Жадные алгоритмы в задачах оптимизации качества ранжирования

Андрей Гулин, Павел Карпович

Москва
2009

Аннотация

Жадные алгоритмы(boosting модели) хорошо зарекомендовали себя при решении практических задач машинного обучения. В докладе будет рассказано об использовании данных техник при оптимизации качества ранжирования поисковой системы. Доклад состоит из двух частей. Первая часть является кратким описанием самой задачи улучшения качества ранжирования и используемых подходов к решению данной проблемы. Во второй части будет изложен один из классических "boosting"алгоритмов и примеры использования его модификаций на практике.

Содержание

- Задача ранжирования.
    - Меры качества(метрики).
    - Факторная модель ранжирования.
    - Задачи оптимизации (прямая максимизация метрик, аппроксимация оценки, оптимизация порядка на парах документов).
- Аппроксимация оценки релевантности. Жадные алгоритмы оптимизации.
- Модификация MatrixNet.
- Прямая максимизация метрик. Аппроксимация сложных дискретных метрик(DCG, nDCG).

Задача ранжирования

Главная цель: упорядочить документы по степени их соответствия поисковому запросу.

Как измерить качество поиска?
Данные:

- Набор поисковых запросов $Q = \{q_1, .., q_n\}$.
- Набор документов для каждого запроса $q \in Q$.

$$q \to \{d_1, d_2, ...\}$$

- Оценки релевантности для каждой пары $(query, document)$ (В нашей модели это будут действительные числа от 0 до 1 - $rel(q, d) \in [0, 1]$)

Меры качества (метрики)

Оценкой качества ранжирования является среднее значение **метрики качества**:

$$\frac{\sum_{q \in Q} EvMeas(ranking\ for\ query\ q)}{n}$$

Пример метрики качества $EvMeas$:

- **Precision-10** - *процент документов с положительными оценками релевантности в top-10*

Меры качества (метрики)

Оценкой качества ранжирования является среднее значение **метрики качества**:

$$\frac{\sum_{q \in Q} EvMeas(ranking\ for\ query\ q)}{n}$$

Пример метрики качества $EvMeas$:

- **Precision-10** - *процент документов с положительными оценками релевантности в top-10*

Я

Меры качества (метрики)

- **MAP** - mean average precision

$$MAP(ranking\, for\, query\, q) = \frac{1}{k}\sum_{i=1}^{k}\frac{i}{n_r(i)}$$

k - количество документов с положительными оценками релевантности для запроса q, $n_r(i)$ - позиция i-го документа с оценкой релевантности большей 0 в ранжировании.

Пример вычисления MAP

Запрос q и документы для него

$$q \rightarrow \{d_1, d_2, d_3, d_4, d_5\}$$

Ранжирование для запроса:

1. d_3 - 0.5
2. d_5 - 0
3. d_1 - 0
4. d_4 - 0.1
5. d_2 - 0

$$MAP = \frac{1}{k}\sum_{i=1}^{k}\frac{i}{n_r(i)} = \frac{1}{2}(\frac{1}{1} + \frac{2}{4})$$

Меры качества (метрики)

- **DCG - discounted cumulative gain**

$$DCG(ranking\ for\ query\ q) = \sum_{j=1}^{N_q} \frac{rel_j}{log_2 j + 1}$$

N_q - количество документов для запроса, rel_j - релевантность документа на позиции j.

- **нормализованный DCG(nDCG)**

$$nDCG(...) = \frac{DCG(ranking\ for\ query\ q)}{DCG(ideal\ ranking\ for\ query\ q)}$$

Пример вычисления метрики DCG

Запрос q и документы для него

$$q \rightarrow \{d_1, d_2, d_3, d_4, d_5\}$$

Ранжирование для запроса:

1. d_3 - 0.5
2. d_5 - 0
3. d_1 - 0
4. d_4 - 0.1
5. d_2 - 0

$$DCG = \sum_{j=1}^{N_q} \frac{rel_j}{log_2 j + 1} = \frac{1}{5}(\frac{0.5}{log_2 1 + 1} + \frac{0.1}{log_2 4 + 1})$$

Факторная модель ранжирования

- Каждая пара $(query, document)$ описывается вектором факторов.

$$(q, d) \rightarrow (f_1(q, d), f_2(q, d), ..)$$

- Поисковое ранжирование осуществляется сортировкой по значению **"функции релевантности"**. ***Функция релевантности*** - некоторая функция от вектора факторов:

$$fr(q, d) = 3.14 \cdot log_7(f_9(q, d)) + e^{f_{66}(q,d)} + ...$$

Факторная модель ранжирования

- Каждая пара $(query, document)$ описывается вектором факторов.

$$(q, d) \rightarrow (f_1(q, d), f_2(q, d), ..)$$

- Поисковое ранжирование осуществляется сортировкой по значению **"функции релевантности"**. ***Функция релевантности*** - некоторая функция от вектора факторов:

$$fr(q, d) = 3.14 \cdot log_7(f_9(q, d)) + e^{f_{66}(q,d)} + ...$$

Задачи оптимизации

Как получить хорошую функцию релевантности?

Обучающая выборка примеров P_l - множество пар (q, d) и их оценки релевантности $rel(q, d)$.

Использование методов машинного обучения для получения функции релевантности fr.

Задачи оптимизации (прямой подход)

- Прямая максимизация метрики:

$$\arg\max_{fr \in F} = \frac{\sum\limits_{q \in Q_l} EvMeas(ranking\ for\ query\ q\ with\ fr)}{n}$$

F - множество допустимых функций релевантности. Q_l - множество различных запросов в обучающем множестве P_l

Сложности: большинство метрик качества не являются непрерывными функциями.

Задачи оптимизации (аппроксимация оценки)

- Сведем оптимизационную задачу к задаче регрессии и минимизируем функцию ошибки - сумму значений *функции потери* на примерах из обучающей выборки:

$$\arg\min_{fr \in F} L_t(fr) = \frac{\sum\limits_{(q,d)\in P_l} L(fr(q,d), rel(q,d))}{n}$$

$L(fr(q,d), rel(q,d))$ - функция потери, F - множество допустимых функций релевантности. Примеры функций потерь:

- $L(fr, rel) = (fr - rel)^2$
- $L(fr, rel) = |fr - rel|$

Я

Задачи оптимизации (оптимизация порядка на парах)

- Использовать разработанные методы машинного обучения для задач классификации (SVM,...) при решении следующей проблемы:

  - упорядоченная пара документов (d_1, d_2)(документы для запроса q) принадлежит первому классу тогда и только тогда, когда $rel(q, d_1) > rel(q, d_2)$
  - упорядоченная пара документов (d_1, d_2)(документы для запроса q) принадлежит второму классу тогда и только тогда, когда $rel(q, d_1) \leq rel(q, d_2)$

Жадные алгоритмы оптимизации

Мы будем решать задачу регрессии:

$$\arg\min_{fr \in F} \frac{\sum_{(q,d) \in P_l} L(fr(q,d), rel(q,d))}{n}$$

Будем искать функцию релевантности в следующей форме:

$$fr(q,d) = \sum_{k=1}^{M} \alpha_k h_k(q,d)$$

Функция релевантности ищется в виде линейной комбинации функций $h_k(q,d)$, слагаемые $h_k(q,d)$ принадлежат простому семейству H (семейство слабых алгоритмов обучения) .

Жадные алгоритмы оптимизации

Мы будем решать задачу регрессии:

$$\arg\min_{fr \in F} \frac{\sum_{(q,d)\in P_l} L(fr(q,d), rel(q,d))}{n}$$

Будем искать функцию релевантности в следующей форме:

$$fr(q,d) = \sum_{k=1}^{M} \alpha_k h_k(q,d)$$

Функция релевантности ищется в виде линейной комбинации функций $h_k(q,d)$, слагаемые $h_k(q,d)$ принадлежат простому семейству H (семейство слабых алгоритмов обучения) .

Жадные алгоритмы оптимизации

Функцию релевантности будем строить итеративно. На каждой итерации мы будем добавлять слагаемое $\alpha_k h_k(q,d)$ к текущей функции релевантности:

$$fr_k(q,d) = fr_{k-1}(q,d) + \alpha_k h_k(q,d)$$

Значение параметра α_k и слабый алгоритм обучения $h_k(q,d)$ будут решением естественной задачи оптимизации:

$$\arg\min_{\alpha,h(q,d)} \frac{\sum_{(q,d)\in P_l} L(fr_{k-1}(q,d) + \alpha h(q,d), rel(q,d))}{n}$$

Данная задача может быть легко решена для квадратичной функции потерь и простых классов H.

Жадные алгоритмы оптимизации

Функцию релевантности будем строить итеративно. На каждой итерации мы будем добавлять слагаемое $\alpha_k h_k(q,d)$ к текущей функции релевантности:

$$fr_k(q,d) = fr_{k-1}(q,d) + \alpha_k h_k(q,d)$$

Значение параметра α_k и слабый алгоритм обучения $h_k(q,d)$ будут решением естественной задачи оптимизации:

$$\arg\min_{\alpha, h(q,d)} \frac{\sum_{(q,d)\in P_l} L(fr_{k-1}(q,d) + \alpha h(q,d), rel(q,d))}{n}$$

Данная задача может быть легко решена для квадратичной функции потерь и простых классов H.

Жадные алгоритмы оптимизации

Мы будем строить слагаемое $\alpha_k h_k(q,d)$ в три шага:

- **Аппроксимация градиента.** Рассмотрим функцию релевантности fr как вектор чисел, проиндексированный примерами из обучающей выборки. Вычислим вектор градиента $g = \{g_{(q,d)}\}_{(q,d)\in P_l}$ для функции ошибки:

$$g_{(q,d)} = \left[\frac{\partial L_t(fr)}{\partial fr(q,d)}\right]_{fr=fr_{k-1}}$$

- **Выбор слабого алгоритма обучения** (с точностью до константы). Найдем функцию $h_k(q,d)$, как решение следующей оптимизационной задачи:

$$\arg\min_{\beta,h(q,d)\in H}\sum_{(q,d)\in P_l}\left(g_{(q,d)} - \beta h(q,d)\right)^2$$

Жадные алгоритмы оптимизации

Мы будем строить слагаемое $\alpha_k h_k(q,d)$ в три шага:

- **Аппроксимация градиента.** Рассмотрим функцию релевантности fr как вектор чисел, проиндексированный примерами из обучающей выборки. Вычислим вектор градиента $g = \{g_{(q,d)}\}_{(q,d)\in P_l}$ для функции ошибки:

$$g_{(q,d)} = \left[\frac{\partial L_t(fr)}{\partial fr(q,d)}\right]_{fr=fr_{k-1}}$$

- **Выбор слабого алгоритма обучения** (с точностью до константы). Найдем функцию $h_k(q,d)$, как решение следующей оптимизационной задачи:

$$\arg\min_{\beta, h(q,d)\in H} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta h(q,d))^2$$

Жадные алгоритмы обучения

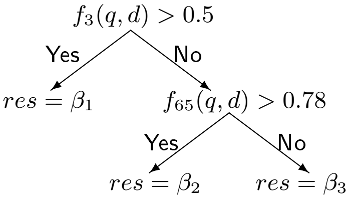
- **Выбор параметра α_k.** Найдем значение α_k, решая однопараметрическую задачу оптимизации:

$$\arg\min_{\alpha} \frac{\sum_{(q,d)\in P_l} L(fr_{k-1}(q,d) + \alpha h_k(q,d), rel(q,d))}{n}$$

Повторяем... Повторяем... Повторяем...

Выбор слабого алгоритма обучения

Пусть наш класс простых функций H будет семейством деревьев решений:



Пример дерева решений. Признаковое пространство разбивается на 3 области условиями в форме $f_j(q,d) > \alpha$ (f_j - признак разбиения, α - граница разбиения). Дерево решения является константной функцией на каждой из областей разбиения.

Выбор слабого алгоритма обучения

В нашей модели мы будем использовать деревья решений, разбивающие пространство на 6 областей. Попытаемся решить задачу оптимизации:

$$\arg \min_{h(q,d)\in H} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta h(q,d))^2$$

Предположим, что мы знаем структуру дерева решения $h(q,d)$ - знаем условия разбиения и области разбиения. Необходимо найти только значения функции в областях разбиения. Задача оптимизации сводится к обычной задаче регрессии:

$$\arg \min_{h(q,d)\in H,\beta} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta \beta_{ind(q,d)})^2$$

$ind(q,d)$ - номер области разбиения, которая содержит вектор факторов для пары (q,d) $(ind(q,d) \in \{1,..,6\})$.

Выбор слабого алгоритма обучения

В нашей модели мы будем использовать деревья решений, разбивающие пространство на 6 областей. Попытаемся решить задачу оптимизации:

$$\arg\min_{h(q,d)\in H} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta h(q,d))^2$$

Предположим, что мы знаем структуру дерева решения $h(q,d)$ - знаем условия разбиения и области разбиения. Необходимо найти только значения функции в областях разбиения. Задача оптимизации сводится к обычной задаче регрессии:

$$\arg\min_{h(q,d)\in H,\beta} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta\beta_{ind(q,d)})^2$$

$ind(q,d)$ - номер области разбиения, которая содержит вектор факторов для пары (q,d) $(ind(q,d) \in \{1,..,6\})$.

Выбор слабого алгоритма обучения

Жадный выбор дерева:

- $bestTree$ = константная функция (дерево с одной областью).
- **Жадное разбиение.** Пытаемся разбить одну из областей дерева $bestTree$ на две и найти лучшее разбиение.

Предположим, что у нас есть ограниченное количество возможных границ разбиения α_k. Тогда количество способов разбиения ограничено числом

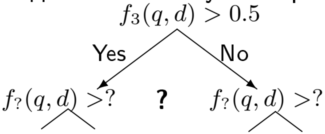
$$\#\{regions\} \cdot \#\{features\} \cdot \#\{split\ bounds\}$$

- Повторяем предыдущий шаг.

Выбор слабого алгоритма обучения

Жадный выбор дерева:

- $bestTree$ = константная функция (дерево с одной областью).
- **Жадное разбиение.** Пытаемся разбить одну из областей дерева $bestTree$ на две и найти лучшее разбиение.



Предположим, что у нас есть ограниченное количество возможных границ разбиения α_k. Тогда количество способов разбиения ограничено числом

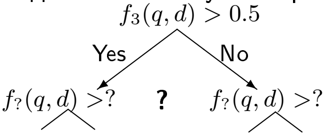
$$\#\{regions\} \cdot \#\{features\} \cdot \#\{split\,bounds\}$$

- Повторяем предыдущий шаг.

Выбор слабого алгоритма обучения

Жадный выбор дерева:

- $bestTree$ = константная функция (дерево с одной областью).
- **Жадное разбиение.** Пытаемся разбить одну из областей дерева $bestTree$ на две и найти лучшее разбиение.



Предположим, что у нас есть ограниченное количество возможных границ разбиения α_k. Тогда количество способов разбиения ограничено числом

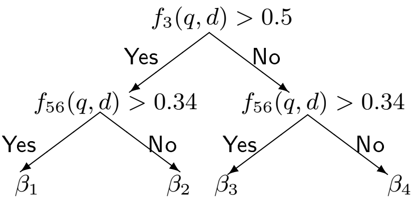
$$\#\{regions\} \cdot \#\{features\} \cdot \#\{split\ bounds\}$$

- Повторяем предыдущий шаг.

MatrixNet

Множество слабых алгоритмов- полные деревья решения с глубиной k и 2^k областями разбиения признакового пространства.

- Фиксированное количество слоев (фиксированная глубина дерева).
- Одни и те же условия разбиения на каждом слое.



Не нужна сложная структура у дерева: глубина дерева является главным параметром.

MatrixNet

Яндекс

Internet Mathematics 2009

company → internet-mathematics

Search

t 2009
Datasets
on
lution
ard
d Conditions

Leaderboard

The table shows both final contest results (May 15, 2009) and new results. Read more about the contest task and evalu
Datasets section.

MatrixNet

Team	Last upload time	Number of trials	Last result (public evaluation)	Final result
Joker	05.09.2009 (05:07 GMT+03)	2	4.283317	*4.151528*
Euclid	24.08.2009 (09:12 GMT+03)	30	4.280853	*4.149605*
alexeigor	07.05.2009 (17:02 GMT+03)	118	4.280676	*4.141230*
MysteriousGuest	24.08.2009 (12:33 GMT+03)	1	4.279174	*4.143886*
Победа	17.03.2009 (16:25 GMT+03)	3	4.276001	*4.139854*
ACGT	15.05.2009 (14:03 GMT+03)	21	4.274666	*4.128807*
WoodWeb	22.04.2009 (23:09 GMT+03)	12	4.267894	*4.127512*
Nordic	15.05.2009 (23:37 GMT+03)	4	4.266904	*3.857102*
stohastic	15.05.2009 (23:43 GMT+03)	176	4.266712	*4.118830*
Test	15.05.2009 (23:45 GMT+03)	58	4.264024	*3.859052*
ZENIT	15.05.2009 (23:20 GMT+03)	206	4.259964	*4.117877*
Euclid	08.05.2009 (21:46 GMT+03)	40	4.257802	*4.122558*

Аппроксимация сложных метрик (DCG)

Рассмотрим вероятностную модель ранжирования. Аппроксимацией метрики DCG для запроса q, множества документов $\{d_1, .., d_n\}$, и функции релевантности $fr(q, d)$ будет метрика $apxDCG$:

$$apxDCG = \sum_{r \in all\ permutations\ of\ docs} P(fr, r) DCG(r)$$

$P(fr, r)$ - вероятность получить ранжирование r в модели

Luce-Plackett. $DCG(r)$ - DCG метрика для перестановки r.

Модель Luce-Plackett

Есть набор документов $\{d_1, .., d_n\}$ и набор значений релевантности $\{fr(q, d_1), .., fr(q, d_n)\}$ для них.

Процесс выбора ранжирования в модели Luce-Plackett:

- Выбираем документ для первой позиции. Вероятность выбора документа d_i равна $\frac{fr(q,d_i)}{\sum\limits_{i=1}^{n} fr(q,d_i)}$. Допустим, что мы выбрали d_x.
- Второй документ выбирается из остальных. Вероятность выбора документа d_i равна $\frac{fr(q,d_i)}{\sum\limits_{i=1}^{n} fr(q,d_i) - fr(q,d_x)}$
- ...

Для каждого шага, если два документа d_i и d_j участвуют в нем, то отношение между вероятностями их выбора должно быть равно $\frac{fr(q,d_i)}{fr(q,d_j)}$

Модель Luce-Plackett

$\{\acute{d}_1, .., \acute{d}_n\}$ - перестановка документов $\{d_1, .., d_n\}$

$$P(fr, \{\acute{d}_1, .., \acute{d}_n\}) = \prod_{j=1}^{n} \frac{fr(q, \acute{d}_j)}{\sum\limits_{k=j}^{n} fr(q, \acute{d}_k)}$$

Спасибо за внимание.

- Tie-Yan Liu. Learning to Rank for Information Retrieval. Tutorial on WWW2008.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.
- Friedman, J. H. (1999). Stochastic gradient boosting (Tech. Rep.). Palo. Alto, CA: Stanford University, Statistics Department.
- Plackett, R. L. (1975). The analysis of permutations. Applied Statistics, 24, 193-202