

# **Разработка и анализ системы формирования плейлистов музыки**

Студент: Артём Соболев, 444 гр.  
Руководитель: Андрей Данильченко

# Рекомендательные системы

Формировать непрерывный поток музыки,  
которая

- понравится данному пользователю
- в данный момент времени

# Подход к рекомендациям

- Модель, описывающая прослушивания
- Используем по-максимуму всю доступную информацию
  - История воспроизведений пользователя
  - Отношения между треками (исполнители, альбомы, последовательность и др.)

# Задача

1. Разобраться в области
2. Реализовать модель из статьи <sup>[1]</sup>
3. Модифицировать модель
4. Подобрать оптимальные параметры
5. Сравнить результаты на нескольких наборах данных

[1]: Build Your Own Music Recommender by Modeling Internet Radio Streams  
O. Somekh, N. Aizenberg, Y. Koren

# Build Your Own Music Recommender ...

Вероятностная модель плейлистов онлайн-радиостанций

- `stationless`: учитываем только плейлисты, станций (пользователей) не существует

# Модель stationless

Вероятностная модель-аналог матричного разложения:

- Введём понятие сходства для плейлиста и трека

$$r(i, t, \mathcal{P}) = \underbrace{b_i}_{\text{popularity}} + \underbrace{v_i^{(1)}}_{\text{current}} \left( \underbrace{\frac{1}{\sqrt{|\mathcal{P}|}} \sum_{j \in \mathcal{P}} v_j^{(2)}}_{\text{station}} + \underbrace{\frac{1}{\sqrt{|\mathcal{P}_{[t-w, t]|}} \sum_{j \in \mathcal{P}_{[t-w, t]}} v_j^{(3)}}_{\text{context}} \right)$$

- Считаем, что данные порождены следующим распределением

$$P(\mathcal{T}, \Theta) = \prod_{\mathcal{P}_s \in \mathcal{T}} \prod_{(i, t) \in \mathcal{P}_s} \frac{\exp(r_{i, t, \mathcal{P}_s})}{\sum_j \exp(r_{j, t, \mathcal{P}_s})} \Lambda(\Theta)$$

# Таксономия

Вектор трека складывается из вектора характеристик исполнителя и вектора собственных характеристик трека

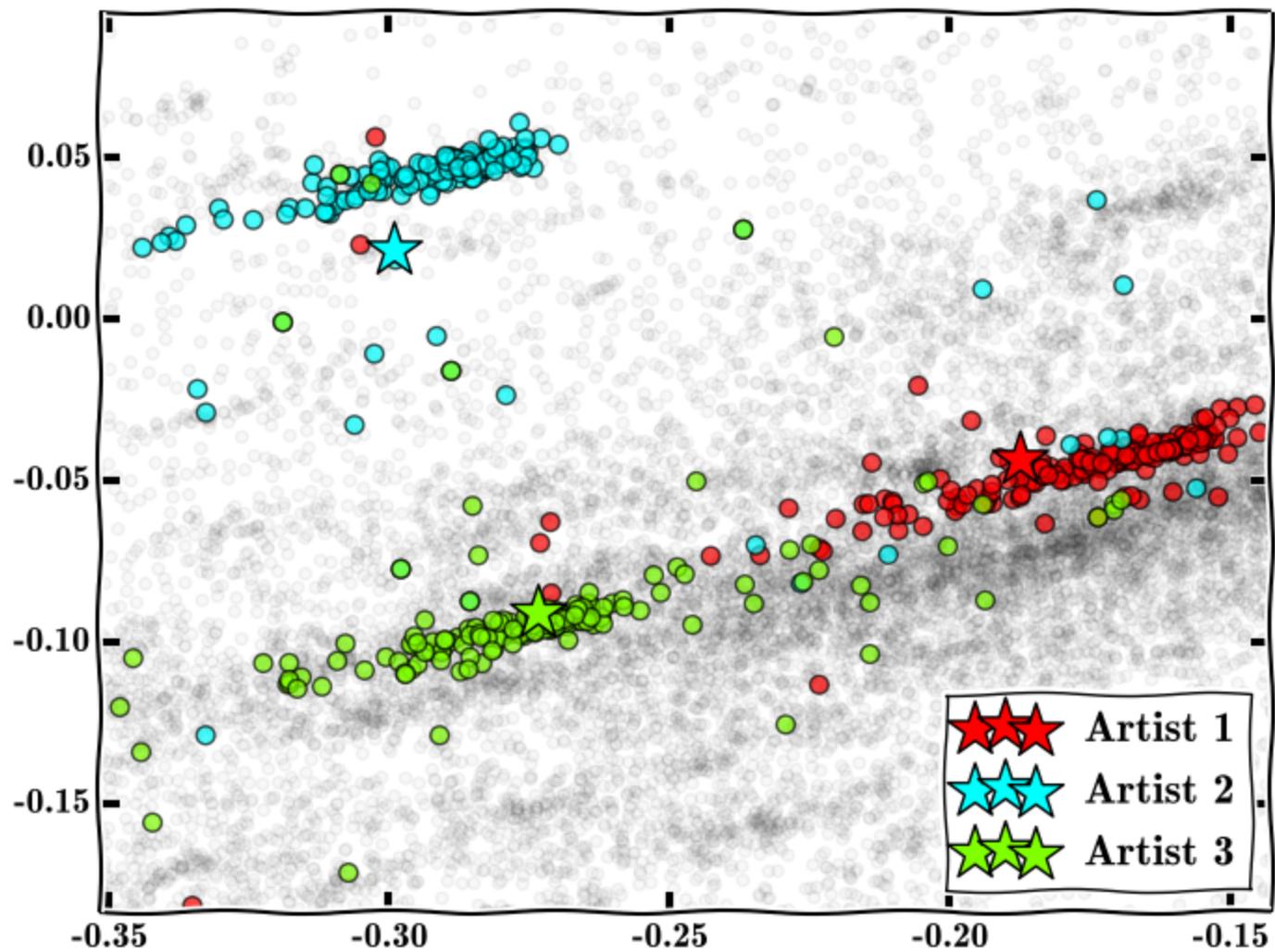
$$v_i^{(k)} = v_{t(i)}^{(k)} + v_{a(i)}^{(k)}$$

$$b_i = b_{t(i)} + b_{a(i)}$$

# Иной подход к таксономии

Хотим заставить параметры треков одного исполнителя быть “близкими”:

- Вектор исполнителя как среднее по его трекам
- Штраф за отклонение от вектора артиста
- Поддержка актуальности векторов исполнителей



# Обучение и подбор параметров

Стохастический градиентный подъём для логарифма правдоподобия

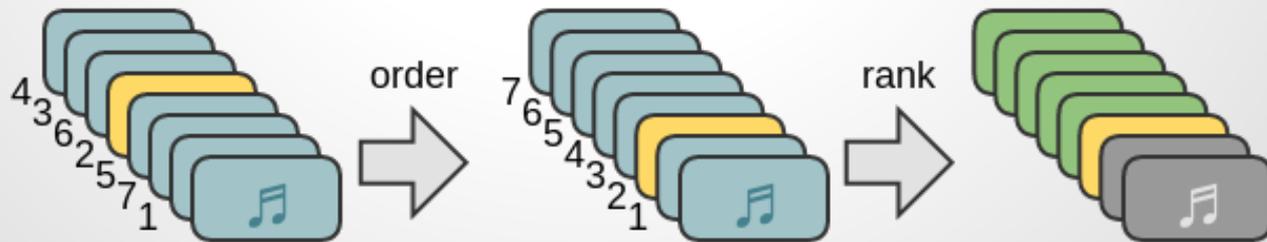
Поиск параметров по сетке на маленьком наборе данных

# Метрика качества: AUC@k

Как посчитать AUC@k по плейлисту:

- Из плейлиста удаляем последний трек
- Выбираем k случайных треков, которых нет в плейлисте
- Предлагаем алгоритму отранжировать список из (k+1) трека в зависимости от согласованности с плейлистом
- Результатом будет доля верно отранжированных пар

AUC@k получается как средний AUC по нескольким плейлистам



# Результаты

Dataset	Оригинал	Модификация
Yahoo R9	0.82065	<b>0.85725</b>
Moskva.FM	<b>0.84987</b>	0.79064
Piter.FM	0.77521	<b>0.80827</b>

- Yahoo R9 – набор данных из статьи
- Moskva.FM, Piter.FM – результат майнинга радиостанций Москвы и Питера

# Итого

1. Опробованы рекомендательные системы
2. Реализована модель из статьи
3. Реализована её модификация
  - а. Которая в некоторых случаях превосходит оригинальный метод
4. Проведено сравнение