Методы обучения ранжированию (Learning to Rank)

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

27 ноября 2013

Содержание

Определения и обозначения

X — множество объектов
$X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка
$i \prec j$ — правильный порядок на парах $(i, j) \in \{1, \ldots, \ell\}^2$

Задача:
построить ранжирующую функцию $a\colon X \to \mathbb{R}$ такую, что

$$i \prec j \quad \Rightarrow \quad a(x_i) < a(x_j)$$

Линейная модель ранжирования:

$$a(x; w) = \langle x, w \rangle$$

где $x \mapsto \big(f_1(x), \ldots, f_n(x)\big) \in \mathbb{R}^n$ — вектор признаков объекта x

Пример 1. Задача ранжирования поисковой выдачи

D — коллекция текстовых документов (documents)
Q — множество запросов (queries)
$D_q \subseteq D$ — множество документов, найденных по запросу q
$X = Q \times D$ — объектами являются пары «запрос, документ»:

$$x \equiv (q, d),\ q \in Q,\ d \in D_q$$

Y — упорядоченное множество рейтингов
$y : X \to Y$ — оценки релевантности, поставленные асессорами:
чем выше оценка $y(q, d)$, тем релевантнее документ d запросу q

Правильный порядок определён только между документами, найденными по одному и тому же запросу q:

$$(q, d) \prec (q, d') \quad \Leftrightarrow \quad y(q, d) < y(q, d')$$

Пример 1. Задача ранжирования поисковой выдачи

Типы признаков

- функции только документа d
- функции только запроса q
- функции запроса и документа (q, d)

- текстовые
  — слова запроса q встречаются в d чаще обычного
  — слова запроса q есть в заголовках или выделены в d
- ссылочные
  — на документ d много ссылаются
  — документ d содержит много полезных ссылок
- кликовые
  — на документ d часто кликают
  — на документ d часто кликают по запросу q

TF-IDF(q, d) — мера релевантности документа d запросу q

n_{dw} (term frequency) — число вхождений слова w в текст d;
N_w (document frequency) — число документов, содержащих w;
N — число документов в коллекции D;

N_w/N — оценка вероятности встретить слово w в документе;
$(N_w/N)^{n_{dw}}$ — оценка вероятности встретить его n_{dw} раз;
$P(q,d) = \prod\limits_{w \in q} (N_w/N)^{n_{dw}}$ — оценка вероятности встретить в документе d слова запроса $q = \{w_1, \ldots, w_k\}$ *чисто случайно*;

Оценка релевантности запроса q документу d:

$$-\log P(q,d) = \sum_{w \in q} \underbrace{n_{dw}}_{\mathrm{TF}(w,d)} \underbrace{\log(N/N_w)}_{\mathrm{IDF}(w)} \;\to\; \max.$$

$\mathrm{TF}(w,d) = n_{dw}$ — term frequency;
$\mathrm{IDF}(w) = \log(N/N_w)$ — inverted document frequency.

PageRank — классический ссылочный признак

Документ d тем важнее,
— чем больше других документов c ссылаются на d,
— чем важнее документы c, ссылающиеся на d,
— чем меньше других ссылок имеют эти документы c.

Вероятность попасть на страницу d, если кликать случайно:

$$\mathrm{PR}(d) = \frac{1-\delta}{N} + \delta \sum_{c \in D_d^{in}} \frac{\mathrm{PR}(c)}{|D_c^{out}|},$$

$D_d^{in} \subset D$ — множество документов, ссылающихся на d,
$D_c^{out} \subset D$ — множество документов, на которые ссылается c,
$\delta = 0.85$ — вероятность продолжать клики (damping factor),
N — число документов в коллекции D.

Sergey Brin, Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. 1998.

Пример 2. Коллаборативная фильтрация

U — пользователи, users
I — предметы, items (фильмы, книги, и т.п.)
$X = U \times I$ — объектами являются пары «user, item»

Правильный порядок определён между предметами, которые выбирал или рейтинговал один и тот же пользователь:

$$(u, i) \prec (u, i') \quad \Leftrightarrow \quad y(u, i) < y(u, i')$$

Рекомендация пользователю u — это список предметов i, упорядоченный с помощью функции ранжирования $a(u, i)$

В роли признаков объекта $x = (u, i)$ могут выступать
$y(u', i)$ — рейтинги, поставленные другими пользователями u'

То есть, поиск коллаборации $\Leftrightarrow$ отбор признаков

Точность и средняя точность

Пусть $Y = \{0, 1\}$, $y(q, d)$ — релевантность,
$a(q, d)$ — искомая функция ранжирования,
$d_q^{(i)}$ — i-й документ по убыванию $a(q, d)$.

Precision, точность — доля релевантных среди первых n:

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

Average Precision, средняя P_n по позициям релевантных документов:

$$AP(q) = \sum_n y(q, d_q^{(n)}) P_n(q) \Big/ \sum_n y(q, d_q^{(n)})$$

Mean Average Precision, средняя AP по всем запросам:

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

Доля «дефектных пар»

Пусть $Y \subseteq \mathbb{R}$, $y(q,d)$ — релевантность,
$a(q,d)$ — искомая функция ранжирования,
$d_q^{(i)}$ — i-й документ по убыванию $a(q,d)$.

Доля инверсий порядка среди первых n документов:

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

Связь с коэффициентом ранговой корреляции (τ Кенделла):

$$\tau(a, y) = 1 - 2 \cdot DP_n(q).$$

Связь с AUC (area under ROC-curve) в задачах классификации с двумя классами $Y = \{-1, +1\}$, $a\colon X \to Y$

$$AUC_n(q) = \frac{1}{\ell_- \ell_+} \sum_{i,j=1}^{n} \big[y_i > y_j\big] \big[a(x_i) < a(x_j)\big] = \frac{n(n-1)}{2\ell_- \ell_+} \cdot DP_n(q).$$

DCG — Discounted Cumulative Gain

Пусть $Y \subseteq \mathbb{R}$, $y(q,d)$ — релевантность,
$a(q,d)$ — искомая функция ранжирования,
$d_q^{(i)}$ — i-й документ по убыванию $a(q,d)$.

Дисконтированная (взвешенная) сумма выигрышей:

$$DCG_n(q) = \sum_{i=1}^{n} \underbrace{G_q(d_q^{(i)})}_{\text{gain}} \cdot \underbrace{D(i)}_{\text{discount}}$$

$G_q(d) = (2^{y(q,d)} - 1)$ — бóльший вес релевантным документам
$D(i) = 1/\log_2(i+1)$ — бóльший вес в начале выдачи

Нормированная дисконтированная сумма выигрышей:

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

$\max DCG_n(q)$ — это $DCG_n(q)$ при идеальном ранжировании

Яндекс pFound — модель поведения пользователя

Пусть $Y \subseteq [0,1]$,
$y(q,d)$ — релевантность, оценка вероятности найти ответ в d,
$a(q,d)$ — искомая функция ранжирования,
$d_q^{(i)}$ — i-й документ по убыванию $a(q,d)$.

Вероятность найти ответ в первых n документах:

$$pFound_n(q) = \sum_{i=1}^{n} P_i \cdot y\left(q, d_q^{(i)}\right),$$

где P_i — вероятность дойти до i-го документа:

$$P_1 = 1;$$

$$P_{i+1} = P_i \cdot \left(1 - y\left(q, d_q^{(i)}\right)\right) \cdot \left(1 - P_{out}\right),$$

где P_{out} — вероятность прекратить поиск без ответа

Яндекс pFound — модель поведения пользователя

Параметры критерия pFound:

- $P_{out} = 0.15$ — вероятность прекратить поиск без ответа;
- $y(q, d)$ — оценка вероятности найти ответ в документе:

оценка асессора	$y(q, d)$
Vital	0.61
Useful	0.41
Relevant+	0.14
Relevant−	0.07
Not Relevant	0.00

Гулин А., Карпович П., Расковалов Д., Сегалович И.
Оптимизация алгоритмов ранжирования методами машинного обучения // РОМИП-2009.

Основные подходы к ранжированию

- Point-wise — поточечный
- Pair-wise — попарный
- List-wise — списочный

Переход к гладкому функционалу качества ранжирования:

$$Q(a) = \sum_{i \prec j} \big[\underbrace{a(x_j) - a(x_i)}_{\text{Margin}(i,j)} < 0\big] \leqslant \sum_{i \prec j} \mathscr{L}\big(a(x_j) - a(x_i)\big) \quad \to \quad \min$$

где $a(x)$ — алгоритм ранжирования;
$\mathscr{L}(M)$ — убывающая непрерывная функция отступа $\text{Margin}(i,j)$:

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

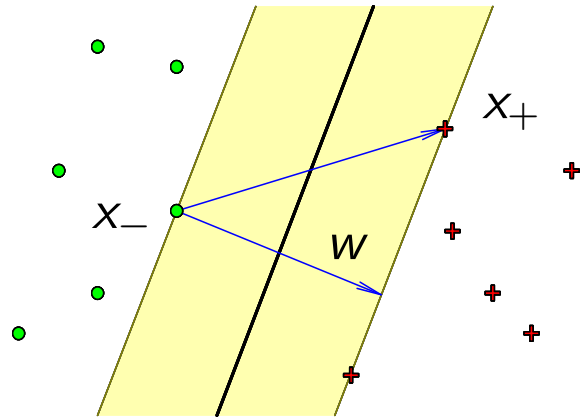
Напоминание: SVM — метод опорных векторов

Линейный классификатор:

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

Задача обучения SVM:

$$\begin{cases} \dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell}\xi_i \to \min\limits_{w,w_0,\xi}; \\ M_i(w,w_0) \geqslant 1 - \xi_i, \quad i = 1,\ldots,\ell; \\ \xi_i \geqslant 0, \quad i = 1,\ldots,\ell. \end{cases}$$



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

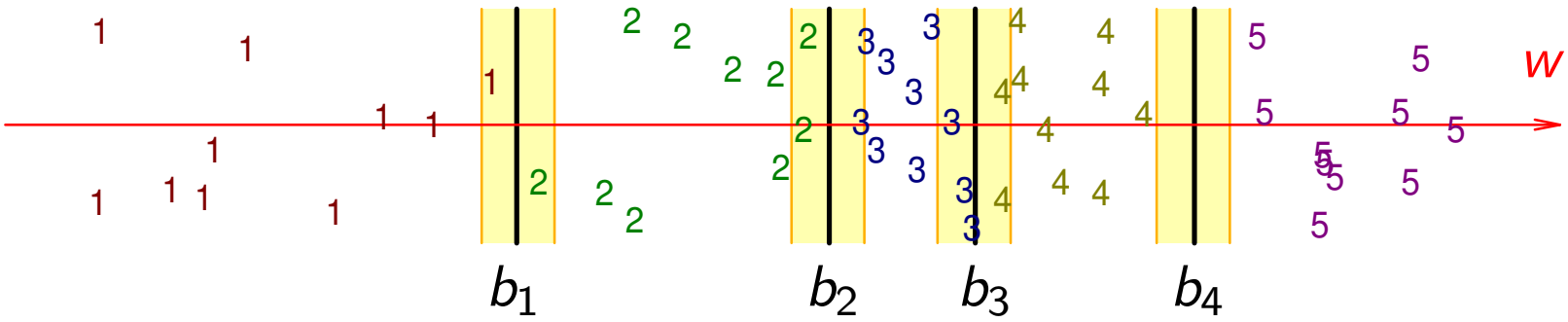
Эквивалентная задача безусловной минимизации:

$$Q(w,w_0) = \sum_{i=1}^{\ell}\big(1 - M_i(w,w_0)\big)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w,w_0}.$$

Ранговая классификация OC-SVM (Ordinal Classification SVM)

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

$$a(x) = y, \quad \text{если} \quad b_{y-1} < \langle w, x \rangle \leqslant b_y$$



Постановка задачи SVM для ранговой классификации:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell} [y_i \neq K]\,(\xi_i + \xi_i^*) \to \min\limits_{w,b,\xi}; \\ b_{y_i-1} + 1 - \xi_i^* \leqslant \langle w, x_i \rangle \leqslant b_{y_i} - 1 + \xi_i; \\ \xi_i^* \geqslant 0, \quad \xi_i \geqslant 0. \end{cases}$$

Ranking SVM

Постановка задачи SVM для попарного подхода:

$$Q(a) = \tfrac{1}{2}\|w\|^2 + C\sum_{i \prec j} \mathscr{L}\Big(\underbrace{a(x_j) - a(x_i)}_{\text{Margin}(i,j)}\Big) \quad \to \quad \min_a,$$

где $a(x) = \langle w, x\rangle$ — функция ранжирования,
$\mathscr{L}(M) = (1 - M)_+$ — функция потерь,
$M = \text{Margin}(i,j) = \langle w, x_j - x_i\rangle$ — отступ,

Постановка задачи квадратичного программирования:

$$\begin{cases} \tfrac{1}{2}\|w\|^2 + C\sum_{i \prec j} \xi_{ij} \to \min\limits_{w,\xi}; \\ \langle w, x_j - x_i\rangle \geqslant 1 - \xi_{ij}, \quad i \prec j; \\ \xi_{ij} \geqslant 0, \quad i \prec j. \end{cases}$$

От RankNet до LambdaRank

RankNet: гладкий функционал качества ранжирования:

$$Q(a) = \sum_{i \prec j} \mathscr{L}\big(a(x_j) - a(x_i)\big) \quad \to \quad \min$$

при $\mathscr{L}(M) = \log(1 + e^{-\sigma M})$ и линейной модели $a(x) = \langle w, x \rangle$.

Метод стохастического градиента:
выбираем на каждой итерации q, $i \prec j$ случайно,

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

Christopher J.C. Burges From RankNet to LambdaRank to LambdaMART: An Overview // Microsoft Research Technical Report MSR-TR-2010-82. 2010.

От RankNet до LambdaRank

Метод стохастического градиента:

$$w := w + \eta \cdot \underbrace{\frac{\sigma}{1 + \exp(\sigma\langle x_j - x_i, w\rangle)}}_{\lambda_{ij}} \cdot (x_j - x_i);$$

Оказывается, для оптимизации негладких функционалов MAP, NDCG, pFound достаточно домножить λ_{ij} на изменение данного функционала при перестановке местами $x_i \leftrightarrows x_j$.

LambdaRank: домножение на изменение NDCG при $x_i \leftrightarrows x_j$ приводит к оптимизации NDCG:

$$w := w + \eta \cdot \frac{\sigma}{1 + \exp(\sigma\langle x_j - x_i, w\rangle)} \cdot |\Delta NDCG_{ij}| \cdot (x_j - x_i);$$

Christopher J.C. Burges From RankNet to LambdaRank to LambdaMART: An Overview // Microsoft Research Technical Report MSR-TR-2010-82. 2010.

Ранжирование в Яндексе

- Ежемесячно в выборку добавляется более 50 000 оценок асессоров
- За 4 года придумано и проверено около 800 признаков
- Технология MatrixNet — градиентный бустинг над ODT (небрежными решающими деревьями)
- PairWise подход лучше, чем PointWise и ListWise
- Технология FML (Friendly Machine Learning) — среда для тестирования алгоритмов машинного обучения

Резюме в конце лекции

- Ранжирование — особый класс задач машинного обучения.
- Критерий качества ранжирования зависит от приложения. Наилучшего универсального критерия не существует.
- Три подхода: поточечный, попарный, списочный. Теоретически списочный должен быть наилучшим. Однако в Яндексе лучше всего работает попарный.

- *Tie-Yan Liu.* Learning to Rank for Information Retrieval. Springer-Verlag Berlin Heidelberg. 2011
- *Hang Li.* A Short Introduction to Learning to Rank // IEICE Trans. Inf. & Syst., Vol.E94–D, No.10 October 2011.