4. Задачи ранжирования

Введение

В этом уроке мы поговорим о задаче ранжирования.
И обсудим качество рекомендательных систем с точки зрения метрик ранжирования.

План

1 Что такое ранжирование

2 Метрики ранжирования

Метрики качества

Метрики для регрессии: MAE, RMSE

Метрики классификации: Precision/Recall

Метрика, ориентированная на пользователя: Охват, удержание пользователей, конверсии, клики, заказы

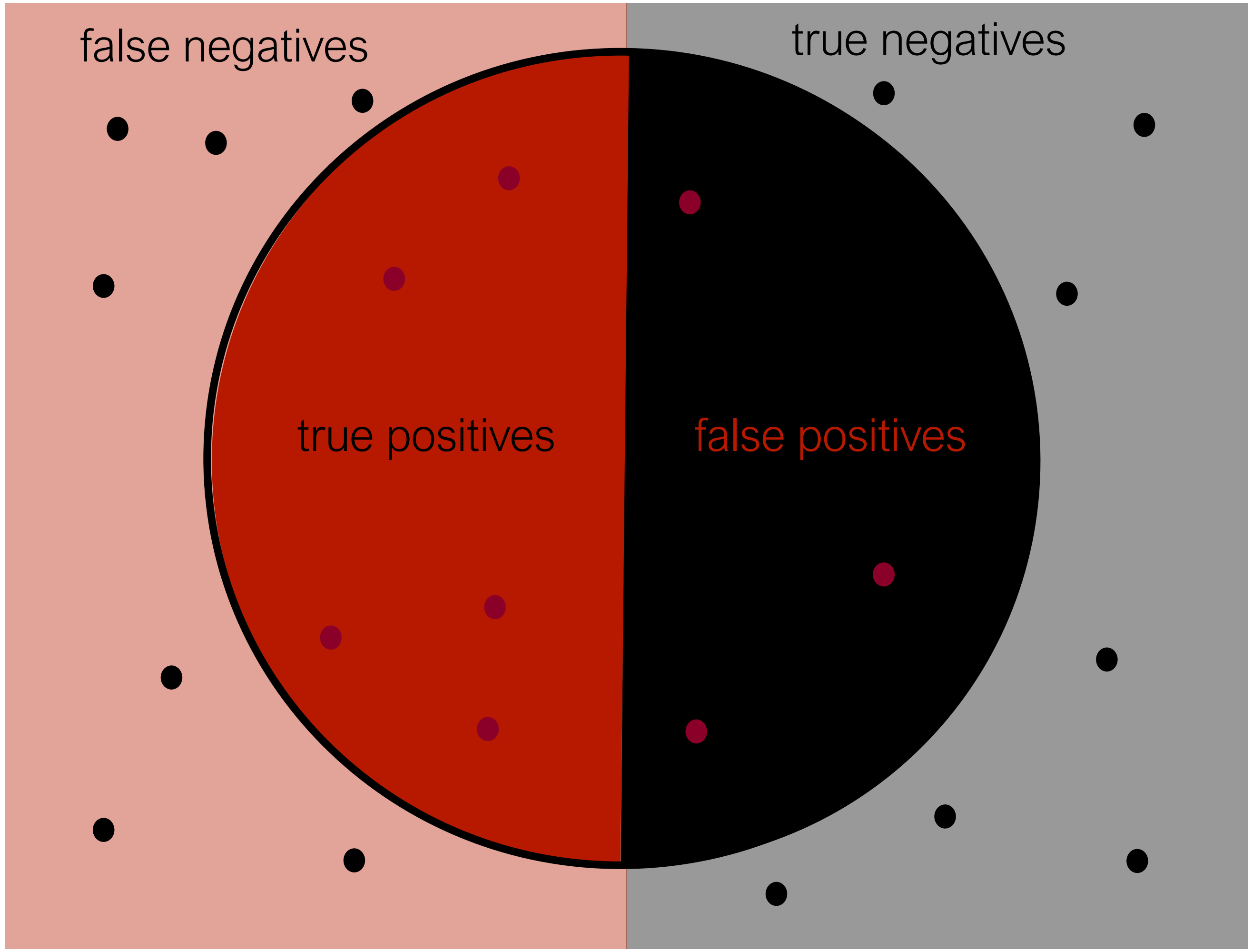
MAE

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|\widehat{x_i} - x_i|$$

RMSE

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\widehat{x_i} - x_i)^2}$$

Матрица ошибок



Accuracy

Accuracy – доля правильных ответов

$$\boxed{accuracy = \frac{TP + TN}{TP + FP + FN + TN}}$$

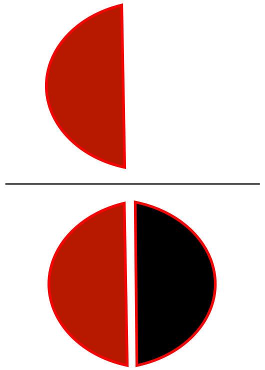
Задача классификации (classification)

Более информативными критериями являются точность **(precision)** и полнота **(recall)**.

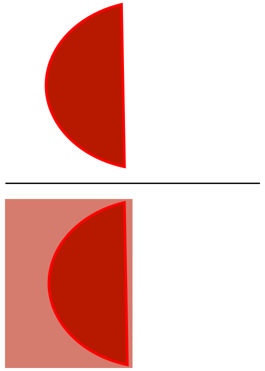
$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Precision =

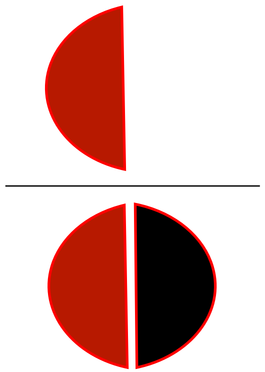


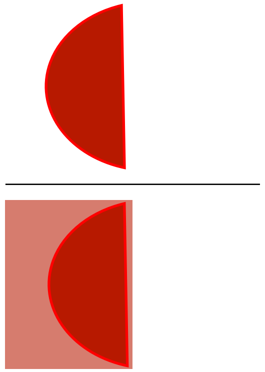
Recall =



Задача классификации (classification)

Precision - доля объектов, названных классификатором положительными и при этом действительно являющимися положительными
Recall - доля объектов положительного класса из всех объектов положительного класса.

Precision = 

Recall = 

Уроки Netflix Prize

Netflix Prize - это конкурс, в котором требовалось спрогнозировать оценку пользователями фильмотеки Netflix. Обучающие данные содержат 100 млн. оценок (от 1 до 5), которые 0,5 млн. клиентов поставили к 17 000 фильмам.

Точность прогноза оценивалась по RMSE.

Метрика:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\widehat{x_i} - x_i)^2}$$

x_i - истинное значение

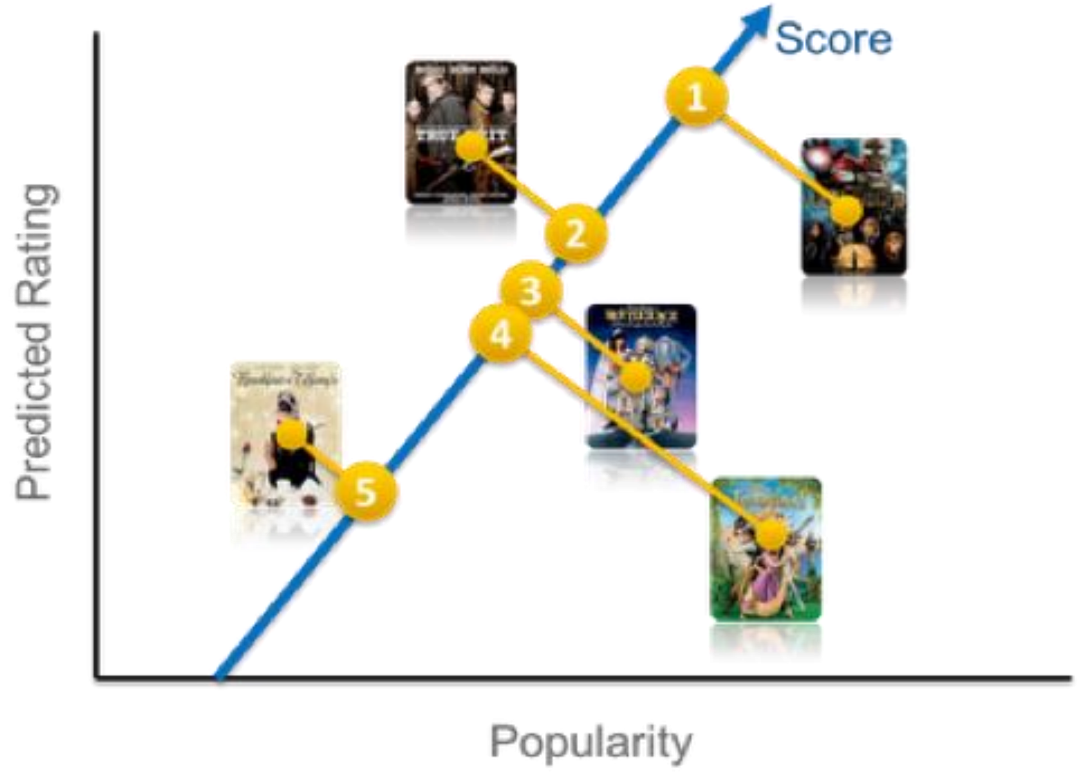
$\widehat{x_i}$ - оценка

Какую задачу мы решаем?

Задача ранжирования

Задача ранжирования

Ранжирование - это класс задач машинного обучения с *учителем*, заключающихся в автоматическом подборе ранжирующей модели по обучающей выборке, состоящей из множества списков и заданных частичных порядков на элементах внутри каждого списка.



Задача ранжирования

Ранжирование (Learning to Rank, LtR, L2R, LETOR) нужно везде, где система предоставляет пользователю выбор из большого числа вариантов.

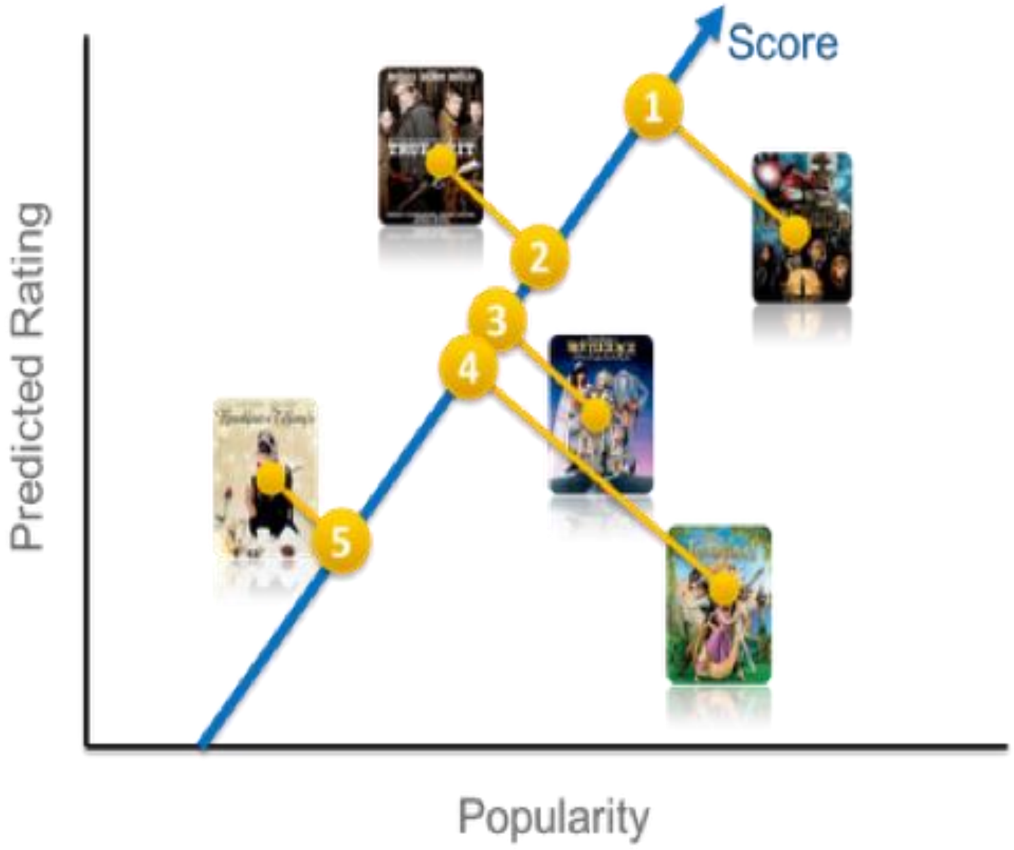
Примеры задач ранжирования

- Ранжирование выдачи поисковой системы
- Ранжирование рекомендаций пользователям (книги, фильмы, музыка и т.п.)
- Ранжирование вариантов саджестов
- Ранжирование вариантов перевода слов (в зависимости от контекста)

Примеры задач ранжирования

Примеры из жизни:

- Отсортировать (отранжировать) документы по релевантности
- Отсортировать письма по приоритету
- Отсортировать товары по вероятности покупки
- Предсказать места команд в чемпионате по футболу



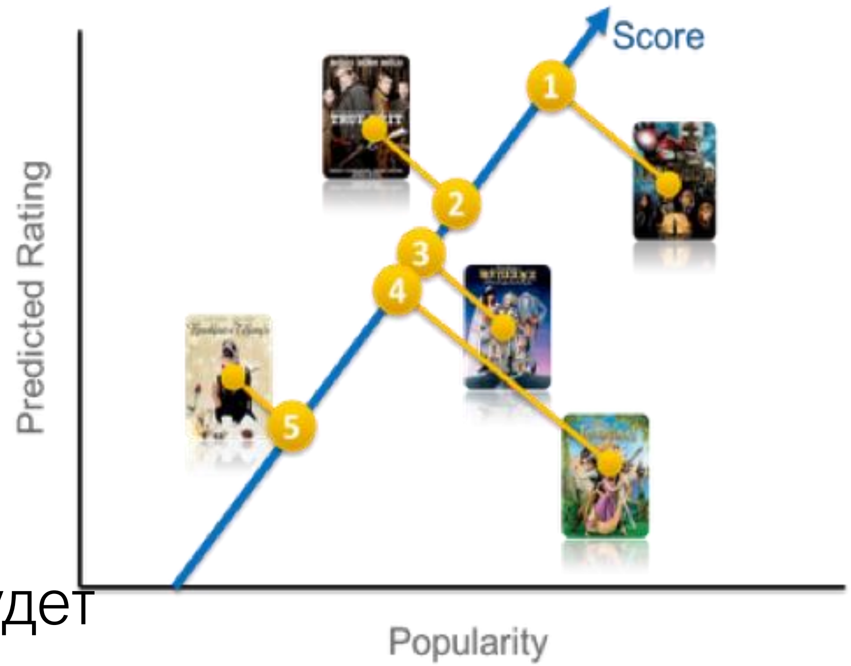
Давайте попробуем формально поставить такую задачу

Задача ранжирования

Дано:
- Объекты: $x_1, \dots, x_l$
- Порядок на *некоторых* парах: $\{(i, j) : x_i < x_j\}$

Найти: Ранжирующую модель $a(x)$, такую что

$$x_i < x_j \Rightarrow a(x_i) < a(x_j)$$



Проще говоря, нужно построить модель, который будет предсказывать парильный результат сравнения двух объектов. И если сравнить все объекты между собой, то можно будет получить их ранги (порядковый номер). И таким образом упорядочить их.

Задача ранжирования

Критерий конструируется по-разному в трех подходах:

- Point-wise – поточечный (аналог регрессии/классификации)
- Pair-wise – попарный (качество парных сравнений)
- List-wise – списочный (качество ранжированного списка)

Мы их рассмотрим чуть позже.

Давайте подумаем: как можно заменить задачу ранжирования задачей регрессии?

Ранжирование в рекомендациях

- В задачах рекомендаций порядок устанавливается для пар (пользователь, объект)
- Порядок задан для каждого пользователя, определяется независимо

Задачи ранжирования

Как правило, порядок задается явной обратной связью:

- Для контента: оценками
- Для товаров: купил / не купил

Когда данных мало используют неявную обратную связь:

- Для контента: время просмотра, частоту просмотра
- Для товаров: клики по товарам, добавления в корзину

Задачи ранжирования

Задачи ранжирования самостоятельный класс задач. Методы для решения задач классификации и регрессии, в общем, случае не подходят для решения задачи ранжирования.

Метрики ранжирования

Метрики ранжирования

1. Precision@n
2. Recall@n
3. MRR
4. NDCG@n
5. MAP

Обозначения

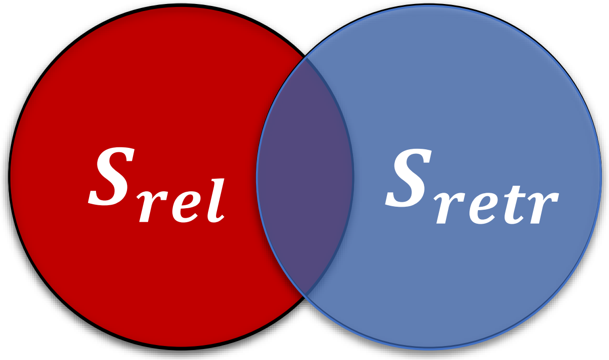
$rel(k) = \{0,1\}$ - релевантность k-го объекта. Бинарная функция, которая принимает значение 1, если объект релевантен и 0 – в противном случае.

u – будем обозначать пользователей.

Precision (в рекомендациях)

Precision - доля релевантных пользователю объектов относительно тех, которые ему показали.

$$\text{precision} = \frac{|S_{\text{rel}} \cap S_{\text{retr}}|}{|S_{\text{retr}}|}$$

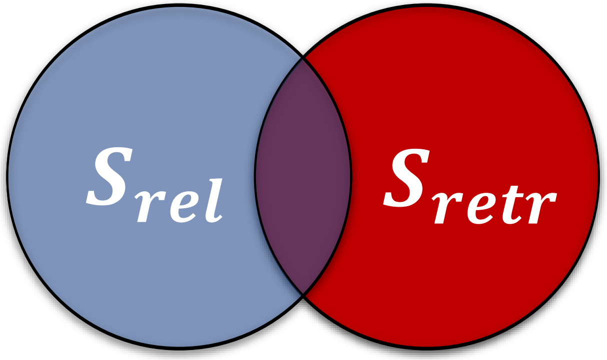


S_{rel} – множество релевантных пользователю объектов
S_{retr} – множество показанных пользователю объектов.

Recall (в рекомендациях)

Recall - доля релевантных объектов, показанных пользователю, относительно всех релевантных объектов.

Эту метрики можно интерпретировать, как вероятность того, что релеватный объект будет показан пользователю.



$$recall = \frac{|S_{rel} \cap S_{retr}|}{|S_{rel}|}$$

Метрики ранжирования

Recall@n

Для одного пользователя:

$$\boxed{Recall@n(u) = \left[\sum_{k=1}^{n} rel\ (u,k) > 0\right]}$$

Для всех пользователей:

$$\boxed{AvgRecall@n(u) = \frac{1}{U}\sum_{u=1}^{U} Recall@n\ (u)}$$

Recall@n – доля из первых n релевантных объектов, показанных пользователю, относительно всех релевантных объектов.

Метрика не учитывает ни порядок, ни количество релевантных объектов.

$AvgRecall@n$ – усреднение по всем пользователям.

Метрики ранжирования

Precision@n – это доля релевантных пользователю объектов из первых n объектов.

$$Precision@k = \frac{|S_{rel}^n \cap S_{retr}^n|}{|S_{retr}^n|} = \frac{|S_{rel}^n \cap S_{retr}^n|}{k}$$

$$\boxed{AvgPrecision@n = \frac{1}{n}\sum_{k=1}^{n} Precision@k \cdot rel(k)}$$

$AvgPrecision@n$ – показывает среднюю точность для первых n объектов. Таким образом эта метрика учитывает порядок документов.

где k – позиция объекта в списке рекомендаций длины n.

Метрики ранжирования

MAP - Mean Average Precision
MAP помимо усреднения по n, дополнительно усредняет по всем пользователям или по всем запросам (u).

Это делается из предположения из соображения, что все пользователи или запросы равноценны.

MAP является достаточно популярной, учитывает и порядок, и количество релевантных объектов.

$$MAP@n = \frac{1}{U}\sum_{u=1}^{U} AvgPrecision@n(u)$$

Метрики ранжирования

MRR - Mean Reciprocal Rank

$rank_u$ - означает
положение первого релевантного объект
для пользователя u.

$$MRR = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{rank_u}$$

Чаще используется для поисковых
запросов, где в место пользователя
фигурируют запросы.

Метрики ранжирования

Ответ	Релевантный ответ	Ранг
А,Б,**В**,Г,Д	**В**	3
Д,Б,А,Z,J	**Д**	1
Я,**Ю**,А,2,Т	**Ю**	2

$$MRR = \frac{1}{|n|}\sum_{i=1}^{n}\frac{1}{rank_i}$$

Усредненный оценка
(1/3+ 1/1 + 1/2)/3 = 0.61

Идеальная оценка 1.

Давайте придумаем «свою» метрику

Метрики ранжирования

nDCG@n - Normalized Discounted Cumulative Gain

DCG@n является популярной метрикой в информационном поиске. Она учитывает и порядок, и количество релевантных объектов.

nDCG@n – это нормированная метрика. nDCG@n = 1 означает, что объекты идеально отранжированны.

$$DCG@n = \sum_{k=1}^{n} \frac{rel(k)}{\log_2(k+1)}$$

$$nDCG@n = \frac{DCG@n}{IDCG@n}$$

$$IDCG@n = \sum_{k=1}^{n} \frac{1}{\log_2(k+1)}$$

- Нормировочная константа

Методы ранжирования

- Pointwise (поточечный)
- Pairwise (попарный)
- Listwise (списочный)

Pointwise

Pointwise (поточечный)

Вместо предсказания порядка, будем предсказывать некоторую метрику, которая задается числом, и на которой сохраняется отношение порядка.
Например, для контента – это предсказание самой оценки или времени потраченного на контент.

Обратите внимание, что с точки зрения задачи ранжирования.
Нам нужно оценить только порядок, а не саму оценку.

Задача ранжирования

Линейная модель ранжирования:

$$a(x, \mathrm{w}) = \langle x, w \rangle$$

где $x \longmapsto (f_1(x), \dots, f_n(x)) \in \mathbb{R}^n$ - вектор признаков объекта x

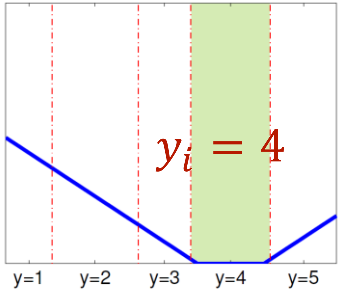
Ранговая (порядковая) регрессия (Ordinal Regression)

Взгляд как на регрессию:

- Обучающая выборка $(x_i, y_i)_{i=1}^{l}$, где $y_i \in Y = 1 \prec 2 \prec \cdots \prec K$
- Функция ранжирования с параметрами w и порогами $b_0 = -\infty, b_1, \dots, b_{K-1}, b_K = +\infty$:
- $a(x, w, b) = y$, если $b_{y-1} < g(x, w) \le b_y$
- Функция потерь $\mathcal{L}(M)$ - убывающая функция отступа M

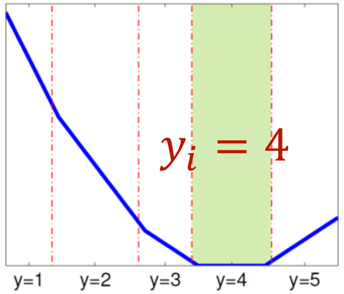
Критерий обучения по двум ближайшим порогам:

$$\sum_{i=1}^{l} \mathcal{L}(g(x_i, w) - b_{y_i - 1}) + \mathcal{L}(b_{y_i} - g(x_i, w)) \rightarrow \min_{w,b}$$



Критерия обучения по всем порогам:

$$\sum_{i=1}^{l} \sum_{y=1}^{K} \mathcal{L}\left((b_y - g(x_i, w))\, sign(y - y_i)\right) \rightarrow \min_{w,b}$$



Pairwise

Pairwise (попарный)

В этом случае задача обучения ранжированию аппроксимируется решением задач классификации.
Каждое сравнение пары объектов рассматривается, как отдельная задача классификации.
Цель состоит в том, чтобы свести к минимуму среднее число инверсий в рейтинге.

Попарный подход

Переход к гладкому функционалу качества ранжирования:

$$Q(w) = \sum_{i \prec j} [\underbrace{a(x_j, w) - a(x_i, w)}_{Margin(i,j)} < 0]$$

$$\leq \sum_{i \prec j} \mathcal{L}\big(a(x_j, w) - a(x_j, \mathrm{w})\big) \to \min_w$$

где a(x,w) – параметрическая модель ранжирования
$\mathcal{L}(M)$ – убывающая непрерывная (гладкая) функция отступа $Margin(i,j)$

- $\mathcal{L}(M) = (1 - M)_+$ – RankSVM
- $\mathcal{L}(M) = \exp(-M)$ – RankBoost
- $\mathcal{L}(M) = \log(1 + e^{-M})$ – RankNet

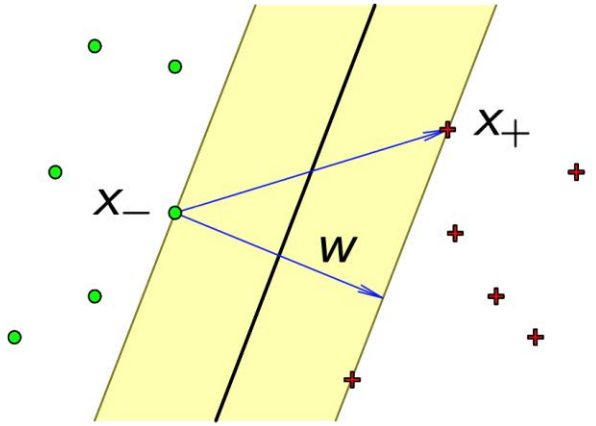
SVM – метод опорных векторов

Линейный классификатор, $Y = \{-1, +1\}$

$$a(x, w, w_0) = sign(\langle w, x\rangle - w_0), w, x \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

Задача обучения SVM:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\mathcal{E}_i \to \min\limits_{w, w_0, \mathcal{E}} \\ M_i(w, w_0) \geq 1 - \mathcal{E}_i, i = 1, \dots, l \\ \mathcal{E}_i \geq 0, i = 1, \dots, l \end{cases}$$



где $M_i(w, w_0) = y_i(\langle w, x\rangle - w_0)$ - отступ объекта x_i

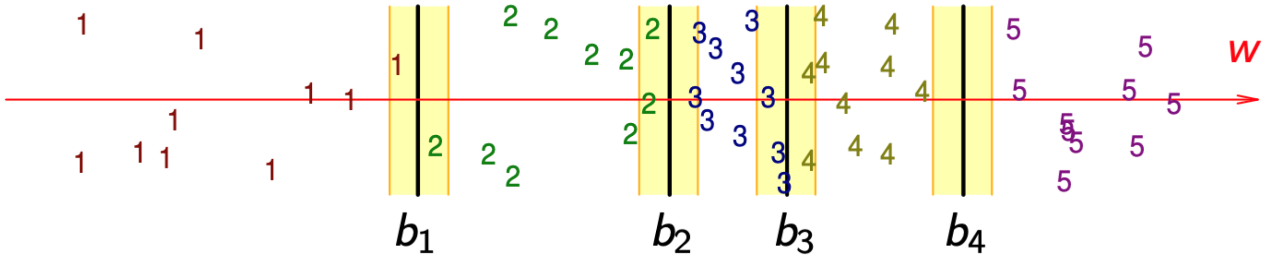
Эквивалентная задача безусловной минимизации:

$$\sum_{i=1}^{l}\left(1 - M_i(w, w_0)\right)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w, w_0}$$

Ранговая классификация OC-SVM (Ordinal Classification SVM)

Пусть $Y = \{1, \dots, K\}$ **функция ранжирования линейная с порогами** $b_0 = -\infty, b_1, \dots, b_{K-1}, b_K = +\infty$:

$$a(x, w, b) = y, \text{ если } b_{y-1} < g(x, w) \leq b_y$$



Постановка задачи SVM для ранговой классификации:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}[y_i \neq K](\mathcal{E}_i + \mathcal{E}_i^*) \to \min\limits_{w, w_0, \mathcal{E}} \\ b_{y_i - 1} + 1 - \mathcal{E}_i^* \leq \langle w, x_i \rangle \leq b_{y_i - 1} - 1 + \mathcal{E}_i \\ \mathcal{E}_i^* \geq 0, \mathcal{E}_i \geq 0 \end{cases}$$

Ranking SVM: метод опорных векторов для ранжирования

Постановка задачи SVM для попарного подхода

$$Q(w) = \frac{1}{2}\|w\|^2 + C\sum_{i \prec j} \underbrace{(a(x_j, w) - a(x_i, w))}_{Margin(i,j)} \to \min_w$$

где $a(x, w) = \langle w, x \rangle$ – линейная функция ранжирования
$\mathcal{L}(M) = (1 - M)_+$ – «шарнирная» функция потерь (hinge loss)
$M = Margin(i, j) = \langle w, x_j - x_i \rangle$ – отступ
Постановка задачи квадратичного программирования:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i \prec j} \mathcal{E}_{ij} \to \min_{w,\mathcal{E}} \\ \langle w, x_j - x_i \rangle \geq 1 - \mathcal{E}_{ij}, \quad i \prec j \\ \mathcal{E}_{ij} \geq 0, \quad i \prec j \end{cases}$$

Rank Net: логистическая регрессия для ранжирования

RankNet: гладкий функционал качества ранжирования

$$Q(w) = \sum_{i \prec j} \underbrace{(a(x_j, w) - a(x_i, w))}_{Margin(i,j)} \to \min_w$$

при $\mathcal{L}(M) = \log(1 + e^{-\sigma M})$ и линейной модели $a(x) = \langle w, x \rangle$
Метод стохастического градиента:
выбираем на каждой итерации случайную пару , $i \prec$ j

$$w \coloneqq w + \eta \cdot \frac{\sigma}{1 + \exp(\sigma\langle x_j - x_i, w \rangle)} \cdot (x_j - x_i)$$

Listwise

Listwise (списочный)

Списочный подход непосредственно пытаются найти оптимальный порядок для всего списка объектов. Но в такой постановке задачи присутствует дискретный функционал. Но с таким функционалом затруднительно работать напрямую, поскольку он не дифференцируется.

Listwise

Listwise (списочный)

RankNet, шаг стохастического градиентного спуска для линейной модели:

$$\boxed{w := w + \eta \frac{1}{1 + \exp\left(\langle w, x_j - x_i \rangle\right)} (x_j - x_i)}$$

Listwise

Listwise (списочный)

LambdaRank
Домножим стохастический градиент по паре $(\boldsymbol{x_j}, \boldsymbol{x_i})$ на $\Delta NDCG_{ij}$ (изменение NDCG при перестановке $\boldsymbol{x_j}$ и $\boldsymbol{x_i}$ местами):

$$\mathrm{w} \coloneqq \mathrm{w} + \eta \frac{1}{1 + \exp\left(\langle \mathrm{w}, \mathrm{x_j} - \mathrm{x_i} \rangle\right)} (\mathrm{x_j} - \mathrm{x_i}) \cdot$$
$$\cdot \, |\Delta \boldsymbol{NDCG_{ij}}| (\boldsymbol{x_j} - \boldsymbol{x_i})$$

Listwise

Listwise (списочный)

Существуют другие методы работы со списочным подходом и на данный момент это еще открытая область машинного обучения и почти каждый год выходят работы.
Их условно разделить на 2 типа:

1. Аппроксимация попарного подхода. Например, LambdaRank, SoftRank, AdaRank.

2. Использование рангом в явном виде. Иногда с использование специфики конкретных задача. Например ListNet, ListMLE.

Вопросы

Заключение

Давайте вместе напишем заключение:

- Узнали метрики ранжирования.
- NDCG самая крутая! Но и MAP неплоха.
- Можно придумать свои метрики ранжирования.
- Разные подходы:
  - Pointwise (поточечный)
  - Pairwise (попарный)
  - Listwise (списочный)

Семинар: метрики ранжирования