# ИССЛЕДОВАНИЕ ТОЧНОСТИ МЕТОДА ГРАДИЕНТНОГО БУСТИНГА СО СЛУЧАЙНЫМИ ПОВОРОТАМИ

УДК 519.6

**Виктор Владимирович Китов,**
к.ф.-м. н., математик 1-й категории Московского государственного университета им. Ломоносова, доцент научно-исследовательского университета «Высшая школа экономики», доцент Российского экономического университета им. Г.В.Плеханова.
Эл. почта: v.v.kitov@yandex.ru

В статье рассматривается метод градиентного бустинга с осуществлением случайных поворотов признакового пространства на каждом шаге обучения алгоритма. Исследуется качество данного метода на различных модельных задачах бинарной классификации. Полученные результаты анализируются и даются рекомендации по применению указанного метода.

***Ключевые слова:*** *прогнозирование, классификация, градиентный бустинг, случайные повороты.*

**Victor V. Kitov,**
PhD in Mathematics, mathematician of Moscow State University, docent at National Research University "Higher School of Economics", docent at Plekhanov Russian University of Economics.
E-mail: v.v.kitov@yandex.ru

**ACCURACY ANALYSIS OF THE GRADIENT BOOSTING METHOD WITH RANDOM ROTATIONS**

Gradient boosting method with random rotations is considered, where before training each base learner random rotation is applied to the feature space. The accuracy metric of the given method is estimated for a broad range of generated problems of binary classification. Obtained results are evaluated and recommendations given for application of this method.

***Keywords:*** *forecasting, classification, gradient boosting, random rotations.*

## 1. Введение

С развитием вычислительных мощностей и запоминающих устройств в последние десятилетия значительно повысились возможности по сбору, обработке, анализу и прогнозированию данных в самых различных предметных областях, таких как торговля, реклама, сотовая связь, предоставление интернет-услуг, и многих других. Наука, занимающаяся алгоритмами анализа и прогнозирования данных в полуавтоматическом режиме, когда большинство параметров моделей подбираются по располагаемым данным, а не вручную, называется машинным обучением.

Одной из важнейших задач анализа данных в машинном обучении является задача прогнозирования. Существует множество прогнозирующих алгоритмов, накладывающих свои предположения о данных, таких как метод ближайших соседей, метод опорных векторов, линейная регрессия, логистическая регрессия, решающие деревья, нейросети и др. – см [1]. Однако, поскольку прогнозируемые данные, скорее всего, имеют более сложные свойства, чем те предположения, которые делаются в рассматриваемых методах, то более выигрышной, с точки зрения точности, стратегией является прогнозирование не единственной моделью, а набором моделей, объединенных в композицию (другое название – ансамбль моделей), см. [2, 3]. В этом случае к данным применяется сразу несколько прогнозирующих моделей, называемых базовыми моделями, а потом результат определяется в виде агрегирования полученных прогнозов – в простейшем случае, в виде линейной комбинации. Одним из наиболее популярных ансамблевых методов прогнозирования является градиентный бустинг. По данным [4], реализация xgBoost данного метода использовалась в большинстве прогнозирующих алгоритмов, победивших в соревнованиях по машинному обучению на сайте kaggle.com в 2015 году.

Ключом к успешному применению ансамблевых методов прогнозирования является разнообразие (diversity) базовых моделей, на базе которых строится финальный прогноз. Очевидно, что если усреднять по идентичным моделям, то выигрыша по сравнению с применением одной базовой модели не будет. И наоборот, чем разнообразнее базовые модели, тем потенциально больше у них возможностей исправлять ошибки друг друга и уточнять финальный прогноз. В работе [5] предложена идея генерации случайных поворотов, которые потом применяются к признакам прогнозируемых объектов перед обучением базовых моделей, в качестве которых выступают решающие деревья. Высказана гипотеза, что за счет различных поворотов базовые алгоритмы становятся более разнообразными, что в результате повышает точность полученной композиции моделей. Данная гипотеза подтверждена для случаев, когда случайные повороты применяются к ансамблевым алгоритмам случайного леса (random forest), и особо случайных деревьев (extra-random trees). Реализация случайных поворотов в алгоритме бустинга менее тривиальна алгоритмически, поскольку требует интеграции поворотов внутри алгоритма, поэтому в указанной статье не рассматривалась (рассматривался упрощенный алгоритм). Тем не менее, интересен вопрос, насколько оправдан данный подход для алгоритма бустинга, который изучается в последующих разделах данной работы.

В разделе 2 дается описание алгоритма градиентного бустинга с поворотами. В разделе 3 дается описание эксперимента по проверке точности метода на различных модельных данных, и обсуждаются результаты. В разделе 4 дается заключение и варианты дальнейших исследований.

## 2. Градиентный бустинг со случайными поворотами

Для расширения класса функций, моделируемых ансамблями деревьев, в работе [5] предложен подход, согласно которому перед каждой настройкой базового алгоритма в ансамбле делается случайный поворот признакового пространства.

В алгоритме бустинга сначала настраивается базовая модель $F_1(x)$, затем настраивается $F_2(x)$ так, чтобы максимально исправить ошибки первой модели, затем $F_3(x)$ так, чтобы максимально исправить ошибки первых двух моделей и т.д. Результирующий прогноз основывается на суммарном прогнозе $F_1(x) + F_2(x) + \ldots + F_M(x)$, см. рис. 1. В задаче бинарной классификации, рассматриваемой в данной статье, прогнозируемый класс $\hat{y} \in \{+1,-1\}$ будет выражаться формулой:

$$\hat{y} = sign(F_1(x) + F_2(x) + ... + F_M(x)),$$

где $sign(u)$ – функция, возвращающая знак аргумента $u$.

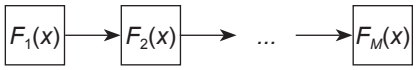


**Рис. 1. Обычная схема бустинга**

В методе бустинга со случайными поворотами перед настройкой каждой модели производится случайный поворот признакового пространства. Перед настройкой модели $F_1(x)$ производится случайный поворот $x \to R_1(x)$, перед настройкой модели $F_2(x)$ производится случайный поворот $x \to R_2(x)$ и т.д., см. схему метода на рис. 2. Поворот на шаге $i$ осуществляется некоторой случайно сгенерированной матрицей поворота $R_i$, обзор методов генерирования таких матриц см. в [6]. После настройки модели последовательность случайных поворотов $R_1, R_2, \ldots R_m$ запоминается, и на этапах применения модели (прогнозирования) используется та же самая последовательность поворотов.

В качестве базовых моделей в градиентном бустинге обычно используются решающие деревья. Решающие деревья представимы в виде деревьев, где каждому листу сопоставлен прогноз, а каждому внутреннему узлу – проверка условия. В наиболее распространенных реализациях деревьев в каждом узле $t$ проверяется условие, что некоторый признак $x^{i(t)}$ больше некоторого порога $h(t)$. Такое решающее дерево приводит к кусочно-постоянному решению, причем области постоянства – прямоугольники с осями, параллельными осям координат. Градиентный бустинг над деревьями приводит к взвешенной сумме таких деревьев, поэтому решение в бустинге также представимо в кусочно-постоянном виде, где области постоянства – прямоугольники с осями, параллельными осям координат. В большинстве случаев это предположение о данных не выполняется. Такое предположение не соответствует действительности в задаче регрессии, когда градиент прогнозируемой функции не параллелен одной из осей координат. В задаче классификации данное предположение не выполнено, когда истинная граница между классами не параллельна осям координат, что почти всегда будет выполняться на практике. Градиентный бустинг будет пытаться аппроксимировать наклонную границу между классами с помощью ступенчатой функции со многим количеством ступенек, что не всегда возможно точно, особенно, когда данных мало, а размерность исходной задачи велика – см. рис. 3. На рис. 3 объекты обучающей выборки представлены в виде точек. Цвет точки определяется ее классом и стоит задача по точкам обучающей экстраполировать прогнозы классов на все точки пространства объектов. Видно, что граница между классами кусочно-линейна, и линии параллельны осям координат.

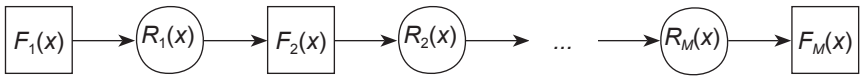


**Рис. 2. Схема бустинга со случайными поворотами**

Градиентный бустинг с поворотами является более гибким методом. Каждое дерево композиции будет по-прежнему выдавать кусочно-постоянный прогноз, где области постоянства будут прямоугольники в пространстве признаков. Но оси прямоугольников будут параллельны не осям исходного признакового пространства, а осям повернутого признакового пространства на величину случайного поворота. Это расширяет круг моделируемых функций, в частности, появляется возможность разделять признаковое пространство наклонными линиями за меньшее число разбиений, чем в случае обычного градиентного бустинга над деревьями. На рис. 4 показано применение градиентного бустинга с поворотами к той же обучающей выборке объектов, что и на рис. 3.

На рис. 4 видно, что разделяющая граница между классами уже не является кусочно-линейной с линиями, параллельными осям координат – здесь уже допустимы наклонные разделения. С одной стороны, это позволяет более гибко и более экономично (меньшим числом разбиений) описывать классы объектов, что может повысить точность прогнозирования. А с другой стороны, за счет большей гибкости, это может внести большую степень переобученности модели на обучающую выборку, что в итоге понизит качество прогнозирования новых данных – см. [3]. Какой из данных факторов окажется

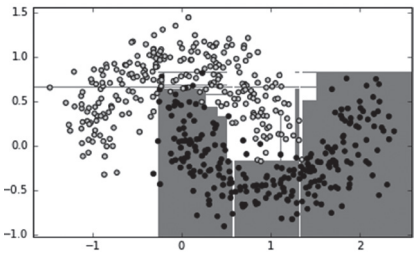


**Рис. 3. Разделение на два класса методом обычного бустинга**

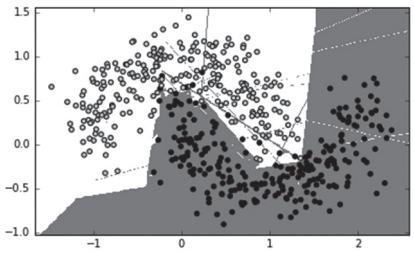


**Рис. 4. Разделение на два класса методом бустинга с поворотами**

более значимым, будет ясно из последующих экспериментов.

### 3. Эксперимент на модельных данных

Изучим вопрос, как соотносится точность обычного градиентного бустинга и градиентного бустинга с поворотами на искусственно сгенерированных данных, про которые заранее известна зависимость между признаками и классами. Будем рассматривать случай независимых признаков, распределенных равномерно на интервале [–1,1] каждый. Класс будет определяться условием, попадает или не попадает объект $x = (x_1, \ldots x_D)$ в определенную область. Определим функцию

$$I[u] = \begin{cases} 1, \text{ если условие } u \text{ выполнено.} \\ -1, \text{ если условие } u \text{ не выполнено.} \end{cases}$$

Будут рассматриваться следующие типы разделения объектов на классы:

1) Классы разделены линейно:

$$y = I[x_1 + \ldots + x_D > 0],$$

показано на рис 5*а*.

2) Объекты одного класса лежат внутри многоугольника:

$$y = I\left[|x_1| + \ldots + |x_D| < \frac{D}{2}\right],$$

показано на рис 5*б*.

3) Объекты одного класса лежат внутри параболы:

$$y = I\left[x_1 > \frac{1}{D}\left(x_2^2 + \ldots x_D^2\right)\right],$$

показано на рис 5*б*.

4) Объекты одного класса лежат внутри шара:

$$y = I\left[x_1^2 + \ldots + x_D^2 < \frac{D}{3}\right],$$

показано на рис 5*б*.

Для каждого типа данных (линейный, многогранник, парабола, шар) сгенерируем $N$ объектов случайно, где $N$ = 200, 500, 1000, 2000, 3000. Точки будут генерироваться в $D$-мерном пространстве, где $D$ = 2, 3, 5, 7, 15. К данным применим инверсии – класс каждой точки будем оставлять прежним с вероятностью 1-$p$ и менять на противоположный с вероятностью $p$. Это позволит оценить устойчивость метода к зашумленным данным, которые не в точности соответствуют заложенной

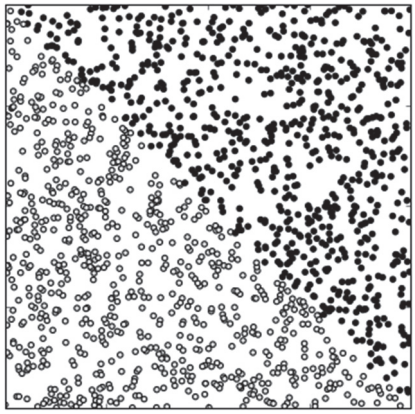

**Рис. 5*а***

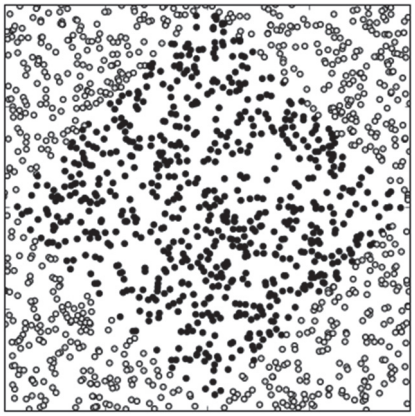

**Рис. 5*б***

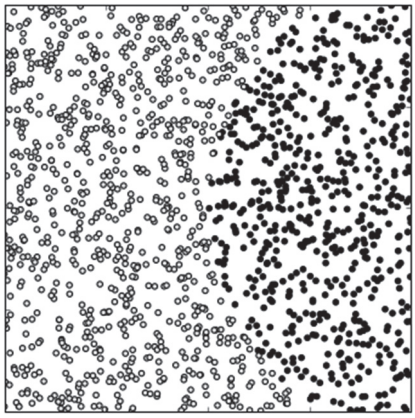

**Рис. 5*в***

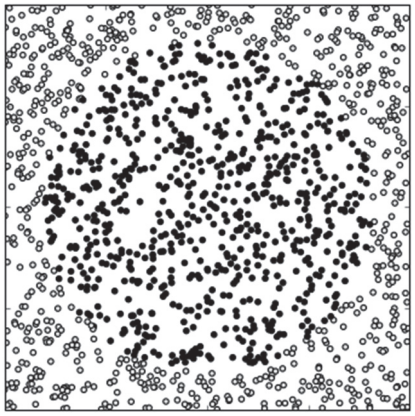

**Рис. 5*г***

в них закономерности. Операции инверсии применим с параметром $p$ = 0, 0.05, 0.1, 0.2, 0.3. С учетом всевозможных типов данных и различных параметров $N$, $D$, $p$ всего получим 500 наборов данных.

На каждом наборе оценивалась точность классификации обычным методом градиентного бустинга и методом градиентного бустинга с поворотами. Каждый набор данных случайно делится на обучающую выборку (50% объектов), валидационную выборку (25% объектов) и контрольную выборку (25% объектов). Параметр shrinkage всегда полагался равным 0.1, сэмплирование объектов на каждом шаге бустинга не использовалось (см. эти параметры в [1]), параметр $M \le 500$ оптимального числа базовых моделей выбирался таким образом, чтобы обеспечить наивысшую точность метода на валидационном множестве. С подобранным параметром $M$ для каждого метода оценивалась его точность на контрольной выборке. Поскольку бустинг с поворотами – это алгоритм со встроенной рандомизацией, то чтобы уменьшить эффект этой рандомизации, метод бустинга с поворотами перезапускался 30 раз, и вычислялись среднее и стандартное отклонение точности и числа базовых моделей.

Далее вычислялся балл по формуле:

$$score = \begin{cases} +1, \bar{a}_{RR} - a > \gamma * std[a_{RR}] \\ -1, \bar{a}_{RR} - a < -\gamma * std[a_{RR}], \\ 0, \text{иначе} \end{cases}$$

где $\bar{a}_{RR}$ – средняя точность, $std[a_{RR}]$ – стандартное отклонение точности, вычисленные по 30 перезапускам бустинга с поворотами, $a$ – значение точности обычного бустинга, $\gamma$ – параметр, показывающий, насколько статистически значимым должно быть изменение в точности. Такая оценка мотивирована неравенством Чебышева, которое для рассматриваемого случая принимает вид:

$$P(|a_{RR} - a| \ge \gamma * std[a_{RR}]) \le \frac{1}{\gamma^2}$$

Брались значения параметра $\gamma = 0, 1, 2, 3$. С ростом $\gamma$, очевидно, число положительных и отрицательных баллом уменьшается (что соответствует значимому отличию в точности), а число случаев, когда балл равен нулю (соответствует незначимому отличию в точности), возрастает. Это показано в табл. 1 ниже.

*Таблица 1*

**Зависимость числа баллов от параметра $\gamma$**

| $\gamma$ | #(score = +1) | #(score = –1) | #(score = +1) / #(score = –1) | #(score = 0) |
|---|---|---|---|---|
| 0 | 310 | 190 | 1,631578947 | 0 |
| 1 | 225 | 104 | 2,163461538 | 171 |
| 2 | 141 | 55 | 2,563636364 | 304 |
| 3 | 73 | 26 | 2,807692308 | 401 |

В табл. 1 #(условие) обозначает число раз, сколько было выполнено условие, указанное в скобках, для различный наборов данных.

В дальнейшем будем рассматривать сумму баллов для различных комбинаций параметров $N$, $D$, $p$ и типа данных. Если эта сумма положительна, то на большинстве наборов данных бустинг с поворотами работает точнее, чем бустинг без поворотов, а если отрицательна, то на большинстве наборов данных бустинг с поворотами работает менее точно. Чем выше эта сумма, тем более предпочтительно использовать бустинг с поворотами по сравнению с обычным бустингом. Оказалось, что дальнейшие качественные выводы, полученные для разных $\gamma$, сохраняются, поэтому ниже будут приведены результаты только для $\gamma = 2$.

*Таблица 2*

**Зависимость числа суммарных баллов от типа данных и вероятности инверсии класса**

| $p$ | 0.0 | 0.05 | 0.1 | 0.2 | 0.3 | сумма: |
|---|---|---|---|---|---|---|
| тип данных: | | | | | | |
| шар | 0 | 6 | 7 | 4 | 3 | 20 |
| линейный | 20 | 17 | 16 | 9 | 8 | 70 |
| многогранник | 3 | 2 | 3 | 9 | 1 | 18 |
| парабола | 0 | –6 | –6 | –4 | –6 | –22 |
| сумма: | 23 | 19 | 20 | 18 | 6 | 86 |

*Таблица 3*

**Зависимость числа суммарных баллов от размерности данных $D$ и размера выборки $N$**

| $D$ | 2.0 | 3.0 | 5.0 | 7.0 | 15.0 | сумма: |
|---|---|---|---|---|---|---|
| $N$ | | | | | | |
| 200.0 | 4 | 5 | 0 | –2 | –1 | 6 |
| 500.0 | 10 | 6 | 4 | 1 | –2 | 19 |
| 1000.0 | 6 | 7 | 5 | 0 | 0 | 18 |
| 2000.0 | 13 | 8 | 3 | –1 | –1 | 22 |
| 3000.0 | 12 | 11 | 2 | 0 | –4 | 21 |
| сумма: | 45 | 37 | 14 | –2 | –8 | 86 |

Из табл. 1 видно, что из 500 наборов данных, метод с поворотами работал лучше, чем стандартный бустинг в 141 случаях, а хуже – в 55 случаях, что свидетельствует об общей предпочтительности бустинга со случайными поворотами.

В табл. 2 показано, как изменялся суммарный балл для разных типов данных и различных значениях параметра $p$, характеризующего уровень «шума» в данных.

Как видно из табл. 2, метод лучше всего работает на линейном типе данных, хорошо работает на шаре и многограннике, а в случае параболы работает хуже, чем обычный бустинг. Это можно объяснить тем, что метод позволяет более экономично проводить наклонные линейные границы между классами, что наиболее важно, когда граница между классами действительно линейна. В случае многогранника граница кусочно линейна, а в случае шара большую роль играет его симметричность относительно начала координат. Случайные повороты позволяют более точно выделить подобные симметричные фигуры. Парабола генерировалась так, что ее ось совпадает с осью $x_1$. Поэтому ее логично выделять разбиениями вдоль именно этой, а не повернутой оси, что объясняет более высокую, в большинстве случаев, точность обычного бустинга.

Также из табл. 2 видно, что с ростом параметра $p$ преимущество бустинга с поворотами падает. Это объясняется тем, что бустинг с поворотами – более гибкий, а потому более склонный к переобучению метод. При увеличении шума в данных этот метод в большей степени начинает переобучаться на шумовые наблюдения обучающей выборки, поэтому его преимущество по сравнению с методом обычного бустинга сокращается.

По табл. 3 видно, что наибольшее преимущество метода достигается при малой размерности пространства. С ростом размерности преимущество уменьшается. Это можно объяснить тем, что в пространстве большой размерности существует слишком большое множество вариантов поворота, и сложно перебрать все варианты, и найти хорошо подходящий под данные. Также из табл. 2 видно, что с ростом числа наблюдений $N$ преимущество метода с поворотами начинает расти. Это логично, поскольку для большего числа наблюдений легче сделать грамотное разбиение данных на классы и степень переобученности на данных будет меньше, что особенно важно для более гибкого и склонного к переобучению метода с поворотами.

По табл. 4 видно, что размерность пространства признаков не играет существенной роли в линейном случае, когда поверхность, разделяющая классы, проста. С усложнением этой поверхности на нелинейный случай, рост размерности пространства

*Таблица 4*

**Зависимость числа суммарных баллов от типа данных и размерности данных *D***

| *D* | 2.0 | 3.0 | 5.0 | 7.0 | 15.0 | сумма: |
|---|---|---|---|---|---|---|
| тип данных: | | | | | | |
| шар | 13 | 11 | 1 | –3 | –2 | 20 |
| линейный | 10 | 13 | 17 | 14 | 16 | 70 |
| многогранник | 15 | 11 | –1 | –4 | –3 | 18 |
| парабола | 7 | 2 | –3 | –9 | –19 | –22 |
| сумма: | 45 | 37 | 14 | –2 | –8 | 86 |

ведет к усложнению поверхности – ее становится сложнее аппроксимировать, и выше риск переобучения, что и отражается в снижении относительной точности бустинга с поворотами.

**4. Заключение**

Применение случайных поворотов в композициях прогнозирующих моделей было предложено в [5] и является разумным подходом для повышения разнообразия базовых моделей и улучшения точности их объединения. Но в [5] не было изучено применение данного подхода к алгоритму бустинга. В данной работе было изучено влияние случайных поворотов на точность бустинга для широкого класса модельных данных. Было обнаружено, что преимущество предложенного подхода тем выше, чем больше размер обучающей выборки, меньше размерность признаков и меньше уровень зашумленности данных. Метод дает явное преимущество для объектов линейно или кусочно-линейно разделимых на классы, а также для случаев, когда классы разделимы поверхностью, симметричной относительно начала координат. Вместе с этим, метод менее предпочтителен для случаев, когда классы разделимы фигурами, имеющими в качестве оси симметрии одну из осей признакового пространства. Темой для дальнейших исследований может быть изучение предложенного подхода на реальных данных, а также использование не случайных поворотов, а таких поворотов, которые лучше всего разделяют объекты по разным классам.

**Литература**

1. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. 2-oe изд. – Stanford, USA: Springer, 2009.

2. Abbott D. Why ensembles win data mining competitions. // Predictive Analytics Centre of Excellence Tech Talks, University of California, San Diego. http://pace.sdsc.edu/sites/pace/files/PACE_Abbott_WhyModelEnsemblesWinDataMiningCompetitions_20121114.pdf / 2012.

3. Китов В.В. Практические аспекты машинного обучения. // Открытые системы. СУБД. №1 / 2016. с. 14–17.

4. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. https://arxiv.org/abs/1603.02754. DOI: 10.1145/2939672.2939785.

5. Blaser R., Fryzlewicz P. Random Rotation Ensembles // Journal of Machine Learning Research №17 / 2016. с. 1–26.

6. Ozols M. How to generate a random unitary matrix. http://home.lu.lv/~sd20008/papers/essays/Random%20unitary%20[paper].pdf / 2009.

**References**

1. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. 2-e izd. – Stanford, USA: Springer, 2009.

2. Abbott D. Why ensembles win data mining competitions. // Predictive Analytics Centre of Excellence Tech Talks, University of California, San Diego. http://pace.sdsc.edu/sites/pace/files/PACE_Abbott_WhyModelEnsemblesWinDataMiningCompetitions_20121114.pdf / 2012.

3. Kitov V.V. Practical aspects of machine learning. // Open systems. SUBD. №1 / 2016. p. 14–17.

4. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. https://arxiv.org/abs/1603.02754. DOI: 10.1145/2939672.2939785.

5. Blaser R., Fryzlewicz P. Random Rotation Ensembles // Journal of Machine Learning Research №17 / 2016. p. 1–26.

6. Ozols M. How to generate a random unitary matrix. http://home.lu.lv/~sd20008/papers/essays/Random%20unitary%20[paper].pdf / 2009.