

Ходорченко Антон Андреевич

### **МОБИЛЬНЫЕ РЕШАЮЩИЕ ПРАВИЛА С МАШИНЫМ ОБУЧЕНИЕМ В ОБЛАКЕ**

В статье рассматривается дистанционный метод создания моделей принятия решений для использования на портативных устройствах с наличием канала передачи данных. Делегирование вычислений для построения моделей позволяет получить выгоду на портативных устройствах, такую как экономия времени, вычислительных ресурсов и энергозатрат, при незначительных системных ограничениях. Проведены теоретические и практические сравнения с методом локальных вычислений и методом облачного распознавания данных.

Адрес статьи: [www.gramota.net/materials/1/2015/7/37.html](http://www.gramota.net/materials/1/2015/7/37.html)

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

### **Альманах современной науки и образования**

Тамбов: Грамота, 2015. № 7 (97). С. 138-142. ISSN 1993-5552.

Адрес журнала: [www.gramota.net/editions/1.html](http://www.gramota.net/editions/1.html)

Содержание данного номера журнала: [www.gramota.net/materials/1/2015/7/](http://www.gramota.net/materials/1/2015/7/)

### **© Издательство "Грамота"**

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: [www.gramota.net](http://www.gramota.net)

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: [almanac@gramota.net](mailto:almanac@gramota.net)

и уравнение (33) являются, соответственно, уравнением движения свободной вихревой поверхности и уравнением изменения поверхностных вихрей свободной вихревой поверхности.

**Заключение.** Из выражения (30) следует, что вектором скорости произвольной точки свободной вихревой поверхности можно назвать любой вектор  $\vec{V}_0$  с нормальной составляющей, равной нормальной составляющей скорости жидкости с разных сторон свободной вихревой поверхности, и касательные составляющие которого обращают в ноль правую часть выражения (30). В частности, скорость произвольной точки свободной вихревой поверхности можно определять как скорость, которая дается формулой (32). Такое определение скорости произвольной точки свободной вихревой поверхности и принято в теории несущей поверхности [1-3].

#### Список литературы

1. Белоцерковский С. М., Ништ М. И. Отрывное и безотрывное обтекание тонких крыльев идеальной жидкостью. М.: Наука, 1978. 351 с.
2. Зайцев А. А. Теория несущей поверхности: математическая модель, численный метод, расчет машущего полета. М.: Наука, 1995. 160 с.
3. Крылов Д. А., Сидняев Н. И., Федотов А. А. Обтекание колеблющегося крыла потоком идеальной несжимаемой жидкости // Труды МГТУ им. Н. Э. Баумана. М.: Изд-во МГТУ им. Н. Э. Баумана, 2013. № 608. С. 74-92.
4. Лойцянский Л. Г. Механика жидкости и газа. М.: Дрофа, 2003. 840 с.
5. Седов Л. И. Механика сплошной среды. М.: Наука, 1976. Т. 1. 536 с.; Т. 2. 576 с.
6. Федотов А. А. Исследование вихревой структуры за колеблющимся крылом // Современные методы теории краевых задач: материалы Воронежской весенней математической школы «Понтрягинские чтения – XIX». Воронеж: ВГУ, 2008. С. 213-214.
7. Федотов А. А. Расчет вихревой структуры за крылом, работающим в режиме создания силы тяги // Альманах современной науки и образования. 2008. № 7 (14). С. 225-229.
8. Федотов А. А. Структура вихревого следа за крылом, работающим в режиме нормального трепещущего полета // Вестник МГУ. Серия 1. Математика. Механика. 1990. № 3. С. 42-46.

#### ABOUT THE EQUATION OF MOTION OF FREE VORTEX SURFACE IN IDEAL INCOMPRESSIBLE FLUID

Fedotov Anatolii Aleksandrovich, Ph. D. in Physical-Mathematical Sciences, Associate Professor  
Bauman Moscow State Technical University  
le-tail@list.ru

The article examines a nonlinear problem about the unsteady flow of the infinitely thin wing by ideal incompressible fluid. The flow takes place with the formation of the surface of the tangential discontinuity of fluid speed descending to the stream from the edge of the wing motion. The wing and the surface of tangential discontinuity are represented, respectively, as a lifting surface and free vortex sheath. It is shown that the velocity definition of free vortex sheath accepted in the theory of lifting surface is a sufficient condition for the closure of equations system describing the unsteady flow by ideal incompressible fluid of the infinitely thin wing of final amplitude.

*Key words and phrases:* wing; surface of tangential discontinuity; lifting surface; free vortex sheath; surface vortex vector.

УДК 004.75

#### Технические науки

*В статье рассматривается дистанционный метод создания моделей принятия решений для использования на портативных устройствах с наличием канала передачи данных. Делегирование вычислений для построения моделей позволяет получить выгоду на портативных устройствах, такую как экономия времени, вычислительных ресурсов и энергозатрат, при незначительных системных ограничениях. Проведены теоретические и практические сравнения с методом локальных вычислений и методом облачного распознавания данных.*

*Ключевые слова и фразы:* машинное обучение; модель принятия решений; PMML; распределенные вычисления; облачные вычисления; мобильное устройство.

**Ходорченко Антон Андреевич**

Новосибирский государственный университет  
anton.khodorchenko@gmail.com

#### МОБИЛЬНЫЕ РЕШАЮЩИЕ ПРАВИЛА С МАШИНЫМ ОБУЧЕНИЕМ В ОБЛАКЕ<sup>©</sup>

Рост производительности современных ПК и портативных устройств, таких как смартфоны, дает возможность использовать методы машинного обучения для решения практически любой задачи с целью

улучшения качества получаемых результатов (например, точность, скорость и персонализация). Среди системных ограничений при решении задач машинного обучения на портативных устройствах можно выделить:

1. Время построения модели принятия решений – производительность портативных устройств уступает серверным решениям и может быть недостаточной при больших объемах данных.

2. Энергоресурсы – обработка среднего объема выборки может потратить значительное количество энергии, что может быть недопустимо на портативных устройствах.

3. Помехи пользовательской деятельности – обработка данных на портативном устройстве может сильно помешать пользователю использовать это устройство в своих целях.

Одним из способов разрешить часть системных ограничений при решении этих задач является использование удаленных вычислительных серверов для машинного обучения. Хотя этот способ и решает три системных ограничения, приведенных выше, но он накладывает новые ограничения, такие как:

1. Канал для передачи данных, наличие которого необходимо каждый раз при проведении любых вычислений.

2. Время распознавания данных – даже самые простые вычисления будут проходить не быстрее, чем время передачи данных в обе стороны.

Если наличие канала передачи данных для смартфонов не является проблемой, хотя и может накладывать ограничения в плане стоимости использования, то время распознавания данных может оказаться критическим, когда нужно принять решение за доли секунды.

Возможным выходом из этой ситуации является создание системы, которая будет использовать удаленные вычислительные сервера для создания моделей принятия решений, в то время как само распознавание будет происходить непосредственно на портативном устройстве.

Преимуществами данного подхода являются ослабление ограничения на наличие канала данных и полное разрешение ограничения на время распознавания данных. Действительно, имея модель принятия решений на портативном устройстве, необходимость в канале данных возникает только для обучения/переобучения, которые можно отложить до появления возможности передачи данных (например, появление бесплатной *Wi-Fi* сети). В отсутствие канала можно использовать более старую версию модели принятия решений. А так как вычислительная сложность распознавания по модели в разы уступает сложности обучения, и сложность распознавания данных обычно линейна и зависит лишь от размера модели принятия решений, то время распознавания данных будет крайне мало.

Такую систему можно реализовать в клиент-серверном варианте, где сервером будет выступать система с использованием REST-запросов с синтаксисом стандарта PMML (Prediction Model Markup Language) – языком на основе XML для использования в области анализа данных, а клиентом – библиотеки для портативных платформ, таких как *Android*, *iOS*, *Windows Phone* и других. Серверную часть, отвечающую за машинное обучение, можно делегировать готовым реализациям, подключая необходимые библиотеки методологий машинного обучения, такие как *FRiS*, *KMean*, *Forel*, *Mahout*, *Spark* и другие. Это позволит использовать систему на различных устройствах и в различных целях, выбирая необходимый инструментарий под каждую задачу.

#### Обзор существующих решений

Существует достаточно много решений в данной области, в том числе и от известных компаний, такие как *Microsoft Azure ML*, *Google Prediction API*, *Prediction API*, *Amazon Machine Learning*, *Prediction.IO* и другие.

Решения от *Amazon* и *Microsoft* предоставляют услуги по моделям обслуживания *IaaS* и *PaaS*, что позволяет получить большую гибкость при конфигурировании конкретных задач. Решение от *Google* абстрагируется от понятий платформы и инфраструктуры и предоставляет решение напрямую для приложения. Для этого существует свой API. Все перечисленные решения платные. *Prediction.IO* предоставляет бесплатный продукт с открытым исходным кодом, который может быть развернут на собственных серверах.

Каждый продукт имеет свои достоинства и недостатки, однако у них есть один общий недостаток – в них отсутствует возможность экспортирования обученной модели принятия решений. Это означает, что при необходимости распознать новые данные их нужно отправлять на сервер и получать распознанные данные обратно. Данный недостаток означает наличие указанных выше ограничений, которые могут быть недопустимы.

#### Составление требований к системе

Для составления требований к системе рассмотрим сценарии использования системы. Существуют два основных сценария использования – передача обучающей выборки на сервер и получение готовой модели принятия решений. Добавление дополнительных сценариев упростит использование системы и избавит от передачи избыточной информации:

- возможность дополнения выборки для переобучения, что избавит от необходимости повторного отправления больших объемов данных;
- возможность проверки состояния вычислительной задачи и готовности модели принятия решений.

Также стоит рассмотреть сценарий использования, по которому множество устройств собирают итоговую выборку и дополняют её личными данными. Такая модель влечет за собой множество других сценариев использования, однако ключевым будет сценарий, в котором модель принятия решений строится на основе общей «мастер»-выборки и персональной выборки вместе.

Вопрос аутентификации не актуален в плане конфиденциальности данных, так как модель принятия решений представляет собой набор численных данных, бесполезных без описания, и данных, на которых её можно применить. Поэтому возникает вопрос идентификации вычислительной задачи и идентификации легитимности пользователя. Идентификация вычислительной задачи решается путем присвоения уникального

идентификатора, по которому можно получить модель принятия решений или же дополнить обучающую выборку. Идентификация легитимности пользователя решается путем выделения уникального ключа (цифровой подписи, пароля или другого способа). При этом пользователями системы являются разработчики приложений, встраивающие функционал в свои приложения.

Исходя из всех перечисленных требований, можно составить общий план взаимодействия с системой:

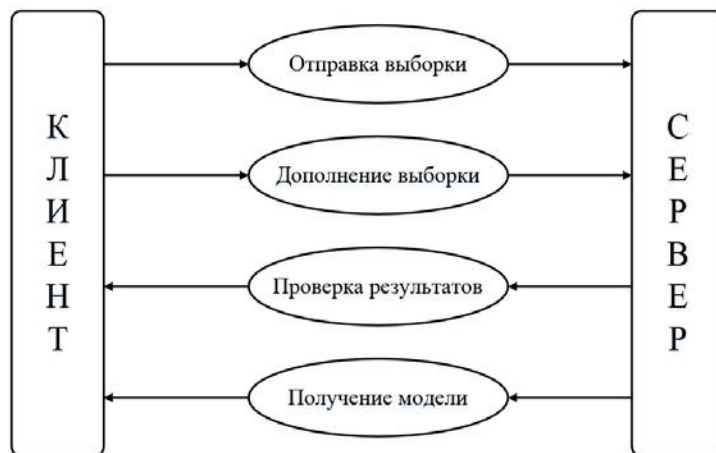


Рис. 1. Способы взаимодействия клиента с сервером

#### Архитектурное решение и реализация

Для взаимодействия с пользователем необходимо реализовать REST-сервис с синтаксисом языка PMML. Для вычислительных ресурсов – использовать подключение к готовым серверам, предоставляющим функционал алгоритмов машинного обучения или же использовать собственные распределенные вычислительные ресурсы для запуска необходимых алгоритмов. Для хранения – использовать распределенную базу данных с возможностью доступа вычислительных серверов напрямую. Для управления – использовать механизм очередности, который будет хранить последовательность вычислительных задач. В качестве этого механизма можно использовать как стандартную очередь (FIFO), так и другие варианты. Например, с учетом предлагаемой области применения, актуальной будет очередь, состоящая из двух параллельных очередей. Первая очередь хранит вычисления, связанные с первичной обработкой обучающей выборки, вторая хранит вычисления, связанные с переобучением на основе дополненных выборок. При наличии вычислительных ресурсов приоритетно вычисляются задачи из первой очереди. Если первая очередь пустая, то вычисляются задачи из второй очереди. Это позволит быстрее получить готовую модель принятия решений для новых задач, в то время как остальные задачи могут пользоваться «устаревшей» моделью. Данный способ актуален при построении моделей, пытающихся улучшить функционал на основе персонализированных данных, когда пользователю важна скорость получения работоспособной модели, в то время как он готов мириться с неточностью «устаревшей» модели. Однако данный способ не универсален, и при наличии большого числа новых задач задачи переобучения могут ожидать недопустимое количество времени, а при отсутствии задач переобучения данный метод превратится в обычную очередь.

В качестве технологии для создания REST-сервиса была выбрана *ASP.Net*. Для хранения данных выбрана распределенная *noSQL* СУБД *Apache Cassandra*.

#### Производительность

Для начала оценим производительность портативных устройств и вычислительных серверов. Под портативными устройствами будем понимать смартфоны, коммуникаторы и КПК.

Так как рынок сильно разнится, то будем исходить из средних показателей и проведем синтетическое сравнение. Например, производительность *CPU* процессора *Apple A8* составляет 1,4 *GFlops*, а *GPU* – 115 *GFlops*. Однако этот процессор если и не является флагманом на данный момент, то являлся таковым полгода назад. Исходя из этого, возьмем за параметры мощного устройства 1 *GFlops* для *CPU* и 100 *GFlops* для *GPU*.

Рынок процессоров и графических процессоров для персональных компьютеров и серверных решений также сильно разнится. Для современных ПК возьмем *Intel Core i7-5960X* с производительностью 350 *GFlops* и графический процессор *AMD Radeon R9 295X2* с производительностью 5,5 *TFlops*. Серверный процессор *Intel Xeon E5-2699* способен достичь отметки 750 *GFlops*.

Скорость передачи данных по протоколам *Wi-Fi 802.11n* и *HSPA+* может достигать 150 Мбит/с. Время установления связи – от долей секунд до нескольких десятков секунд, однако при активном соединении и хорошем качестве связи составляет около 1 секунды.

Многие алгоритмы по сложности вычисления сопоставимы с  $O(n^2)$ . 100 МБ выборка из 10 признаков несет в себе 2,6 миллиона объектов.



Рис. 2. Обычный менеджер очереди и менеджер с параллельными очередями  
Все это объединяется основной управляющей программой.

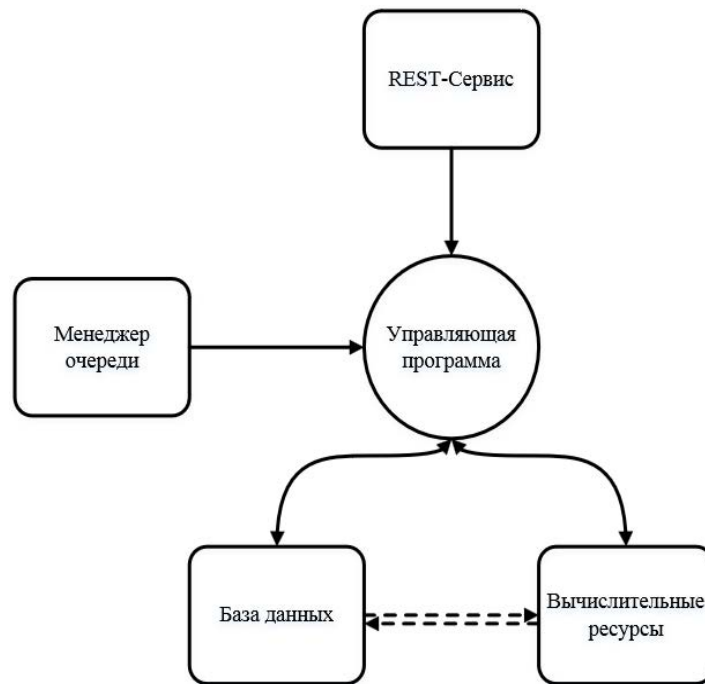


Рис. 3. Архитектура системы

Таким образом, передача 100 МБ информации на максимальной скорости займет около 10 секунд. Вычисление с количеством операций  $n^2$  и производительностью 1 GFlops потребует 2 часа, с производительностью 120 GFlops – 1 минуту, с производительностью 750 GFlops – 10 секунд.

Сравним время работы системы, время вычислений, проводимых на портативном устройстве, и время вычислений на полностью облачных ресурсах по двум показателям – времени обучения и времени принятия решения.

Таблица 1. Синтетическое сравнение производительности

	Время обучения	Время принятия решения
Облачное построение модели	20* секунд	0*(1) секунд
Локальные вычисления	60 секунд	0** секунд
Облачные вычисления	20* секунд	1 секунда

Примечание:

\* время вычисления при наличии свободных ресурсов на сервере;

\*\* время принятия решения крайне мало на фоне производительности и объемов требуемых вычислений.

Однако стоит учесть, что при работе с облачными вычислениями выполнение задачи может быть не мгновенным. Таким образом, обучение в системе за 20 секунд возможно, если вычислительные ресурсы свободны. А 0 (1) секунд требуется на принятие решения в случае наличия готовой модели (и в случае необходимости её получения с сервера).

Для реального сравнения времени обучения возьмем алгоритм KNN [2, р. 90] из библиотеки Weka, выборку Sесom с 591 признаком и 1567 элементами, также в ней присутствуют пробелы, мобильное устройство

HighScreen Boost 2 SE и ПК на базе процессора *Intel i5 2500k*. Сравнение будем проводить по четырем моделям: *J48*, *PART*, *KNN* и *Decision Stump* (одноуровневое дерево принятия решений).

Получившиеся результаты описаны в Таблице 2.

**Таблица 2.** Сравнение производительности обучения на основе *KNN*

	Мобильное устройство	ПК
<i>J48</i>	28000 мс	630 мс
<i>PART</i>	10900 мс	350 мс
<i>KNN</i>	46000 мс	835 мс
<i>Decision Stump</i>	4000 мс	127 мс

Данное сравнение показывает, что разница в производительности может быть около 50 раз.

Для сравнения времени распознавания проведем распознавание по готовой модели принятия решений на мобильном устройстве и через клиент-серверную архитектуру. Была взята модель на основе выборки *Madelon*, состоящая из 1820 элементов и 500 признаков, без пробелов. Проводилось 3 сравнения производительности по 2 выборкам для распознавания.

Первая выборка состояла из 10 элементов и 100 итераций по выборке. Вторая выборка состояла из 1 элемента и 1000 итераций по выборке. Результаты сравнения времени распознавания 1 элемента представлены в Таблице 3.

**Таблица 3.** Сравнение производительности распознавания

	10 элементов / 100 итераций	1 элемент / 1000 итераций
Локальное распознавание	72 мс	76 мс
Облачное распознавание (Wi-Fi)	12 мс	24 мс
Облачное распознавание (3G)	66 мс	263 мс

Данные результаты показывают, что распознавание через *Wi-Fi* быстрее за счет качества связи внутри локальной сети. Распознавание через 3G становится быстрее при повышении количества операций – сглаживаются транспортные затраты и используется преимущество мощности вычислительных серверов, т.к. объем выборки и количество вычислений возрастают. При более компактной модели результаты локального распознавания будут еще более выгодными на фоне облачного распознавания. Также в облачном распознавании возможно присутствие случайных задержек и задержек установления соединения, что может негативно сказаться на результатах.

### Вывод

Накладывая ограничение необходимости передачи данных, по сравнению с методом локальных вычислений на портативных устройствах, данный метод дает прирост производительности и экономии ресурсов.

По сравнению с полностью облачными вычислениями, данный метод дает два преимущества: первое из них – это возможность мгновенного принятия решений за счет локальной модели принятия решений, и второе – это возможность передавать обучающие выборки при наличии канала передачи данных.

Недостатком данного метода по сравнению с обоими перечисленными является возможное отсутствие актуальной модели принятия решений. Однако данный недостаток может оказаться не столь значимым в большинстве задач.

### Список литературы

1. **Загоруйко Н. Г.** Прикладные методы анализа данных и знаний. Новосибирск: Изд. ИМ СО РАН, 1999. 273 с.
2. **Larose D. T.** *Discovering Knowledge in Data: An Introduction to Data Mining.* John Wiley & Sons, Inc., 2005.
3. **Zagoruiko N. G., Borisova I. A., Dyubanov V. V., Kutnenko O. A.** Methods of Recognition Based on the Function of Rival Similarity // *Pattern Recognition and Image Analysis.* 2008. Vol. 18. № 1. P. 1-6.

### MOBILE DECIDING RULES WITH MACHINE LEARNING IN THE CLOUD

**Khodorchenko Anton Andreevich**

*Novosibirsk State University  
anton.khodorchenko@gmail.com*

The article discusses the remote method of creating decision-making models for the use on portable devices with the data channel. The delegation of computations for modeling enables getting benefit on portable devices, such as saving time, computational resources and energy inputs with little systemic constraints. Theoretical and practical comparisons with the method of local computing and the method of data cloud recognition are carried out.

*Key words and phrases:* machine learning; decision-making model; PMML; distributed computing; cloud computing; mobile device.