Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования «Московский государственный
технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА НА ТЕМУ:
«Применение машинного обучения в поисковых системах»

Студент: Волков Г.В.
Руководитель: Шаповалова М.С.

Цель и задачи работы

Цель работы - изучить алгоритмы машинного обучения, применяемые в поисковых системах

Для достижения поставленной цели следует решить следующие задачи:

- изучить основные понятия алгоритмов обучения ранжированию
- описать и классифицировать существующие алгоритмы
- произвести сравнительный анализ рассмотренных алгоритмов

Машинное обучение

Машинное обучение — раздел информатики, посвященный созданию алгоритмов, опирающихся на набор данных о каком-либо явлении. Алгоритмы формируют статистическую модель на основе специально подобранных обучающих данных, которую потом используются для решения практических задач.

Выделяется три основных способа обучения: с учителем, без учителя и с подкреплением.

Обучение ранжированию

Существует множество методов подбора формулы для ранжирования, но один из самых популярных – на основе машинного обучения, а именно обучение ранжированию с учителем. Целью этих методов является подбор ранжирующей модели, которая способна наилучшим образом приблизить и обобщить способ ранжирования на новые данные.

Для получения набора примеров используют асессоров, которые оцениваю степень релевантности документа запросу.

Классификация алгоритмов

Существующие алгоритмы обучения ранжированию делятся на три группы по подходу к обучению :

- поточечный
- попарный
- списочный

Алгоритмы обучения ранжированию

В данной работе рассмотрено несколько популярных алгоритмов:

- Linear Regression (поточечный)
- Ranking SVM (попарный)
- LambdaRank (попарный)
- ListNet (списочный)

На этапе ранжирования методы имеют схожий алгоритм. Для каждого документа вычисляется рейтинг релевантности, который зависит от вектора признаков документа и параметров метода ранжирования. Затем рейтинги сортируются по убыванию и получается ранжированный список документов.

Linear Regression

Метод обучения на основе регрессии для решения задачи оптимизации метрик DCG.

Для решения проблемы ранжирования можно использовать простой подход, основанный на регрессии.

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f, S_i, \{y_{i,j}\}_j)$$

$$L(f, S, \{y_j\}) =$$

$$= \sum_{j=1}^{m} w(x_j, S)(f(x_j, S) - y_j)^2 + u \max_j w'(x_j, S)(f(x_j, S) - \delta(x_j, S))_+^2$$

Ranking SVM

Ключевая идея алгоритма заключается в использовании метода SVM для попарного сравнения документов на то, какой из них более релевантный.

Теперь рассматривая разность векторов как новые объекты, получаем стандартную постановку SVM алгоритма.

$$\begin{cases} \frac{1}{2}\|\omega\|^2 + C\sum\limits_{i=1}^{N}\xi_i \to \min\limits_{\omega,\xi}, \\ y_i(\omega, x_i^1 - x_i^2) \leq 1 - \xi_i, \\ \xi_i \geq 0, \end{cases}$$

LambdaRank

В поточечных и попарных методах ранжирования итоговый функционал при обучении обычно не дифференцируемый. Алгоритм LambdaRank не определяет непрерывный приближенный функционал, вместо этого он определяет градиент функционала на всем списке документов:

$$\frac{\partial L}{\partial s_i} = -\lambda\big(s_1, y_1, \ldots, s_n, y_n\big)$$

$$\lambda_i = \frac{\partial L}{\partial s_i} = \frac{1}{G_{max}} \sum_j \left(\frac{1}{1 + \exp(s_j - si)}\right)(G(y_j) - G(y_i))(D(\pi_j) - D(\pi_i))$$

LambdaRank

λ – показывает насколько надо увеличить рейтинг i-го документа. Для этого надо изменить веса ω:

$$\frac{\partial L}{\partial \omega} = \sum_{i=1}^{n} \frac{\partial s_i}{\partial \omega} \sum_{j \in P_i} \frac{\partial L(s_i, s_j)}{\partial s_i} + \sum_{j=1}^{n} \frac{\partial s_j}{\partial \omega} \sum_{i \in P_j} \frac{\partial L(s_i, s_j)}{\partial s_j}$$

Таким образом, алгоритм LambdaRank заключается в итерационном пересчете весов:

$$\omega = \omega - \eta \frac{\partial L}{\partial \omega}$$

ListNet

Цель обучения формализована как минимизация общих потерь в отношении обучающих данных. В данном алгоритме используется вероятностные модели для вычисления функции потерь по списку.

$$P_{z^{(i)}(f_\omega)}(x_j^{(i)}) = \frac{\exp(f_\omega(x_j^{(i)}))}{\sum_{k=1}^{n^{(i)}} \exp(f_\omega(x_k^{(i)}))}$$

$$L(y^{(i)}, z^{(i)}(f_\omega)) = -\sum_{j=1}^{n^{(i)}} P_{y^{(i)}}(x_j^{(i)}) \log(P_{z^{(i)}(f_\omega)}(x_j^{(i)}))$$

ListNet

Градиент функции потери можно найти по следующей формуле:

$$\Delta\omega = \frac{\partial L(y^{(i)}, z^{(i)}(f_\omega))}{\partial\omega} = -\sum_{j=1}^{n^{(i)}} P_{y^{(i)}}(x_j^{(i)}) \frac{\partial f_\omega(x_j^{(i)})}{\partial\omega} + \\ + \frac{1}{\sum_{j=1}^{n^{(i)}} \exp(f_\omega(x_j^{(i)}))} \sum_{j=1}^{n^{(i)}} \exp(f_\omega(x_j^{(i)})) \frac{\partial f_\omega(x_j^{(i)})}{\partial\omega}.$$

Для минимизации целевой функции используется градиентный спуск

Критерии сравнения

В качестве критериев сравнения используются основные метрики оценки качества ранжирования: MAP, NDCG

MAP — метрика средней точности нахождения релевантных документов.

NDCG — мера качества

$$map@N = \frac{1}{K}\sum_{j=1}^{K} ap@N_j$$

$$DCG@N = \sum_{k=1}^{N} \frac{2^{r_t(P'(k))} - 1}{\log_2(k+1)}$$

Сравнение

Для оценки общей эффективности обучения методам ранжирования используется показатель «выигрышное число». Оно определяется как количество алгоритмов, которое алгоритм может превзойти на наборе датасетов

Метод	NDCG@3		NDCG@5		NDCG@10		MAP	
	NWN	к.д	NWN	к.д	NWN	к.д.	NWN	к.д.
Linear Regression	0,1053	9	0,2105	9	0	8	0	8
Ranking SVM	0,5000	11	0,4000	10	0,5217	16	0,5500	13
LambdaRank	0,8000	3	0,8000	3	0,6250	5	0,3333	3
ListNet	0,8000	10	0,8000	10	0,8500	10	0,8824	9

Заключение

Цель, которая была поставлена в начале научно-исследовательской работы, была достигнута: рассмотрены алгоритмы машинного обучения, применяемые в поисковых системах.

Решены все поставленные задачи:

- изучены основные понятия алгоритмов обучения ранжированию
- описаны и классифицированы существующие алгоритмы
- произведён сравнительный анализ рассмотренных алгоритмов