ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧАХ РАНЖИРОВАНИЯ В ПОИСКОВЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Борисовская А. А.[1]

(*Российский университет транспорта* (*МИИТ*)*, Москва*)

Задача ранжирования в современных поисковых информационных системах может рассматриваться как особый класс задач машинного обучения. По типу обучающей выборки она может быть отнесена к задачам классификации, по типу целевой функции – к задачам регрессии. Задача ранжирования состоит в отыскании верного ранжирующего отображения на основе обучающей выборки и заданного на ней верного порядка документов, найденных по данному запросу. Задача ранжирования в данном случае может быть решена с помощью методов опорных векторов и стохастического градиентного спуска. Метод опорных векторов может быть представлен задачей квадратичного программирования, в которой задействованы наборы весов признаков ранжирования. Алгоритм минимизирует целевую функцию, зависящую от неверно присвоенных признакам весов, т.е. от отступов между классами релевантности. Метод стохастического градиентного спуска основан на случайном выборе поискового запроса и пары документов и последующем градиентном шаге на основе этих документов. Результирующая ранжирующая функция может быть оценена с помощью таких метрик, как доля «дефектных» пар, средняя точность и нормализованная приведенная совокупная полезность. Однако не существует наилучшей метрики оценки качества полученной модели. Каждая метрика подходит для своей предметной области.

Ключевые слова: информационный поиск, ранжирование, машинное обучение.

1. Введение

Работа поисковой системы заключается в том, чтобы по запросу пользователя найти документы, содержащие либо указанные ключевые слова, либо слова, как-либо связанные с ключевыми словами. При этом поисковая система генерирует страницу результатов поиска [5]. Такая поисковая выдача может содержать различные типы результатов, например: веб-страницы, изображения, аудиофайлы.

[1] *Алена Алексеевна Борисовская, выпускник бакалавриата* (*borisovsk0101@gmail.com*).

То есть непосредственно процесс пользовательского поиска происходит следующим образом: пользователь формулирует поисковый запрос, состоящий из ключевых слов, наиболее точно задающих область поиска. Далее поисковая система отбирает веб-страницы (документы), содержащие ключевые слова или с ними связанные. Затем генерирует поисковую выдачу – список ссылок на найденные веб-документы. Финальный этап – оценка точности и полноты поиска.

Таким образом, основной задачей системы информационного поиска является нахождение, отбор и выдача информации из массива структурированных данных, наилучшим образом удовлетворяющей информационную потребность пользователя, выраженную в форме поискового запроса. Единицы информации могут быть представлены веб-страницами, текстовыми документы, таблицами БД, файлами формата JSON. При этом основными критериями качества результатов информационного поиска являются полнота, точность и оперативность поиска.

Поскольку мощность поисковой выдачи D_q может быть очень большой, ее элементы необходимо ранжировать по степени релевантности запросу q, а наиболее релевантные выдать пользователю [6].

Для решения этой проблемы широко применяются подходы машинного обучения, а именно, особый класс задач машинного обучения – обучение ранжированию с учителем.

2. Машинное обучение ранжированию

2.1. ПОСТАНОВКА ЗАДАЧИ

В общем случае задачу машинного обучения с учителем можно сформулировать так:

Дано обучающее множество из N пар вида:

(1) $X^N = \{(x_1, y_1), ..., (x_N, y_N)\}$.

Здесь x_i – вектор входных признаков i-го элемента, а y_i – его целевое значение.

Алгоритм обучения должен найти функцию g: $X \to Y$, где X – множество допустимых значений входных параметров, Y – множество возможных выходных значений алгоритма.

Функция g может быть выбрана из соображений минимизации эмпирического риска. Функция эмпирического риска характеризует среднюю ошибку модели g на обучающей выборке и может быть выражена формулой:

$$(2) \quad Q(g, X^N) = \frac{1}{N} \sum_{i=1}^{N} L(g(x_i), y_i) \cdot$$

Здесь $L(y, y')$ – заранее заданный функционал потерь при отклонении выходного значения модели от целевого значения в обучающей выборке.

Таким образом, задачей обучения является нахождение функции g, доставляющей минимум функционалу эмпирического риска:

$$(3) \quad g = \arg\min_{g} Q(g, X^N).$$

В ходе обучения ранжированию (см. рис. 1) автоматически подбирается ранжирующая модель по обучающей выборке, состоящей из множеств элементов и заданных на них отношений частичного порядка $\{y^{(1)}, \ldots, y^{(n)}\}$. Причем каждый элемент представлен парой «запрос – документ» так, что формируется несколько списков документов $\{d_1^{(i)}, \ldots, d_n^{(i)}\}$, найденных по данному запросу q_i. Отношение порядка на этих списках может быть задано с помощью бинарной (1 – релевантен, 0 – не релевантен) или вещественной оценки релевантности каждого документа, либо задано относительным порядком документов в парах (d_i, d_j). В результате обучения на основе эталонной выборки ранжирующая модель должна наилучшим образом упорядочивать новые, тестовые данные.

Общая постановка задачи ранжирования выглядит следующим образом. Дано $Q \times D$ – множество возможных пар «запрос, документ» (q, d), X – множество объектов, состоящих из запроса и списка соответствующих документов, $X^N = \{(q_1, d_1^{(1)}, \ldots, d_m^{(1)}, y^{(1)}), \ldots, (q_1, d_1^{(N)}, \ldots, d_m^{(N)}, y^{(N)})\}$ – обучающая выборка, где q_i – запрос, $\{d_1^{(i)}, \ldots, d_m^{(i)}\}$ – список документов,

найденных по нему, $y^{(i)}$ – отношение порядка на данном списке. Причем вектор $y^{(i)}$ состоит из вещественных чисел или {0, 1} в зависимости от способа оценки релевантности документа либо задан правильный порядок на парах $(d_p^{(i)}, \ldots, d_k^{(i)})$ из $d^{(i)}$. Также для дальнейшей оценки ошибки модели задан функционал потерь $L(y, y')$, характеризующий отклонение выходных оценок релевантности документов от эталонных значений в обучающей выборке.

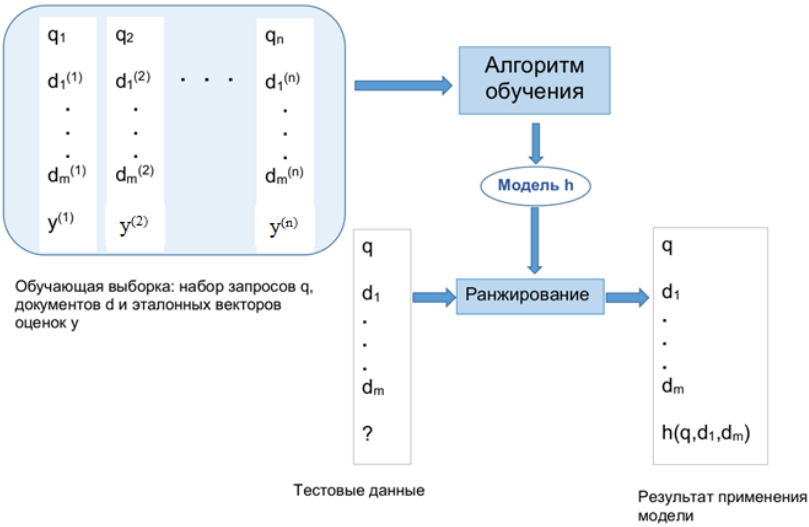

Рис. 1. Обучение ранжированию

Необходимо построить ранжирующее отображение a, восстанавливающее правильное отношение порядка на элементах «запрос, документ» из множества $Q \times D$. При этом $f^{(j)} = (f_1, \ldots, f_n)$ – вектор оценок признаков документа j по отношению к соответствующему запросу, $w^{(j)}$ – вектор весов признаков ранжирования. Тогда ранжирующее отображение является скалярным произведением вектора признаков на вектор весов $a(f, w) = \langle f, w \rangle$ и в результате дает вещественную или бинарную оценку релевантности соответствующего документа. Вектор

весов *w* является параметром задачи, подлежащим обучению и улучшению на каждом шаге алгоритма построения *a*.

2.2. КЛАССИФИКАЦИЯ ПОДХОДОВ

Согласно классификации Ти-Ян Лю из Microsoft Research [4], существующие алгоритмы обучения ранжированию можно разделить на три группы по их входным и выходным данным, основной функции модели и функциям потерь: поточечный, попарный и списочный подход.

Поточечный подход является частным случаем задачи регрессии, в ходе ее решения возможно использовать метод градиентного спуска, т.е. нахождения направления наискорейшего убывания функционала потерь. На практике поточечный подход дает не очень качественный результат.

В попарном подходе оптимизируемый функционал вычисляется на основе числа неверно упорядоченных («дефектных») пар документов и имеет следующий вид:

(4) $$Q(a) = \sum_{i \prec j} [M_{ij} < 0] \to \min. \cdot$$

Здесь $M_{ij} = a(x_i) - a(x_j)$ – отступ между «дефектными» парами, $a(x)$ – искомая функция ранжирования.

Этот функционал недифференцируемый, при решении задачи он аппроксимируется следующим гладким функционалом:

(5) $$Q(a) \le \sum_{i \prec j} L(M_{ij}) \to \min.$$

Здесь $L(M_{ij})$- невозрастающая непрерывная функция отступа. На практике наиболее успешно используется эвристическая оценка

(6) $$L(M_{ij}) = \log(1 + e^{-M_{ij}}) \cdot$$

2.3. МЕТОД ОПОРНЫХ ВЕКТОРОВ

Метод опорных векторов – один из известных алгоритмов обучения ранжированию, использующий попарный подход [3]. Он является разновидностью задачи квадратичного программирования. Основное преимущество этого метода – быстрое получение результирующей функции и эффективное дальнейшее

ее применение для новых классификаций. Метод опорных векторов имплементирован в библиотеке обучения ранжированию RankSVM (supported vector machines, машины опорных векторов).

В общем случае машины опорных векторов – семейство алгоритмов, основанных на обучении с учителем, использующих разделение объектов на группы с помощью классификатора. Далее рассмотрим линейный классификатор, т.е. прямую, в общем случае – гиперплоскость.

Дано пространство признаков ранжирования документов (f_1, …, f_n) по отношению к некоторому запросу q.

Таким образом, документ в этом пространстве представлен вектором ($f_1(d_i)$, …, $f_n(d_i)$). Для определения релевантности документа, т.е. отнесения его к одному из классов «релевантен, не релевантен», необходимо построить прямую (гиперплоскость), разделяющие объекты на эти два класса. Следовательно, все векторы (документы), расположенные с одной «стороны» гиперплоскости, относятся к первому классу, с другой — ко второму. Алгоритм выбирает оптимальные векторы, на которых строятся параллельные прямые (гиперплоскости), максимизирующие расстояние между объектами двух классов. Расстояние между ними называют зазором. Такие векторы называются опорными. Классификатор – прямая – таким образом, делит зазор между классами пополам.

Применительно к задаче ранжирования целевой функционал эмпирической ошибки метода опорных векторов представляется в виде

$$Q(a) = \frac{1}{2}\|w\|^2 + c\sum_{i \prec j} L(M_{ij}) \to \min_a \cdot \quad (7)$$

Здесь функция потерь L зависит от величины расстояния M между неверно классифицированными документами i и j и вычисляется как

$$L(M_{ij}) = (1 - M_{ij})_+, \quad (8)$$

$$M_{ij} = < w, f^{(j)} - f^{(i)} > . \quad (9)$$

Таким образом, можно сформулировать задачу квадратичного программирования следующего вида:

$$(10)\ \begin{cases} \dfrac{1}{2}\|w\|^2 + c\sum\limits_{i \prec j} \xi_{ij} \to \min\limits_{w,\xi} \\ < w, f^{(j)} - f^{(i)} > \geq 1 - \xi_{ij}, \quad i \prec j, \\ \xi_{ij} \geq 0, \quad i \prec j, \end{cases}$$

где ξ_{ij} – штраф за неверно упорядоченную пару документов. Решением является оптимальный вектор весов документов w^*. Он находится путем приведения задачи к двойственной.

2.4. МЕТОД СТОХАСТИЧЕСКОГО ГРАДИЕНТНОГО СПУСКА

При решении задачи ранжирования часто бывает трудно на каждом шаге алгоритма анализировать функционал потерь для всех существующих пар документов. Получение ранжирующей функции и дальнейшее обучение модели требуют слишком большого количества вычислительных ресурсов и по времени значительно превышают время стандартного отклика поисковой системы. Поэтому в задачах ранжирования с большими объемами обучающей и тестовой выборок применяют простой и эффективный алгоритм стохастического градиентного спуска [2].

В общем случае метод градиентного спуска – итерационный метод оптимизации гладкого целевого функционала путем вычисления направления скорейшего спуска – антиградиента. В отличие от стандартного, стохастический алгоритм не вычисляет сумму градиентов от каждого объекта, а работает с одним элементом на каждом шаге, что значительно упрощает вычисления без потери точности. Этот элемент выбирается случайным образом. Применительно к задаче ранжирования на каждом шаге алгоритма необходимо определить верное отношение порядка для документов *i, j*, т.е. этот метод реализует попарный подход к ранжированию. При этом оптимизируемая функция эмпирического риска вычисляется как:

$$(11)\ Q(a) \leq \sum_{i \prec j} L(M_{ij}) \to \min.$$

В качестве функционала потерь от неверно упорядоченной пары документов необходимо взять неубывающую гладкую функцию, например

(12) $L(M_{ij}) = \log(1 + e^{-\sigma M_{ij}})$,

где σ – коэффициент масштабирования, позволяющий пересчитать величину расстояния между неверно классифицированными документами в величину вероятности. Как и в методе опорных векторов, каждый документ задан вектором вещественных оценок признаков ранжирования $f = (f_1, \ldots, f_k)$. Пусть также Y – конечное множество этих вещественных оценок. Дана обучающая выборка $\{(f^{(1)}, y^{(1)}), \ldots, (f^{(1)}, y^{(1)})\}$, где $f^{(j)}$ – вектор вещественных оценок признака ранжирования для соответствующего документа относительно соответствующего запроса, $y^{(j)}$ – эталонная оценка релевантности данного документа данному запросу, отсюда выводится эталонный вектор весов признаков документа и, как следствие, вектор весов документов внутри выборки, соответствующей одному запросу.

Целью алгоритма является минимизация функции эмпирического риска, реализующей попарный подход:

(13) $Q(\bar{w}) = \sum_{i=1}^{n} L_i(\bar{w}) \to \min_{\bar{w}}$.

Здесь L_i – функция потерь от неверно упорядоченной i-й пары документов, т.е. в случае, когда менее релевантный из них оказался выше в поисковой выдаче. Начальное приближение вектор весов документов задано.

На каждой итерации алгоритма выбираются случайным образом запрос q, индексы j, k такие, что документ j менее релевантен данному запросу, чем документ k. Далее вычисляется градиент функции потерь и с его помощью переопределяется вектор весов документов, т.е. делается градиентный шаг только по данным выбранным документам:

(14) $w := w + \eta \frac{\partial L(M_{jk})}{\partial \sigma}(f^{(k)} - f^{(j)})$,

(15) $\frac{\partial L(M_{jk})}{\partial \sigma} = \frac{\sigma}{1 + \exp(\sigma < f^{(k)} - f^{(j)}, \bar{w} >)}$.

Здесь вектор-градиент называется градиентом ошибки и необходим для обновления весов внутри пары документов в правильном направлении на правильную величину.

Таким образом, обновление вектора весов на каждом можно выразить с помощью итеративной формулы

(16) $\overline{w}^{(t+1)} = \overline{w}^{(t)} - h\nabla L_i(\overline{w}^{(t)})$.

В методе стохастического градиента подсчет функции эмпирического риска на каждом шаге может быть очень долгим, вместо этого используются приближенные рекуррентные формулы среднего арифметического и экспоненциального скользящего среднего:

(17) $$\overline{Q_m} = \frac{1}{m}\cdot M_{jk_m} + \frac{1}{m}\cdot M_{jk_{m-1}} + \ldots = \\ = \frac{1}{m}\cdot M_{jk_m} + (1-\frac{1}{m})\overline{Q_{m-1}},$$

(18) $$\overline{Q_m} = \lambda\cdot M_{jk_m} + (1-\lambda)\cdot M_{jk_{m-1}} + \\ +(1-\lambda)^2\cdot M_{jk_{m-2}} + \ldots = \lambda\cdot M_{jk_m} + (1-\lambda)\overline{Q_{m-1}}.$$

где λ – коэффициент темпа забывания предыстории ряда. Критерий останова алгоритма – сходимость функции эмпирического риска или вектора весов.

Метод стохастического градиентного спуска имплементирован в алгоритме ранжирования LambdaRank [1]. Основной его особенностью является возможность оптимизации негладких функционалов (*MAP*, *NDCG*, *pFound*). Для этого производная заменяется изменением функционала при изменении порядка документов *j*, *k* в поисковой выдаче:

(19) $w := w + \eta\left|\Delta NDCG_{jk}\right|(f^{(k)} - f^{(j)})$.

3. Оценка качества ранжирующей модели

3.1. ТОЧНОСТЬ

Для оценки качества ранжирования результатов поиска применяются специальные метрики. Такие метрики должны определять, насколько полученные оценки релевантности соответствуют истинным значениям релевантности и порядку для данных документов и запроса. Наиболее распространенный подход к оценке эффективности ранжирующего отображения основан на проверке критериев точности и полноты поиска.

Точность (precision) определяет долю релевантных документов среди всех найденных, полнота (recall) – долю найденных системой среди всех релевантных.

Пусть имеем бинарную оценку $Y = \{0, 1\}$ релевантности документа d запросу q, выраженную функцией $y(q, d)$. Также имеем ранжирующее отображение $a(q, d)$. Точность (precision) чаще всего измеряется для первых n документов выдачи и определяется формулой

$$P_n(q) = \frac{1}{n}\sum_{i=1}^{n} y(q, d_q^{(i)}) \cdot \quad (20)$$

Поскольку выбор оптимального количества документов n, для которых производится оценка, неоднозначен, далее переходят к следующим двум метрикам. Средняя точность (average precision, *AP*) по позициям n релевантных. Здесь вклад в сумму вносят только релевантные документы ($y = 1$):

$$AP(q) = \sum_{i=1}^{n} y(q, d_q^{(i)}) \cdot P_n(q) \Big/ \sum_{i=1}^{n} y(q, d_q^{(i)}) \cdot \quad (21)$$

В общем случае среднюю точность можно определить как площадь под кривой – убывающей функцией зависимости точности от полноты:

$$AP = \int_0^1 precision(recall)dr \cdot \quad (22)$$

Средняя точность, усредненная по всем запросам (mean average precision, *MAP*), – наиболее часто используемая метрика.

$$MAP = \frac{1}{|Q|}\sum_{q \in Q} AP(q) \cdot \quad (23)$$

Эту метрику можно модифицировать, присвоив запросам веса в зависимости от их важности.

3.2. ДОЛЯ «ДЕФЕКТНЫХ» ПАР

«Дефектными» называют пары документов, между которыми в результате применения ранжирующей функции произошла инверсия порядка относительно эталонного, иными словами, менее релевантный документ оказался выше в выдаче. Чем больше таких транспозиций, тем менее эффективна ранжирующая

функция. Доля «дефектных» пар измеряется на n документах и определяется формулой

(24) $DP_n(q) = \frac{2}{n(n-1)} \sum_{i,j=1}^{n} [i < j][y(q, d_q^{(i)})] \cdot$

где $d_q^{(i)}$ – i-й по убыванию документ в выдаче, найденный по запросу q, $n(n-2)/2$ – количество всех возможных пар документов, а $i < j$ означает, что документ i хуже документа j. Преимущество данной метрики в том, что для нее могут использоваться вещественные оценки релевантности.

3.3. НОРМИРОВАННАЯ ПРИВЕДЕННАЯ СОВОКУПНАЯ ПОЛЕЗНОСТЬ

Данная метрика используется для оценки выигрыша релевантности на первых n документах.

(25) $DCG_n(q) = \sum_{i=1}^{n} G_q(d_q^{(i)}) \cdot D(i),$

(26) $G_q(d_q^{(i)}) = (2^{y(q,d)} - 1),$

(27) $D(i) = 1 / \log_2(i+1).$

Здесь множитель gain придает больший вес релевантным документам, а множитель discount придает больший вес документам в начале выдачи. Нормированная приведенная совокупная полезность является отношением полученной *DCG* к значению *DCG* при идеальном ранжировании:

(28) $NDCG_n(q) = \frac{DCG_n(q)}{\max DCG_n(q)}.$

4. Заключение

Подводя итог, необходимо заметить, что не существует универсального алгоритма ранжирования. Существующее на данный момент множество библиотек и методов оправдано разнообразием предметных областей информационного поиска и форм представления информации. Кроме того, критерии качества ранжирования зависят от типа основного алгоритма,

формата данных и области применения алгоритма. Универсального критерия также не существует.

Перспективы развития информационного поиска и дальнейшие улучшения моделей основываются на разработке и внедрении новых признаков ранжирования и анализе данных о поведении пользователя (кликовые признаки ранжирования) для создания персонализированных рекомендаций.

Литература

1. *From RankNet to LambdaRank to LambdaMART: An Overview*. – Microsoft, 2010.
2. BURGES C.J.C., SHAKED T., RENSHAW E. *Learning to rank using gradient descent* // Proc. of ICML. – 2005. – P. 89–96.
3. JOACHIMS T. *Support Vector Machine for Ranking*. – Cornell University, 2009.
4. TIE-YAN LIU *Learning to Search Web Pages with Query-Level Loss Functions*. – Microsoft Research, 2006.
5. MANNING C.D., RAGHAVAN P., SCHÜTZE H. *An introduction to information retrieval*. – Cambridge UP, 2009.
6. TURNBULL D., BERRYMAN J. *Relevant search with applications for Solr and Elasticsearch*. – Manning, 2018.

APPLICATION OF MACHINE LEARNING METHODS IN RANKING PROBLEMS IN INFORMATION RETRIEVAL SYSTEMS

Alyona Borisovskaya, Russian University of Transport, Moscow, bachelor's graduate (borisovsk0101@gmail.com).

Abstract: Ranking in information retrieval system is a type of special machine learning problem. According to the training sample it can be attributed to a classification problem, and according to the objective function – to regression problem. The ranking task is to find correct ranking mapping over training set with the given correct order ratio. In this case, the problem of ranking can be solved by algorithms based on support vector machine method and stochastic gradient method. The support vector machine method can be represented as the quadratic programming problem involving vectors of weights and features. The algorithm should minimize an objective function of incorrectly assigned weights (margin between classes). The

stochastic gradient method is based on randomly selecting a query and a pair of documents and performing a gradient step only on these documents. The resulting function can be evaluated using metrics such as the proportion of «defective» pairs, mean average precision and the normalized discounted cumulative gain. As for the ranking quality evaluation, there is no universal metric. Each metric is suitable for its own subject area.

Keywords: information retrieval, ranking, machine learning.

УДК 021.8 + 025.1
ББК 78.34