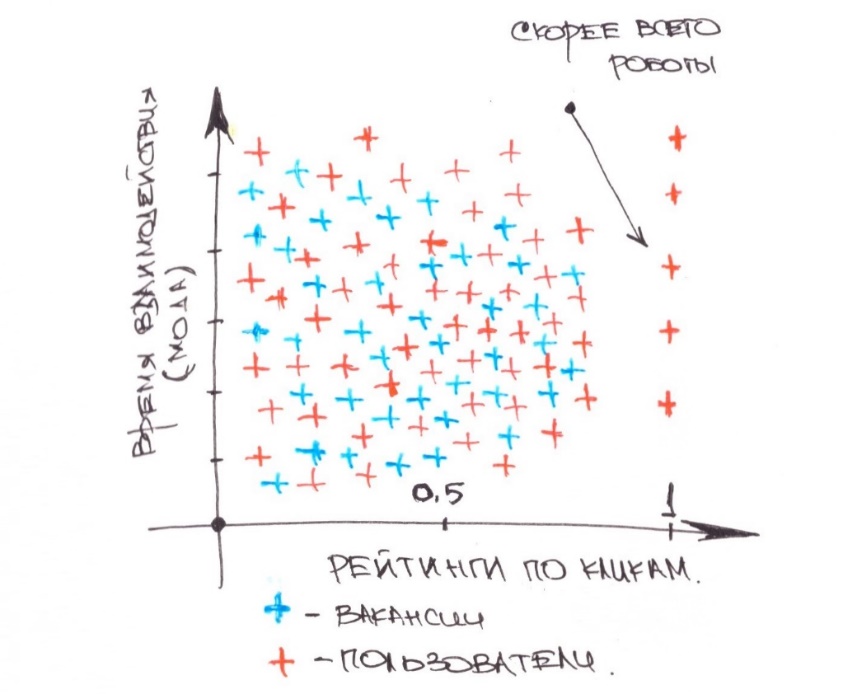
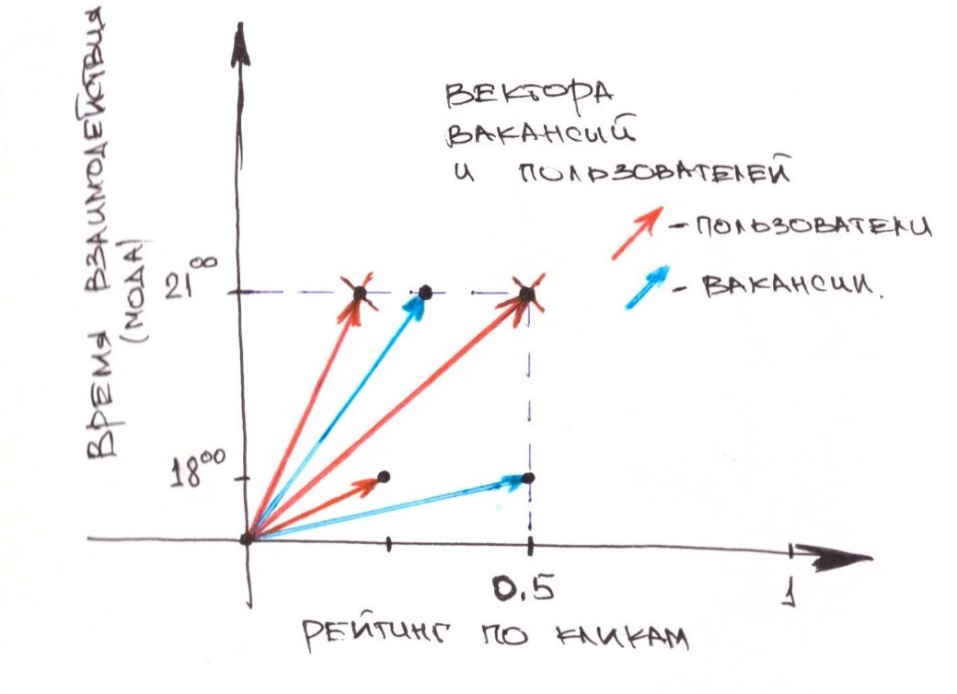
**Замысел метода улучшения показателей модели с помощью векторного сдвига**

1. **Выбор похожего**

Выбор пользователем вакансии определяется насколько она соответствует его пожеланиям. Максимально приятный вариант, когда вакансия полностью отвечает ожиданиям. Например, если человек ищет работу юриста с заработной платой в 100 000 рублей, то очевидно, что он с гораздо большей вероятностью откликнется на вакансию «юрист. ЗП 90 000», чем«бухгалтер. ЗП 50 000». Указанные пожелания и вакансии можно представить в виде векторов: желание [1, 100 000], предложения [1, 90 000] и [0, 50 000]. При такой реализации и пожелания и предложения находятся в одном векторном пространстве и очень просто найти наиболее похожие. В данных, которые были даны в задаче конечно, нет ни наименований профессии, ни указания на размер заработной платы, но тем не менее некоторые метрики мы можем из таких данных получить. Например, мы можем получить среднее количество положительных взаимодействий. Для вакансии это будет по сети срез ее популярности, а для пользователя эта цифра может быть характеристикой его формы активности: заходит посмотреть или реально ищет работу (+ так мы скорее всего сможет выявлять роботов). Возьмем еще один срез время взаимодействия (мода).

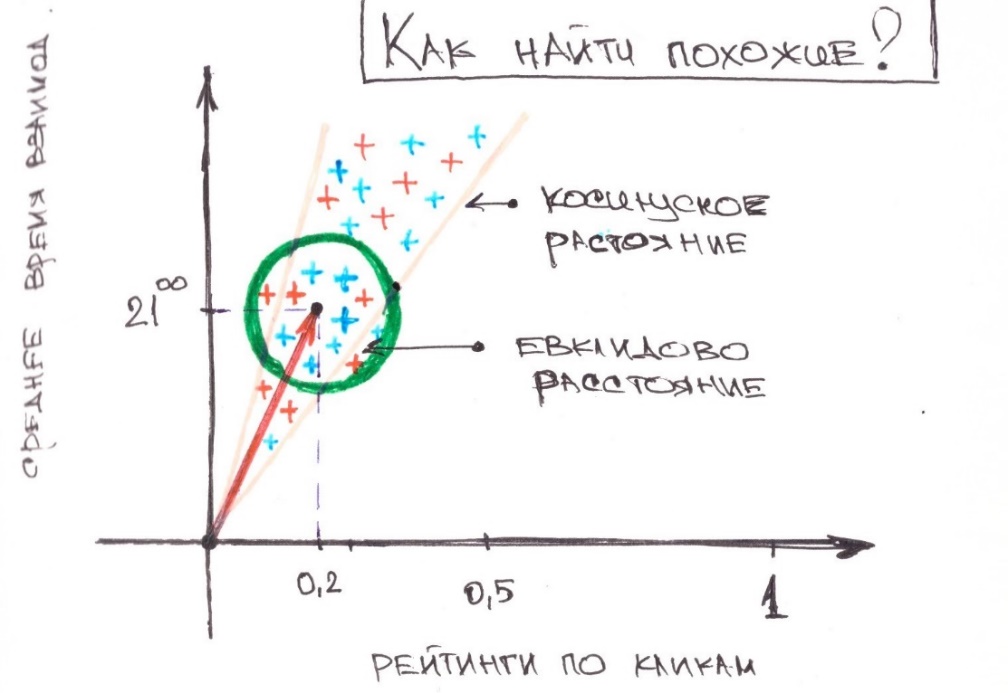
****

Если человек смотрит вакансию вечером в районе 21:00, то скорее всего он только что зашел домой с работы, которая ему очень не нравится. Для вакансий если на нее реагируют чаще всего в это время, то она чем-то привлекает людей, которые склонны выбирать новую работу в 21:00. Все это меры схожести интересов пользователей и предлагаемых вакансий.

****

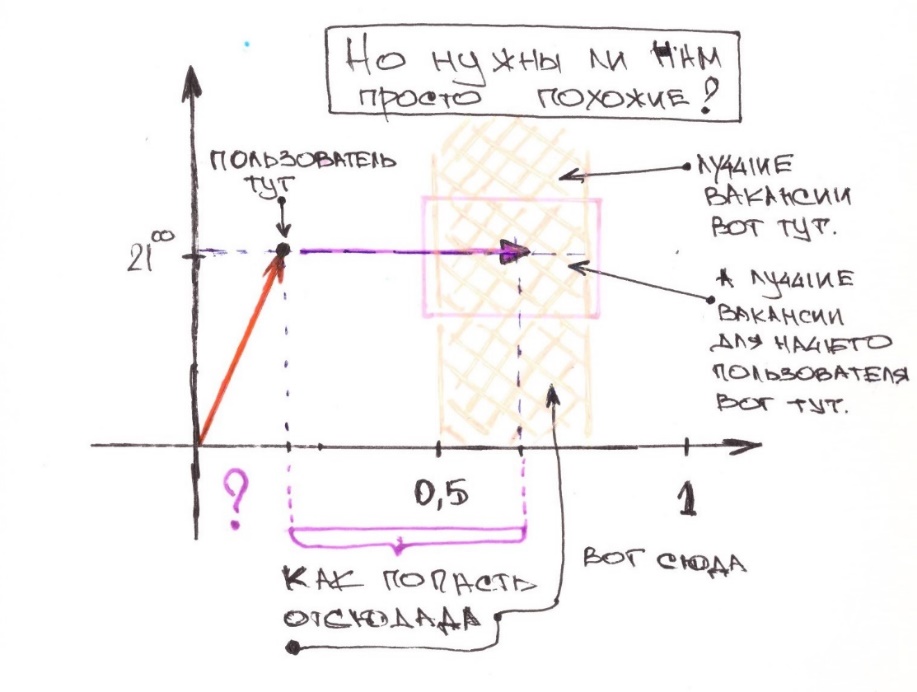
**2. Как найти схожие пожелания и вакансии?**

Наиболее популярный подход для поиска схожести в векторном пространстве, это нахождение евклидова расстояния и косинусного расстояния.

****

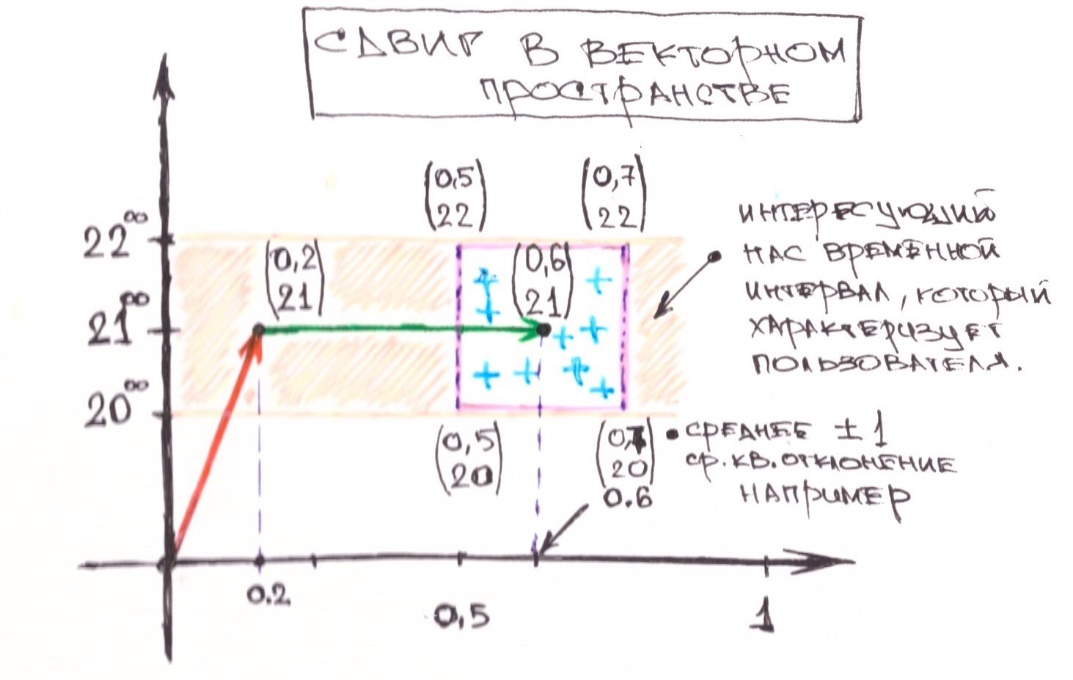
Реализовать это достаточно просто, но обратите внимание на рисунок, очевидно, что для человека, который только что начал пользоваться сервисом вне зависимости от метода мы получим список совсем не популярных вакансий, так как его рейтинг кликов будет близок к нулю (так сказать холодный старт). Очевидно, что нам нужны не то что бы максимально одинаковые, но похожие в чем, а в чем-то выгодно отличающиеся.

В нашем конкретном примере, мы хотим сохранить сходство по характеристике смотрит вакансии в 21:00 (+ - 1 час), но хотим получит те вакансии, на которые часто откликаются, или по-другому с более высоким средним рейтингом по кликам. Именно из этого массива нам и стоит подбирать вакансии для рекомендаций.

****

**3. Сдвиг в векторном пространстве**

Как нам попасть в область популярных вакансий сохранив при этом подчеркнутую временную особенность пользователя? Оказывается, достаточно просто – необходимо просто поменять нужную нам координату вектора, сохранив все остальные. В нашем случае это координата средний рейтинг по кликам, мы как бы сдвигаем рамку наших предпочтений 21:00 +/- час в сторону искомой координаты (вычислить ее можно по-разному, например, максимум – минус 1 среднеквадратичное отклонение)

****

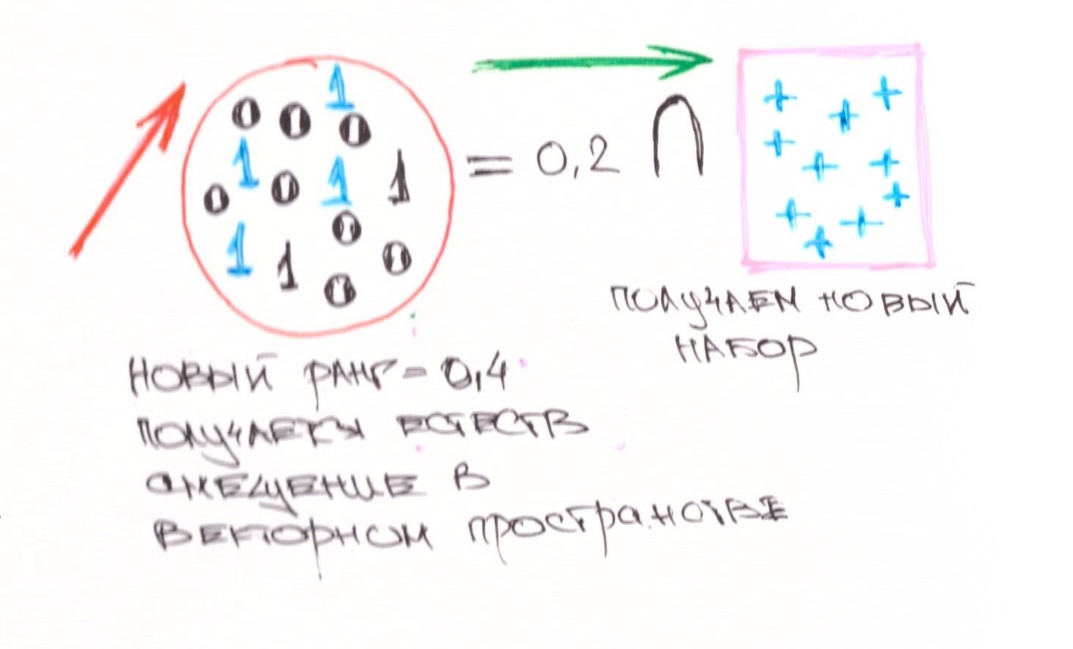
Проведенные исследования показали, что рамка и сдвиг сохраняют понятие ближайших похожих и для косинусного и для евклидова расстояний. В таблицах ниже приведены значения косинусных расстояний (левая таблица) и евклидовых расстояний (правая таблица) и применена тепловая карта для поэлементных смещений для пользователя которому из 10 вакансий понравились 2 и чаще всего он смотрел объявления в 21:00



Таким образом, сдвиг в сторону большего искомого значения не выходит за пределы идеи по поиску похожих векторов. Но что самое замечательное, он вычислительно прост. Нет смысла вычислять расстояния, достаточно просто установить фильтры на таблицу с данными и получить нужный нам набор вакансий.

**4. Сдвиг как периодический бурст**

Такой сдвиг можно применять не всегда, он актуален для холодного старта, и позволяет быстро нарастить показатель по кликам и естественным образом сдвинуть вектор пожеланий в область среднего. Например, наш пользователь посмотрел 10 вакансий и ему понравились 2, его рейтинг = 0.2. Мы применяем к нему сдвиг его вектор смещается и в рекомендации ему попадают ТОП-овые вакансии для его характеристики «время» (а возможно это и заработная плата, и должность, и характер работы, координат в векторе может быть очень много). В таком изобилии действительно хороших рекомендаций пользователей положительно взаимодействует еще с 4, то есть из 14 ему понравились 6 вакансий, его рейтинг естественным образом сдвигается до 0,4 и вот его окружают уже хорошие вакансии и сдвиг ему пока не нужен, до тех пор, пока его рейтинг снова не опустится до какой-нибудь оговоренной отметки

****

**По сути такой алгоритм является немного усложненной версией baseline алгоритма.**