Этап L0: проработка и формализация задачи, определение таргета.
Итог: определен таргет, определена мл задача

Решение. Общая постановка задачи

Определение задачи

Мы имеем дело с задачей построения рекомендаций, что является **частным случаем задачи ранжирования**, где запросом q является сам пользователь и его предпочтения. Факт заказа (или покупки) товара ещё не говорит о том, что товар понравился пользователю, поэтому фидбек неявный (implicit feedback). Однако количество покупок само по себе является важным отражением предпочтений пользователя, и это свойство учитывает ALS, который обсудим далее.

Решать такую задачу будем с помощью метода коллаборативной фильтрации, потому что взаимодействия пользователей несут самую важную информацию. В частности, будет необходимо факторизовать полученную матрицу взаимодействий пользователей с товарами. В качестве условного рейтинга товара взять количество покупок товара. Такая матрица разреженная: в ней много пустых ячеек. Для итогового ранжирования необходимо восстановить значения “предпочтений” в этой матрице. Такой метод называется матричной факторизацией.

Матричная факторизация

Разложим матрицу взаимодействий на 2 матрицы и перейдём в пространство меньшей размерности (подобно SVD, но сингулярная квадратная матрица для этой задачи нам не нужна): $R \approx U \times V^T$, где размерность матрицы R равна $[|Users| \times |Items|]$, для U – $[|Users| \times f]$, для V^T – $[f \times |Items|]$. f - это гиперпараметр, который будем подбирать по кросс-валидации, $|Users| = 1\,057\,265$, $|Items| = 6\,562$.

Такое представления исходной матрицы взаимодействий R позволяет выявить скрытые закономерности и поэтому данный подход относится к моделям со скрытыми переменными. Матрица U условно отвечает за свойства пользователя: по строкам отложены числовые значения, которые отражают “предпочтения” пользователя. Столбцы матрицы V^T - это такие же векторные числовые представления, но уже латентных качеств товара. Значения таких векторов не интерпретируемы: это “эмбединги” свойств пользователя и товара.

Переход от частоты покупок к уверенности предпочтений

Все пропущенные значения матрицы R заменим на 0. В этой связи необходима будут регуляризация, которую обсудим после того, как поговорим про обработку ненулевых элементов R.

Далее $r_{ui} \in R$, т.е. элементы матрицы R бинаризуем, и от r_{ui} перейдём к p_{ui} по следующему принципу:

$$p_{ui} = \begin{cases} 1, & \text{если } r_{ui} > 0 \\ 0, & \text{если } r_{ui} = 0 \end{cases}$$

Во работах [1] и [2] бинаризация “предпочтений” r_{ui} давала более хороший результат в сравнении с моделью с реальным значениями r_{ui} (в целом во всех прочитанных статьях делается такой переход; в этих работах наглядно показана разница в показателях метрик). В случае перехода к бинарному представлению значений матрицы взаимодействий нам понадобятся дополнительные веса, отвечающие за степень уверенности в том, что товар пользователю по вкусу.

Соберём функцию потерь

Перейдём к выводу функции потерь и отметим 2 её важных свойства.

Во-первых, мы заменили пропущенные значения нулями, и теперь имеем дисбаланс классов: ненулевых элементов, точнее говоря, элементов положительного класса (единиц) в матрице R всего 14 070 857 из 6 937 772 930 (0.2%), поэтому необходимо дать бОльшие веса ненулевым r_{ui} (или p_{ui} для модели с бинарными значениями в матрице предпочтений, которая будет интересовать в первую очередь). Так, каждое слогаемое функции потерь взвесим на c_{ui} (или w_{ui} в некоторых русскоязычных источниках): $c_{ui} = 1 + \alpha r_{ui}$, где α подбирается кросс-валидацией, авторы метода рекомендуют брать $\alpha \in [15; 40]$. Авторы[3] называют веса c_{ui} “уверенностью” в наблюдении p_{ui}: чем больше раз пользователь u купил товар i, тем с большей уверенностью можно говорить о том, что товар ему понравился. Мне кажется, что уместно сказать, что, задав таким образом веса, мы обращаем внимание, что ошибки, которые получаются на единицах, важнее ошибок на нулях.

Во-вторых, надо добавить регуляризацию для векторов матрицы со скрытыми свойствами пользователей и для товаров. Что на хабре, что в оригинальных статьях, все вносят обе нормы под один множитель λ, а не под разные, поэтому. Возможно, в дальнейшем следует попробовать разбить это составное слогаемое на 2 с разными множителями λ и μ, и тогда регуляризация будет иметь вид: $\lambda \sum_u ||x_u||^2 + \mu \sum_i ||y_i||^2$.

Чтобы не отходить от оригинальных обозначений, обозначим за x_u вектор скрытых предпочтений пользователя u (т.е. строка матрицы U в обозначениях выше), вектор скрытых свойств товара - за y_i. Наконец, можем выписать функцию потерь для алгоритма:

---

[1] http://yifanhu.net/PUB/cf.pdf
[2] https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe
[3] Статья в сноске №1

$$\sum_{u,i} c_{ui}(r_{ui} - x_u^T y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2) \to min(x_u, y_i)$$

Минимизация происходит сразу по двум переменным, поэтому вместо обычного градиентного спуска оптимизация будет выполняться с помощью ALS.

Alternating least squares

Поочерёдно фиксируем то x_u, то y_i и дифференцируем по другой переменной. Это и есть ALS. После дифференцирования[4] на каждом шаге итеративно обновляем значения в матрицах по формулам:

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p_u,$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p_i.$$

Оптимизация остановится, когда пройдёт установленное число итераций (в обеих работах за 10 шагов нашли оптимум).

Таргет

Когда найдены x_u для всех пользователей и y_i для всех товаров (другими словами, заполнены матрицы U и V, с помощью которых можно восстановить исходную матрицу R), можем восстановить пропущенные значения:

$$\hat{r}_{ui} = x_u^T y_i = \langle x_u, y_i \rangle$$

Затем полученные значения сортируем по убыванию и в качестве рекомендуемых товаров берём первые K айтемов. Так, **таргетом в нашей задаче являются элементы матрицы взаимодействий** r_{ui}, часть из которых дана, а пропущенные спрогнозируем по формуле выше (с колпачком).

Дополнительно про решение задачи

Алгоритм, рекомендующий самые популярные товары[5], можно взять в качестве базового[6], а пробовать его превосходить изложенной выше матричной факторизацией.

По кросс-валидации будут подбираться:

- f - количество скрытых свойств товара и предпочтений юзера, которые будут использованы при разложении;
- α - коэффициент во втором слагаемом весов c_{ui}, отвечающий за уверенность в предпочтениях пользователя;

---

[4] Выкладки можно посмотреть здесь: https://habr.com/en/companies/prequel/articles/567648/, причём в статье используются разные коэффициенты регуляризации для норм векторов.
[5] Как, например, в статье, также приведённой выше: http://yifanhu.net/PUB/cf.pdf
[6] https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093

- λ - коэффициент регуляризации. В целом, как отмечено выше, возможно следует попробовать регуляризовать с разными коэффициентами, и тогда подбирать будет надо λ и μ.