Space была собрана информация о пользователях, каналах, репозиториях и структура организации.

- Полная структура организации
- ~ 2000 пользователей
  - < 30% имели хоть одну подписку
  - < 10% имело более двух подписок
- ~ 200 каналов
  - < 100 с подписчиками и активные за три месяца
- ~ 2000 репозиториев

Другие данные

GitHub: ~ 700 репозиториев, сопоставление со Space через email адреса в профилях

Slack: планировалось получить ~1500 публичных каналов.

- Нет открытого API, нужен доступ для работы
- Для этого нужно организовать безопасное окружение для работы с данными

Модели

Тестирование

- Были использованы только
  - каналы с не менее 3 подписчиками (<100)
  - пользователи с не менее 3 подписками (<200)
- Тест: самые поздние 3 подписки каждого пользователя и 0.2 всех “негативных” каналов
- Предсказание: топ 3
- Метрики: precision@k, map, roc auc

Унимодальные системы

Лучшие из моделей на каждом типе данных.

Коллаборативная фильтрация (**CF**), Node2Vec (**n2v**), Slack-подобный способ (**slack**): мало данных о подписках, при усреднении оценки близки к нулю.

Матричная факторизация Bayesian Personalized Ranking (**BPR**): мало сущностей, низкие размерности, а значит не нужно много данных.

	Pr@3	MAP	ROC AUC
cf_channel	17	18	43
cf_repository	15	18	50
cf_role_team	22	22	56
bpr_channel	37	27	77
bpr_repository	12	16	40
n2v_channel	18	19	45
n2v_repository	17	19	51
n2v_role_team	20	21	53
slack_thread	18	17	49

Мультимодальная система

Градиентные решающие деревья (XGboost) на лучших результатах бейзлайнов (10 факторов вместо 60) + несколько статистик.

	Channel	Structure	Repositories
	#subscribers, #subscriptions, activity-[thread|message]-2021-05-01, activity-[thread|message]-2020-05-01		bm25-repository2channel, bm25-channel2repository
User Based Collaborative Filtering		cf_team, cf_role, cf_team_role	cf_repository
Matrix Factorization	bpr_channel, bpr_[thread|message]_2020-05-01		bpr_repository
Node2Vec		n2v_struct	n2v_repository

Результаты

Сравнение с бейзлайнами

На тренировочных данных roc auc мультимодальной системы >90%.

Решающие деревья переобучаются.

Значимость факторов коррелирует с эффективностью их моделей.

	Pr@3	MAP	ROC AUC
xgb-all	34	25	66
bpr_channel	**37**	**27**	**77**
bpr_thread_2020-05-01	30	23	76
topk_subscribers_size	20	18	44
activity_thread_2020-05-01	11	15	52
cf_role_team	22	22	56
cf_team	19	20	53
cf_role	12	15	46
BM25_repo2channel	21	20	56
cf_repository	15	18	50
bpr_repository	12	16	40

Погрешность: 1%

Сравнение модальностей

Использование различных типов данных помогает бороться с переобучением.

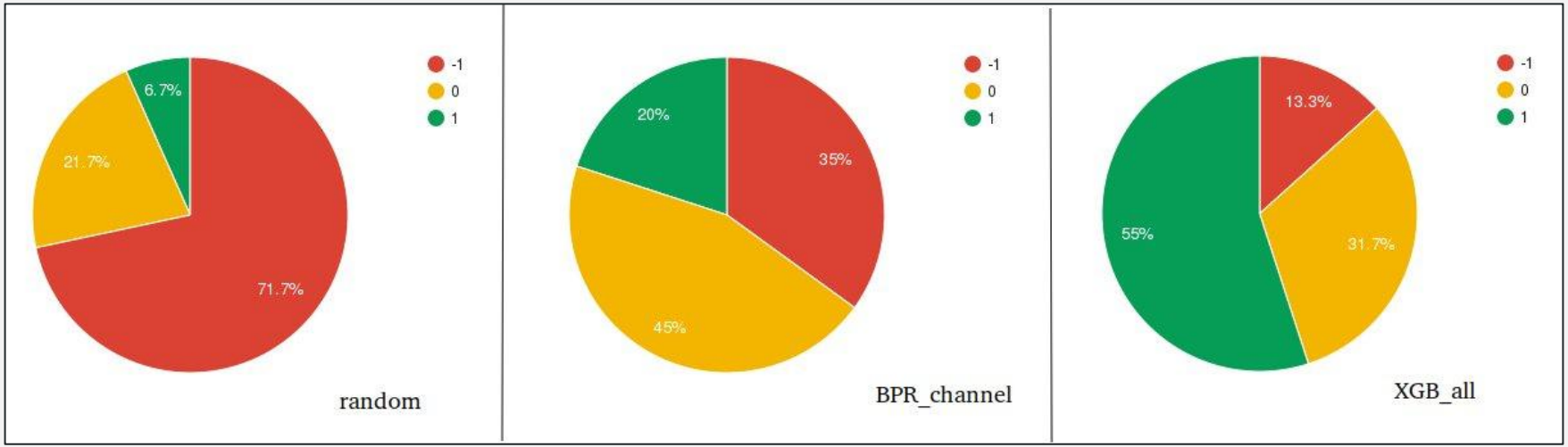
	Pr@3	MAP	ROC AUC test	ROC AUC train
bpr_channel	37	27	77	72
xgb-all	34	25	66	93
xgb-channel+structure	31	23	62	95
xgb-channel+project	29	21	61	96
xgb-channel	25	17	57	98

Погрешность: 1%

Оценка пользователями.

По 3 рекомендации от трёх моделей. 20 участников опроса.

Оценка: -1 (нерелевантно), 0 (релевантно, но недостаточно), 1 (подписка).



Итог

- Построены датасеты с разными типами данных
  - Можно переиспользовать для других алгоритмов поддержки
- Построен набор стандартных моделей рекомендации
  - Лучший результат — матричная факторизация Bayesian Personalized Ranking на подписках
- Построена мультимодальная система рекомендации. Использование нескольких типов данных
  - помогает бороться с переобучением
  - предположительно, решает проблему холодного старта пользователей

Другое

Space данные

- **Каналы** — социальные взаимодействия и общие интересы пользователей — название и описание, треды с текстовыми сообщениями, список подписчиков.
- **Структура** — позиция пользователя в компании — менеджер, команда, роль
- **Репозитории** — информация о профессиональных интересах и экспертизах пользователей, взаимодействие в рамках проектов — название и описание, исходный код, теги, список авторов, коммиты.

Схожесть пользователей

$$user2user\_*[us_1][us_2] = \frac{\sum_{ch \in C(us_1)} w\_*(us_1, us_2, ch)}{|C(us_1)|},$$

$$w\_channel(us_1, us_2, ch) = \{us_1, us_2\} \in S(ch),$$

$$w\_thread(us_1, us_2, ch) = \frac{\sum_{thr \in Thr(ch)} \{us_1, us_2\} \in S(thr)}{|Thr(ch)|} \cdot (\{us_1, us_2\} \in S(ch)),$$

$$w\_message(us_1, us_2, ch) = \frac{\sum_{thr \in Thr(ch)} \frac{|Mes(thr, us_1)| + |Mes(thr, us_2)|}{|thr|}}{|Thr(ch)|} \cdot (\{us_1, us_2\} \in S(ch)),$$

$$w\_mention(us_1, us_2, ch) = \frac{\sum_{thr \in Thr(ch)} \frac{|Ment(thr, us_1, us_2)|}{|thr|}}{|Thr(ch)|} \cdot (\{us_1, us_2\} \in S(ch)),$$

Схожесть пользователей

$$user2user\_*[us_1][us_2] = \frac{\sum_{ch \in C(us_1)} w\_*(us_1, us_2, ch)}{|C(us_1)|},$$

$$user2user\_repository[us_1][us_2] = \{us_1, us_2\} \in S(repo)$$

$$user2user\_structure[us_1][us_2] = \{bit\_manager, bit\_role, bit\_team, bit\_lead\}$$

Релевантность каналов

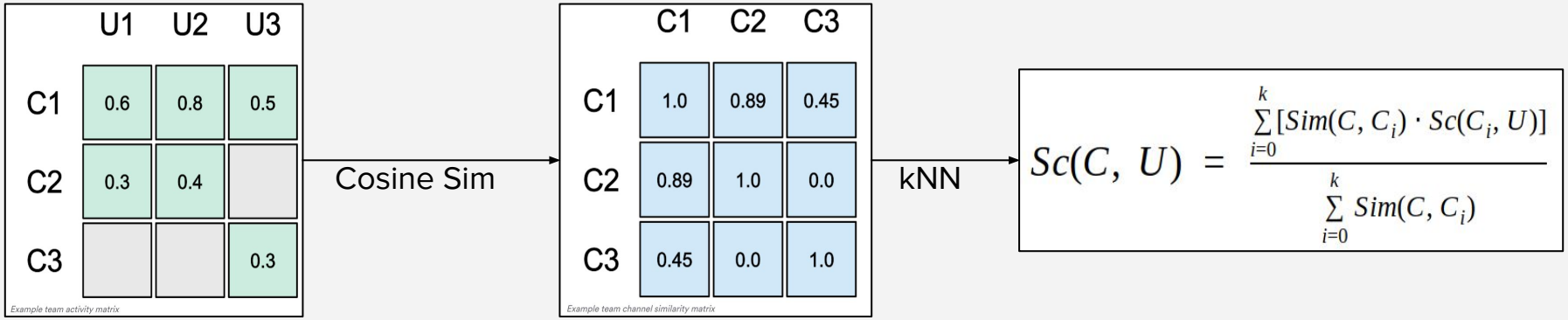
$$user2channel\_channel[us][ch] = us \in S(ch),$$

$$user2channel\_threadl[us][ch] = \frac{\sum_{thr \in Thr(ch)} us \in S(thr)}{|Thr(ch)|} \cdot (us \in S(ch)),$$

$$user2channel\_message[us][ch] = \frac{\sum_{thr \in Thr(ch)} \frac{|Mes(thr, us)|}{|thr|}}{|Thr(ch)|} \cdot (us \in S(ch)),$$

$$user2rep\_repository[us][repo] = us \in S(repo),$$

Slack алгоритм



1. Релевантность канала пользователю как отношение времени чтения канала к написанию в канал.
2. Косинусная смежность векторов матрицы релевантности как схожесть каналов.
3. Итог — среднее по самым близким каналам, взвешенное на релевантность пользователю.

Slack-подобный алгоритм

Нет информации о времени, проведенном пользователем в Space, поэтому используется активность в сообщениях и тредах.

	Precision@3	MAP	ROC AUC
slack_thread_2020-05-01	18	17	49
slack_thread_2020-11-01	17	17	47
slack_thread_2021-02-01	15	16	47
slack_thread_2021-05-01	10	14	43
slack_message_2020-05-01	16	16	46
slack_message_2020-11-01	14	15	46
slack_message_2021-02-01	12	15	44
slack_message_2021-05-01	10	13	42

Коллаборативная фильтрация на пользователях

Каналы: схожесть на уровнях подписок, общих тредов, сообщений и упоминаний. Всё это за разные промежутки времени.

Репозитории: пересечение соавторства в репозиториях.

Структура: схожесть по команде, роли, команде+роли.

	Pr@3	MAP	ROC AUC
cf_channel	17	18	43
cf_thread_2020-05-01	18	22	44
cf_thread_2020-11-01	18	21	44
cf_thread_2021-02-01	17	21	43
cf_thread_2021-05-01	14	18	44
cf_message_2020-05-01	17	21	44
cf_message_2020-11-01	16	21	44
cf_message_2021-02-01	16	20	43
cf_message_2021-05-01	13	18	43
cf_mention_2020-05-01	15	17	46
cf_mention_2020-11-01	15	17	45
cf_mention_2021-02-01	15	16	45
cf_mention_2021-05-01	11	14	45
cf_repository	15	18	50
cf_team	19	20	53
cf_role	12	15	46
cf_role_team	22	22	56

Матричная факторизация

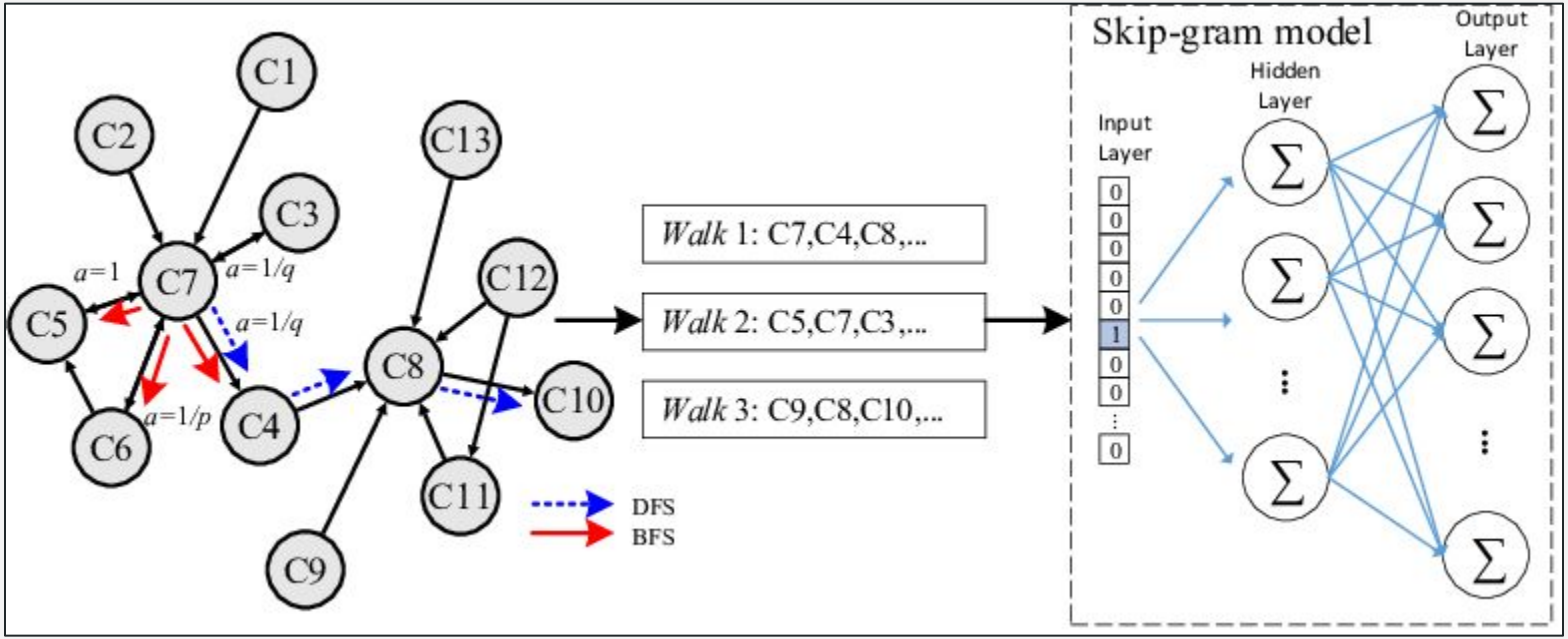
Модели: Alternating Least Squares (als), Bayesian Personalized Ranking (bpr) и Logistic Matrix Factorization (lmf).

Каналы: релевантность на уровнях подписок, общих тредов, сообщений. За разные промежутки времени.

Репозитории: авторство

	Pr@3	MAP	ROC AUC
bpr_channel	37	27	77
bpr_thread_2020-05-01	30	23	76
bpr_message_2020-05-01	29	23	74
bpr_repository	12	16	40
als_channel	27	21	61
als_thread_2020-05-01	24	20	57
als_message_2020-05-01	22	18	57
als_repository	13	16	45
lmf_channel	29	22	64
lmf_thread_2020-05-01	25	21	61
lmf_message_2020-05-01	24	17	59
lmf_repository	10	16	43

Node2Vec


C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
a=1
a=1/q
a=1/q
a=1/p
Walk 1: C7,C4,C8,...
Walk 2: C5,C7,C3,...
Walk 3: C9,C8,C10,...
DFS
BFS
Skip-gram model
Input Layer
Hidden Layer
Output Layer
0
0
0
0
0
1
0
0
0