Avito Context Ad Clicks

Остапец Андрей

23 сентября 2015 г.

Общие сведения

- Проходил меньше 2 месяцев: 2 июня - 28 июля.
- Участникам было предоставлено 8 файлов для анализа, суммарным объемом более 38 Gb.
- 414 участников
- Ни один участник после валидации результатов не был дисквалифицирован ;)

Объявления

Объявления бывают разных типов...

Context Ad

Стол обеденный с тонированным стеклом (ЛДСП

4 680 руб.

Обеденный стол подойдёт к любому интерьеру

bellucci-mebel.ru

О рекламодателе

Highlighted Ad

Бильярдный стол

13 000 руб.

Компания

Иркутск

Сегодня 01:19

Regular Ads

Кухонный гарнитур арт. 3571

65 030 руб.

"Дан мебель"- фабричные кухни

Одинцово

Сегодня 01:15

Кухонный гарнитур арт. 1982

64 400 руб.

"Дан мебель"- фабричные кухни

Долгопрудный

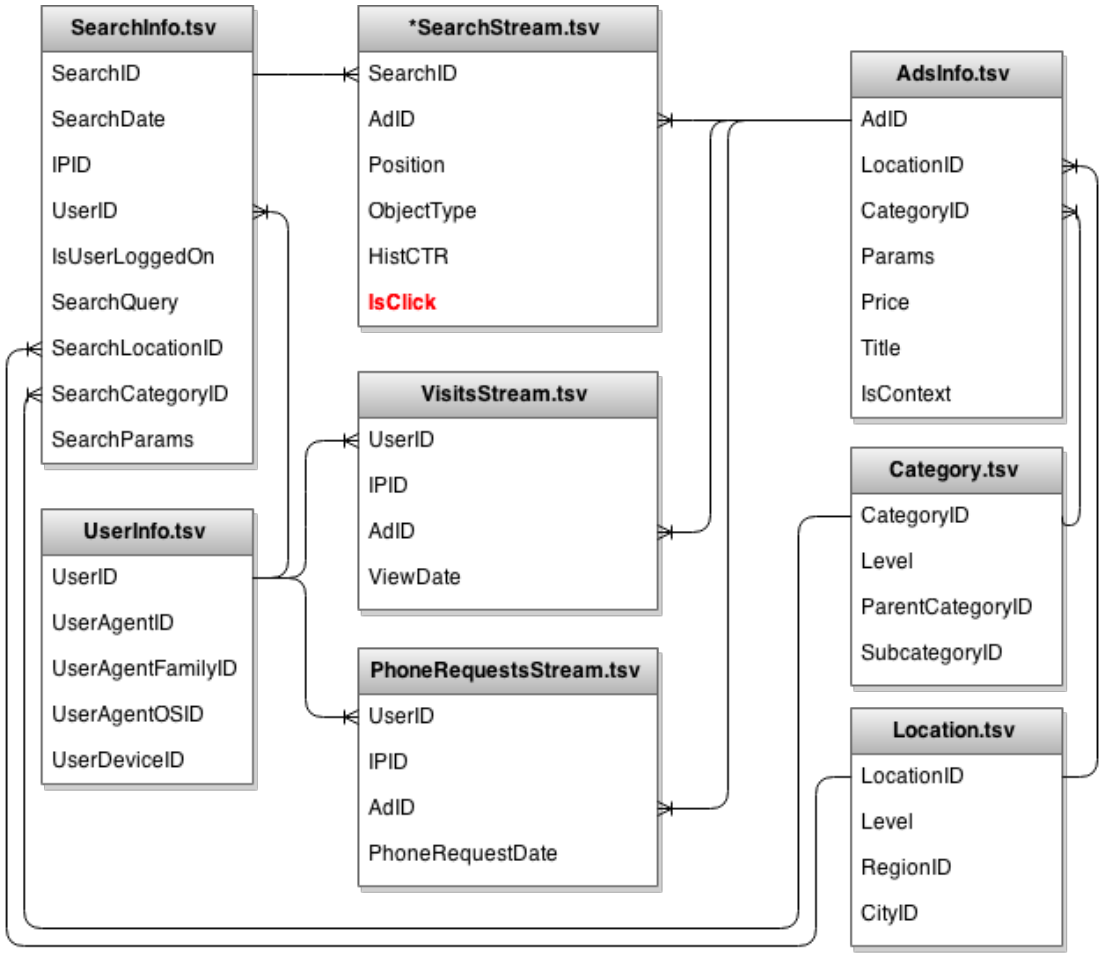
Сегодня 01:14

Постановка задачи

Задача: по информации о пользователе, объявлении, контексте поисковой выдачи предсказать вероятность клика на данное контекстное объявление

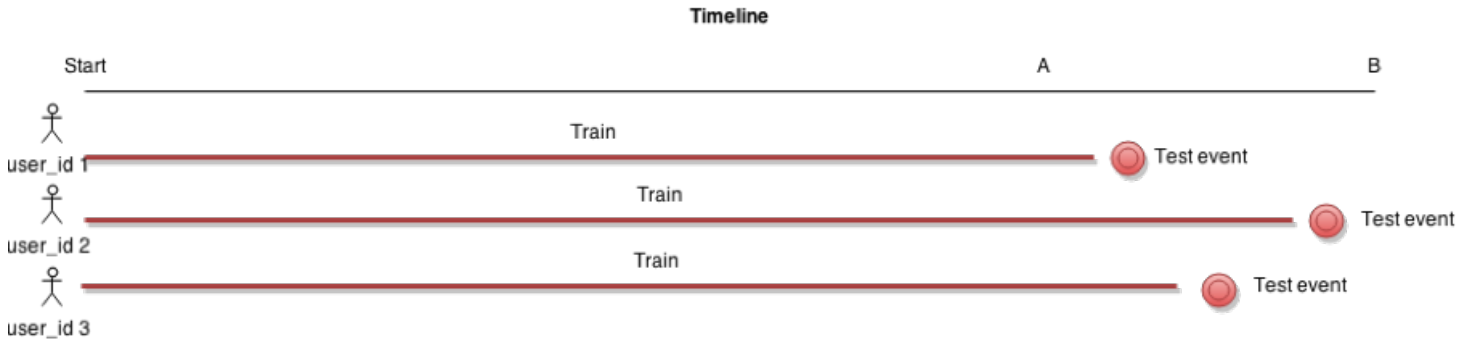
Связи между разными файлами

8 различных таблиц



Разделение на обучение и тест

Данные с 25 апреля (Start) по 20 мая (B). В тестовую выборку попадает последняя сессия пользователя совершенная не раньше 12 мая (A).



Данные для обучения

- В trainSearchStream данные о 392 миллионах объявлений в поисковой выдаче. Из них 190 миллионов контекстных.
- 28 тысяч различных контекстных объявлений.
- Почти 5 миллионов уникальных пользователей.
- Контекстные объявления были только на 1 или 7 позициях.
- Логировалась информация только об объявлениях на 1, 2, 6, 7 и 8 позициях.

Тестовая выборка

- Для 8 миллионов контекстных объявлений нужно было предсказать вероятность клика.
- 4.3 млн пользователей для которых нужно сделать предсказание
- В качестве функционала качества использовался

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Первые подходы

Первые подходы

Тривиальные решения

Что было известно с самого начала?

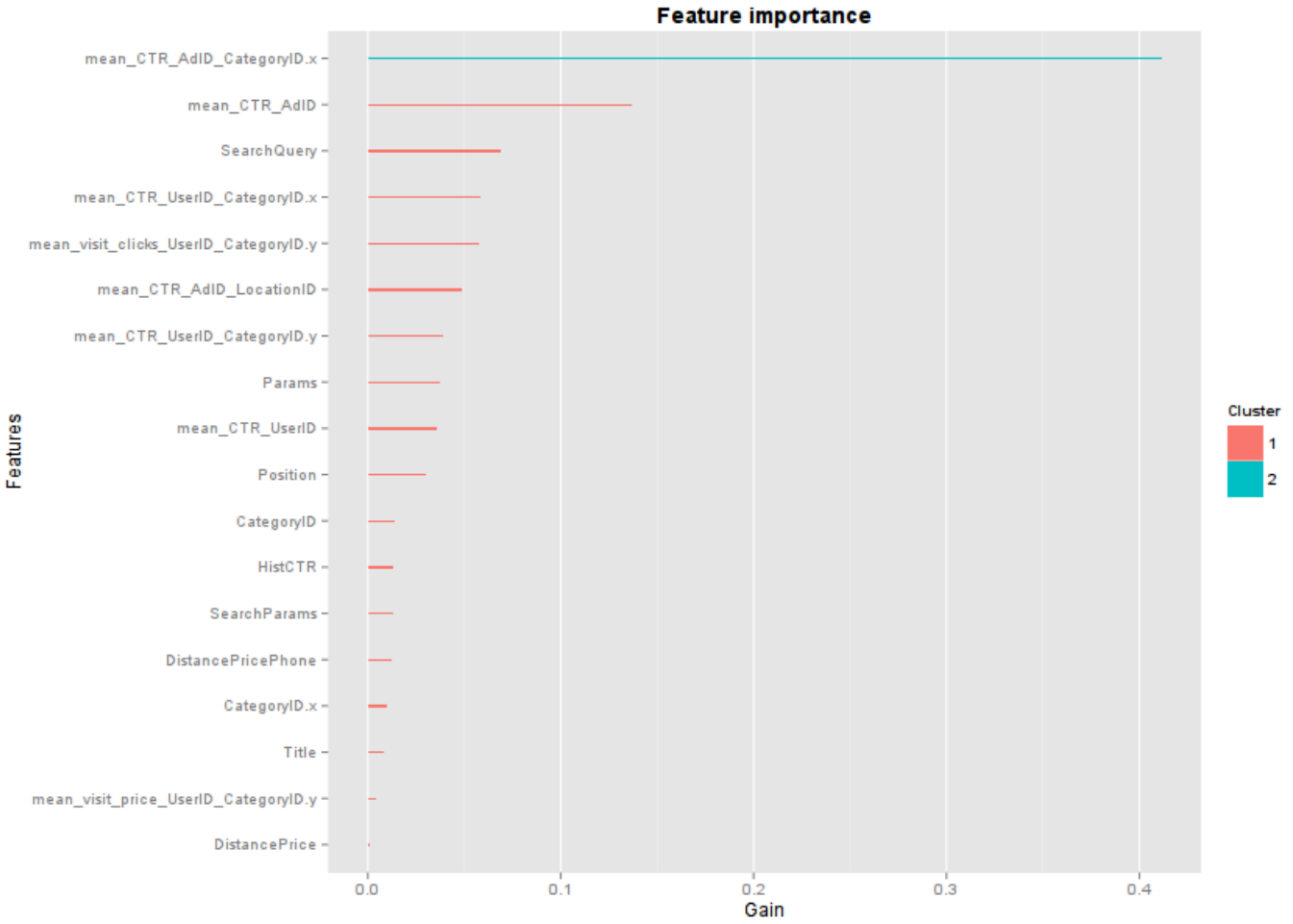
- HistCTR Benchmark: 0.05717
- HistCTR + Position + RandomForest: 0.04836

Boosting

Разбиение тренировочной выборки на несколько частей:

- Первая часть (до 12 мая) - для подсчета статистик
- Вторая часть (80% сессий с 12 мая по 20 мая) - для обучения
- Третья часть (20% сессий с 12 мая по 20 мая) - для валидации

Xgboost


Feature importance
mean_CTR_AdID_CategoryID.x
mean_CTR_AdID
SearchQuery
mean_CTR_UserID_CategoryID.x
mean_visit_clicks_UserID_CategoryID.y
mean_CTR_AdID_LocationID
mean_CTR_UserID_CategoryID.y
Params
mean_CTR_UserID
Position
CategoryID
HistCTR
SearchParams
DistancePricePhone
CategoryID.x
Title
mean_visit_price_UserID_CategoryID.y
DistancePrice
Features
0.0
0.1
0.2
0.3
0.4
Gain
Cluster
1
2

Результаты

Лучший результат, который удалось получить с использованием бустинга: **0.04584** (линейная комбинация 2 моделей обученных на 2 разных подвыборках).

FTRL algorithm

FTRL (Follow the Regularized Leader) algorithm

Статья

Ad Click Prediction: a View from the Trenches

H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young,
Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov,
Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg,
Arnar Mar Hrafnkelsson, Tom Boulos, Jeremy Kubica
Google, Inc.
mcmahan@google.com, gholt@google.com, dsculley@google.com

ABSTRACT

Predicting ad click–through rates (CTR) is a massive-scale learning problem that is central to the multi-billion dollar online advertising industry. We present a selection of case studies and topics drawn from recent experiments in the setting of a deployed CTR prediction system. These include improvements in the context of traditional supervised learning based on an FTRL-Proximal online learning algorithm (which has excellent sparsity and convergence properties) and the use of per-coordinate learning rates.

We also explore some of the challenges that arise in a real-world system that may appear at first to be outside the domain of traditional machine learning research. These include useful tricks for memory savings, methods for assessing and visualizing performance, practical methods for providing confidence estimates for predicted probabilities, calibration methods, and methods for automated management of features. Finally, we also detail several directions that did not turn out to be beneficial for us, despite promising results elsewhere in the literature. The goal of this paper is to highlight the close relationship between theoretical advances and practical engineering in this industrial setting, and to show the depth of challenges that appear when applying traditional machine learning methods in a complex dynamic system.

Categories and Subject Descriptors

I.5.4 [**Computing Methodologies**]: Pattern Recognition—*Applications*

learning. Sponsored search advertising, contextual advertising, display advertising, and real-time bidding auctions have all relied heavily on the ability of learned models to predict ad click–through rates accurately, quickly, and reliably [28, 15, 33, 1, 16]. This problem setting has also pushed the field to address issues of scale that even a decade ago would have been almost inconceivable. A typical industrial model may provide predictions on billions of events per day, using a correspondingly large feature space, and then learn from the resulting mass of data.

In this paper, we present a series of case studies drawn from recent experiments in the setting of the deployed system used at Google to predict ad click–through rates for sponsored search advertising. Because this problem setting is now well studied, we choose to focus on a series of topics that have received less attention but are equally important in a working system. Thus, we explore issues of memory savings, performance analysis, confidence in predictions, calibration, and feature management with the same rigor that is traditionally given to the problem of designing an effective learning algorithm. The goal of this paper is to give the reader a sense of the depth of challenges that arise in real industrial settings, as well as to share tricks and insights that may be applied to other large-scale problem areas.

2. BRIEF SYSTEM OVERVIEW

When a user does a search **q**, an initial set of candidate ads is matched to the query **q** based on advertiser-chosen keywords. An auction mechanism then determines whether these ads are shown to the user, what order they are shown in, and what prices the advertisers pay if their ad is clicked.

Алгоритм

Algorithm 1 Per-Coordinate FTRL-Proximal with L_1 and L_2 Regularization for Logistic Regression

#With per-coordinate learning rates of Eq. (2).
Input: parameters α, β, λ_1, λ_2
$(\forall i \in \{1, \dots, d\})$, initialize $z_i = 0$ and $n_i = 0$
for $t = 1$ **to** T **do**
  Receive feature vector $\mathbf{x}_t$ and let $I = \{i \mid x_i \neq 0\}$
  For $i \in I$ compute

$$w_{t,i} = \begin{cases} 0 & \text{if } |z_i| \leq \lambda_1 \\ -\left(\frac{\beta + \sqrt{n_i}}{\alpha} + \lambda_2\right)^{-1} (z_i - \operatorname{sgn}(z_i)\lambda_1) & \text{otherwise.} \end{cases}$$

  Predict $p_t = \sigma(\mathbf{x}_t \cdot \mathbf{w})$ using the $w_{t,i}$ computed above
  Observe label $y_t \in \{0, 1\}$
  for all $i \in I$ **do**
    $g_i = (p_t - y_t)x_i$ *#gradient of loss w.r.t. w_i*
    $\sigma_i = \frac{1}{\alpha}\left(\sqrt{n_i + g_i^2} - \sqrt{n_i}\right)$ *#equals $\frac{1}{\eta_{t,i}} - \frac{1}{\eta_{t-1,i}}$*
    $z_i \leftarrow z_i + g_i - \sigma_i w_{t,i}$
    $n_i \leftarrow n_i + g_i^2$
  end for
end for

Градиент

Используем логистическую регрессию, обучаем последовательно. На шаге t:

- $\mathbf{x_t} \in \mathbb{R}^d$ - текущий объект
- $\mathbf{w_t} \in \mathbb{R}^d$ - текущий вектор весов
- Предсказание $p_t = \sigma(\mathbf{w_t} \cdot \mathbf{x_t})$, где $\sigma(a) = \frac{1}{1+exp(-a)}$
- LogLoss: $l_t(\mathbf{w_t}) = -y_t \log p_t - (1 - y_t) \log(1 - p_t)$
- $\bigtriangledown l_t(\mathbf{w}) = (\sigma(\mathbf{w} \cdot \mathbf{x}_t) - y_t)\mathbf{x}_t = (p_t - y_t)\mathbf{x}_t$

Метод стохастического градиента

Для последовательности градиентов $\mathbf{g}_t \in \mathbb{R}^d$ пересчет весов осуществляется по формуле:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{g}_t \ (1),$$

где η_t - невозрастающая последовательность (например, $\eta_t = \frac{1}{\sqrt{t}}$)

FTRL-Proximal

Для последовательности градиентов $\mathbf{g}_t \in \mathbb{R}^d$ пересчет весов осуществляется по формуле:

$$\mathbf{w}_{t+1} = \underset{w}{\text{argmin}}\left(\sum_{s=1}^{t} g_s \cdot \mathbf{w} + \frac{1}{2}\sum_{s=1}^{t} \sigma_s ||\mathbf{w} - \mathbf{w_s}||_2^2 + \lambda_1 ||w||_1\right) \textbf{(2)},$$

где σ_s - темп обучения $\left(\sum_{s=1}^{t} \sigma_s = \frac{1}{\eta_t}\right)$

Эквивалентность двух формул

При $\lambda_1 = 0$ метод стохастического градиента и FTRL-Proximal выдают совершенно одинаковые последовательности весов $\mathbf{w}_t$!

Эквивалентность двух формул

Выражение под argmin в формуле (2) можно переписать в виде:

$$(\sum_{s=1}^{t} g_s - \sum_{s=1}^{t} \sigma_s \mathbf{w}_s) \cdot \mathbf{w} + \frac{1}{\eta_t}||w||_2^2 + \lambda_1||w||_1 + (const)$$

Если мы храним $\mathbf{z}_{t-1} = \sum_{s=1}^{t-1} g_s - \sum_{s=1}^{t-1} \sigma_s \mathbf{w}_s$, тогда

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \mathbf{g}_t + (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})\mathbf{w}_t$$

Пересчет весов

Теперь можем получить аналитическое решение в покоординатной форме:

$$w_{t+1,i} = \begin{cases} 0, & \text{если } |z_{t,i}| \leq \lambda_1; \\ -\eta_t(z_{t,i} - sgn(z_{t,i})\lambda_1), & \text{иначе.} \end{cases}$$

Темп обучения

- В методе стохастического градиента темп обучения один для всех координат
- В [M. J. Streeter and H. B. McMahan Less regret via online conditioning] показано, что для некоторых семейств задач такое поведение асимптотически неоптимально.
- Для каждой координаты свой темп обучения $\eta_{t,i} = \frac{\alpha}{\beta + \sqrt{\sum_{s=1}^{t} g_{s,i}^2}}$, где $g_{s,i}$ - i-ая координата вектора g_s

Hashing trick

- Исходные признаки: {'AdID': '13524889', 'Position': '1', 'HistCtr': '0.03', 'UserID': '1371072', 'SearchQuery': '', ...}
- Пытаемся добавить новый признак: AdID_UserID.
- Уникальных объявлений 28 тыс., уникальных пользователей 5 млн.
- Новый признак имеет более чем $1.4 \cdot 10^{11}$ значений!

Hashing trick

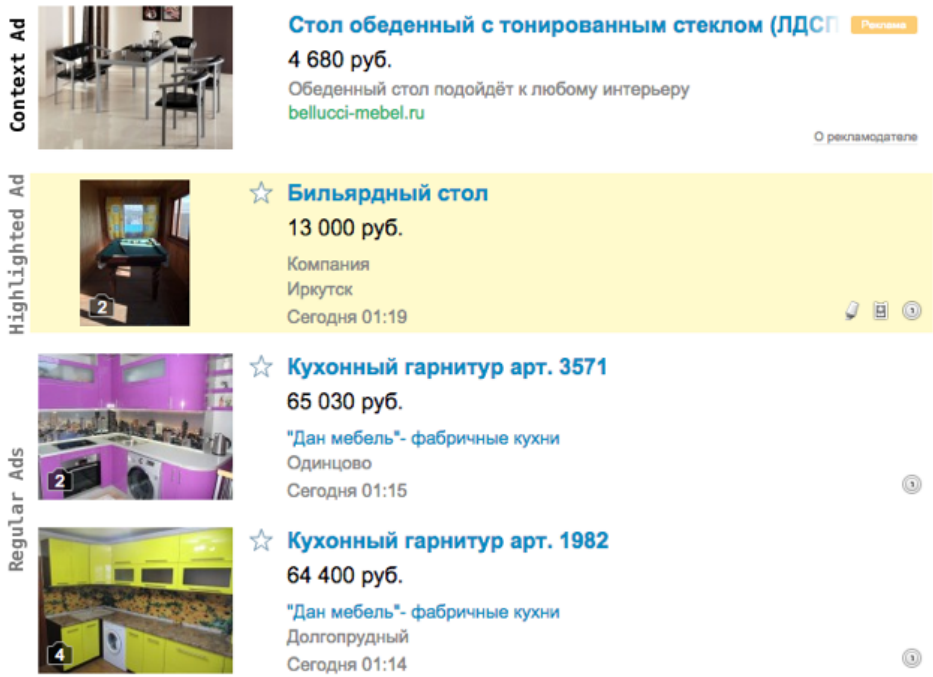
- Объединяем 2 исходных признака: 'AdID_UserID': '13524889_1371072',
- Берем хэш от этой строки: hash('AdID_UserID_13524889_1371072') % bits, где bits - размерность признакового пространства (bits $= 2^K$)
- В массив X на место этого индекса записываем 1
- Повторяем для всех признаков
- Получаем sparse vector X $= [0, \ldots, 0, 1, 0, \ldots, 0, \ldots, 1, 0, \ldots, 0]$
- Размерность вектора признаков обычно $\geq 2^{20}$, но для каждого объекта ненулевые обычно десятки признаков.

Результаты

Только за счет смены модели удалось улучшить результат до **0.04424**

Golden feature

PositionFactor = [Position: 1place:ObjectType; 2place:ObjectType; 6place:ObjectType; 7place:ObjectType; 8place:ObjectType]



Результаты

Добавление одного этого признака повысило результат до **0.04212**

Финальные результаты

Completed • $20,000 • 414 teams

Avito Context Ad Clicks

Tue 2 Jun 2015 – Tue 28 Jul 2015 (55 days ago)

Dashboard

Private Leaderboard - Avito Context Ad Clicks

This competition has completed. This leaderboard reflects the final standings.

See someone using multiple accounts?
Let us know.

#	Δrank	Team Name ‡ model uploaded * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	–	Owen *	0.04028	20	Tue, 28 Jul 2015 23:47:23
2	↑1	Gzs_iceberg ‡ *	0.04042	18	Tue, 28 Jul 2015 19:01:35
3	↓1	Dmitry & Leustagos ‡ *	0.04046	131	Tue, 28 Jul 2015 15:37:37
4	–	J.A. Guerrero	0.04074	9	Tue, 28 Jul 2015 13:13:07 (-17.1h)
5	–	Gilberto&Stanislav	0.04107	108	Tue, 28 Jul 2015 23:50:43 (-0.4h)
6	↑1	ash hafez	0.04109	28	Tue, 28 Jul 2015 23:31:02
7	↓1	typedef	0.04110	7	Tue, 28 Jul 2015 12:12:07 (-20.2d)
8	–	Lars Ropeid Selsås	0.04123	112	Tue, 28 Jul 2015 11:28:36
9	–	**Andrey&Alexander**	**0.04155**	**112**	**Tue, 28 Jul 2015 21:30:19 (-0.6h)**
10	–	Lulu Freyman	0.04167	13	Tue, 28 Jul 2015 08:37:28 (-9.7d)
11	–	D&D	0.04173	82	Tue, 28 Jul 2015 23:42:14 (-1.3h)
12	–	kipa	0.04202	4	Mon, 27 Jul 2015 21:15:11
13	–	Kaffo	0.04238	37	Tue, 28 Jul 2015 23:55:50 (-2.7d)
14	↑1	Sergey Yurgenson	0.04283	21	Tue, 28 Jul 2015 18:51:55