

Инструментальные средства для автоматизированной количественной оценки метакомпетенций учащихся. Обзор

10, октябрь 2015

DOI: 10.7463/1015.0821623

Белоножко П. П.^{1,*}, Белоус В. В.¹,
д.ф.-м.н. Карпенко А. П.¹, Храмов Д. А.²

УДК: 519.6

¹МГТУ им. Баумана, Москва, Россия

²Институт технической механики Национальной Академии наук Украины
и Государственного космического агентства Украины,
Днепропетровск, Украина

^{*}byelonozhko@mail.ru

Введение

Рассматривается задача разработки комплекса научно-технических решений, направленных на создание системы количественной оценки метапредметных, метакогнитивных и метакреативных компетенций учащихся (уникальный идентификатор прикладных научных исследований (проекта) RFMEFI57714X0135).

Содержанию рассматриваемых в рамках исследования метакомпетенций и формам их проявления посвящены работы [1 – 3]. Предполагается, что уровень владения конкретным учащимся умениями и навыками, соответствующими данному типу метакомпетенции, может быть интегрально оценен (например, с привлечением эксперта) с помощью нормализованной целочисленной шкалы [4]. С другой стороны, каждому типу метакомпетенции ставится в соответствие некоторая совокупность способностей, умений, навыков и т.д., которая для конкретного учащегося может быть формализована в виде вектора индикаторов, содержащего оценки по каждому элементу упомянутой совокупности, также сформированные с помощью нормализованной целочисленной шкалы. Разрабатываемые методы сбора данных и получения в результате их обработки численных значений индикаторов основываются как на непосредственном тестировании, так и на анализе поведения учащихся в различных пользовательских средах, в том числе системах дистанционного обучения [1 – 4].

Принятый подход к представлению интегральных численных оценок метакомпетенций, а также к формированию вектора индикаторов подразумевает использование методов машинного обучения при создании системы их количественной оценки. Сопоставительный анализ различных методов машинного обучения применительно к задачам исследований приведен в работе [4].

По результатам исследования планируется реализация разработанных методов, моделей и алгоритмов в экспериментальном образце программного комплекса автоматизированной количественной оценки метапредметных, метакогнитивных и метакреативных навыков и умений на основе анализа поведения пользователей в различных обучающих средах. Реализация программного комплекса предполагается на платформе свободно распространяемого программного обеспечения (ПО) с использованием открытых компонентов.

Как отмечается в [5], на начальных стадиях создания систем машинного обучения велика роль разведочного анализа и предварительной подготовки данных (задача подготовки признаков – *feature engineering*), включающих, в том числе, проведение вычислительных экспериментов по их тестовой обработке с использованием альтернативных программных реализаций различных методов машинного обучения. В этой связи представляет практический интерес актуальная информация о наличии ПО, реализующего те или иные методы машинного обучения, позволяющая в ходе исследований оперативно решать вопрос о выборе наиболее эффективного программного инструмента.

С другой стороны, разработка экспериментального образца программного комплекса предполагает использование некоторого универсального ПО, обеспечивающего единообразный подход к решению задач, непосредственно связанных с реализацией методов машинного обучения, а также вспомогательных по отношению к ним задач – предварительная обработка данных, визуализация результатов и пр.

Для автоматизированной оценки метапредметных, метакогнитивных и метакреативных навыков и умений учащихся планируется использовать следующие типы машинного обучения (*machine learning*):

- 1) обучение с учителем (*supervised learning*);
- 2) обучение без учителя (*unsupervised learning*);
- 3) обучение с частичным привлечением учителя (*semi-supervised learning*);
- 4) обучение с подкреплением (генетические алгоритмы) (*reinforcement learning, genetic algorithm, GA*);
- 5) активное обучение (*active learning*);
- 6) многовариантное обучение (*multiple instance learning, MIL*);
- 7) многозадачное обучение (*multi task learning, MTL*).

Предполагается использовать следующие методы машинного обучения:

- 1) линейная регрессия (*linear regression*);
- 2) логистическая регрессия (*logistic regression*);

- 3) искусственные нейронные сети, в том числе спайковые нейронные сети (artificial neural networks, neural network, multilayer perceptron, MLP, probabilistic neural network, PNN, self-organizing maps, SOM, spiking neural network, SNN);
- 4) метод опорных векторов (support vector machine, SVM);
- 5) метод ближайших соседей (nearest neighbors algorithm, NN).

В рамках обзора объектом поиска являлось свободное ПО — программы, пакеты программ и библиотеки — с открытым исходным кодом, реализующее указанные выше типы и методы машинного обучения. Предполагалось, что библиотеки не только относятся к свободному ПО сами, но и используются совместно с подобным ПО. Поэтому свободные пакеты расширений к коммерческому ПО, например, к MATLAB, в обзоре не рассматривались.

Представляющее интерес ПО должно поддерживаться в актуальном состоянии (текущая стабильная версия должна иметь возраст не более двух лет) и регулярно обновляться.

Основным критерием выбора ПО предложено считать универсальность — желательно использовать один инструмент, реализующий все методы обучения для всех указанных типов или такой, к которому при необходимости можно было бы подключить ПО, реализующее недостающие возможности. Поэтому особо отмечались возможности совместной работы различного ПО.

Важно, чтобы выбранное ПО обладало активным и достаточно большим сообществом пользователей, с помощью которого можно было бы решать технические проблемы реализации методов.

Следует также обратить внимание на вопрос повышения производительности ПО, т. е. на скорость обработки данных и возможности работы с большими массивами данных. На стадии разработки экспериментального образца программного комплекса эти характеристики менее существенны, однако они могут оказаться важными впоследствии.

Исходными для поиска информации послужили веб-страницы [6–9]. Поиск проводился в два этапа. На первом этапе ставилась цель найти максимальное число программных средств, удовлетворяющих указанным требованиям, проанализировать их возможности и выделить наиболее перспективное ПО.

Найденное в результате первого этапа поиска ПО можно разделить на два вида:

- специализированное ПО, реализующее один или несколько методов машинного обучения;
- универсальное ПО для решения группы задач машинного обучения.

На втором этапе более подробно рассматривалось выбранное ПО и возможности его использования для решения задач количественной оценки метакомпетенций учащихся.

Предлагаемый обзор выполнен применительно к задачам конкретного прикладного исследования, однако полученные результаты могут быть также использованы для решения широкого круга теоретических и прикладных задач, связанных с необходимостью программной реализации различных методов машинного обучения.

1. Специализированное ПО, реализующее один или несколько методов машинного обучения

[gretl](#) (Gnu Regression, Econometrics and Time-series Library) [10] — кроссплатформенный пакет, написанный на языке С и реализующий, в частности, различные виды регрессии. Возможность обмена данными и результатами расчетов с R, GNU Octave и Python.

[libnabo](#) [11] — библиотека, реализующая метод k-ближайших соседей для пространств небольшой размерности. Написана на C++, использует шаблоны языка, а также библиотеки [Eigen](#) (матричные операции и линейная алгебра) [12] и [Boost](#) [13].

[LIBSVM](#) [14], [LIBLINEAR](#) [15] — взаимосвязанные библиотеки, разработанные в Национальном университете Тайваня. LIBSVM реализует метод опорных векторов, а LIBLINEAR — линейный вариант метода опорных векторов и логистические регрессионные модели, обучаемые с помощью метода покоординатного спуска (coordinate descent algorithm). Кроме того, существует пакет [LIBSVM Tools](#) [16], расширяющий возможности LIBSVM. Обе библиотеки широко используются в виде компонентов универсального ПО для решения задач машинного обучения, например, в Orange и scikit-learn, имеют интерфейсы для работы с Python, R, GNU Octave, Scilab, Weka, PHP и др. языками, а также модификации для работы с [CUDA](#) [17]. Существуют версии исходного кода LIBSVM и LIBLINEAR на C++ и на Java.

[Mixmod](#) [18] — пакет для решения задач классификации количественных и качественных данных как с учителем, так и без него. В частности, пакет использует методы кластеризации и дискриминантного анализа. Основу пакета составляет библиотека [mixmodLib](#) [19], реализованная на C++. Существует также графический интерфейс и расширение для работы с R — пакет Rmixmod.

[MLPACK](#) [20] — библиотека машинного обучения, написанная на языке C++, основанная на библиотеке матричных операций [Armadillo](#) [21] и реализующая ряд методов регрессии и кластеризации. Отличительной особенностью MLPACK является высокое быстродействие реализованных алгоритмов. В [22] показано, что по этому показателю MLPACK, как правило, превосходит своих основных конкурентов — scikit-learn, SHOGUN, Weka, mlpy, и MATLAB.

Краткий перечень возможностей MLPACK:

- коллаборативная фильтрация (collaborative filtering);
- одноуровневые деревья принятия решений (decision stumps);
- вычисление евклидовых минимальных остовных деревьев;
- смешанные гауссовские модели;
- скрытые марковские модели;
- ядерный метод главных компонент;
- кластеризация методом k-средних;
- метод регрессии наименьшего угла (least angle) (LARS/LASSO);
- линейная регрессия (метод наименьших квадратов);

- локально-чувствительное хеширование и алгоритм двойного дерева для поиска ближайших соседей;
- логистическая регрессия;
- наивный байесовский классификатор;
- неотрицательная матричная факторизация (non-negative matrix factorization);
- перцептроны;
- анализ главных компонент (PCA);
- RADICAL (анализ независимых компонент).

Доступ к возможностям MLPACK из R осуществляется с помощью пакета RcppMLPACK.

1.1. Инструменты машинного обучения, реализованные на языке JavaScript

Для создания интеллектуальных веб-приложений могут показаться интересными инструменты машинного обучения, реализованные на языке JavaScript. Обзор таких инструментов приведен в [23]. К нему можно лишь добавить библиотеку [Lyric](#) [24], реализующую метод линейной регрессии, а также библиотеки [ConvNetJS](#) [25] и [Synaptic](#) [26], предназначенные для создания и обучения искусственных нейронных сетей. Отметим, что появление платформы [Node.js](#) [27] позволяет проводить обучение на стороне сервера.

1.2. Спайковые нейронные сети

Спайковые нейронные сети [28] отличаются от более ранних моделей нейронных сетей тем, что для обмена сообщениями в них используются импульсы или спайки (*англ. spike* — всплеск, скачок) — кратковременные изменения напряжения. Время задержки между импульсами и частота их следования представляет собой дополнительный способ кодирования информации. Отвечая на входные стимулы, спайковые нейроны способны группироваться в ансамбли, что потенциально увеличивает информационную емкость сети, поскольку позволяет хранить информацию распределенно. При этом одни и те же нейроны могут участвовать одновременно в различных ансамблях.

Наиболее известными программами-симуляторами спайковых нейронов являются [GENESIS](#) (GEneral NEural SIMulation System) [29] и [NEURON](#) [30], но эти программы предназначены, скорее, для создания детальных биофизических моделей изолированного нейрона, чем для моделирования сетей из подобных нейронов (хотя в последних версиях NEURON возможности моделирования сетей существенно расширились). Напротив, пакет [Brian](#) [31], написанный на Python предназначен для моделирования сетей спайковых нейронов. Brian распространяется в виде пакета расширения языка Python.

Краткий обзор ПО для моделирования спайковых нейронов и сетей из них приведен в [28]. Более подробный обзор сведений по спайковым нейронным сетям и инструментам для их моделирования приводится в [32]. Сравнительный анализ различных симуляторов выполнен в [33]. Здесь мы остановимся на двух пакетах моделирования спайковых ней-

ронных сетей, получивших активное развитие в последние годы, в силу чего их свойства недостаточно отражены в указанных выше работах.

Пакет [NEST](#) (Neural Simulation Technology) [34] предназначен для моделирования спайковых нейронных сетей любого масштаба. Процесс моделирования в NEST выстраивается согласно логике электрофизиологического эксперимента, а исследуемая «нервная система» задается экспериментатором. NEST может использоваться как самостоятельное приложение и имеет ряд интерфейсов для работы в интерактивном режиме, в частности, PyNEST и PyNN. PyNEST представляет собой подмножество команд интерпретатора Python, которые дают доступ к вычислительному ядру NEST. PyNN — язык для моделирования нейронных сетей, основанный на Python, но, в отличие от PyNEST, не зависящий от конкретной программы-симулятора сети. Помимо NEST, PyNN используется также в симуляторах [30, 31].

Пакет [Nengo](#) (Neural Engineering Object) [35], созданный в Университете Ватерлоо (Онтарио, Канада), ориентирован на моделирование крупномасштабных нейронных сетей и работает как под управлением скриптов, так и с помощью графического интерфейса. Последний позволяет создавать и соединять ансамбли нейронов средствами визуального моделирования (рис. 1). В отличие от традиционного нейронного моделирования в Nengo не требуется использовать правило обучения, чтобы найти весовые коэффициенты нейронной сети. Вместо этого в Nengo для поиска весов используется [Neural Engineering Framework](#) (NEF) [36]. Подробное описание этого фреймворка приведено в [37].

Кроме того, в Nengo используется адаптированный формализм теории управления, что позволяет реализовать средствами пакета сложные динамические системы, в частности, различные колебательные системы, хаотические аттракторы и фильтры Калмана.

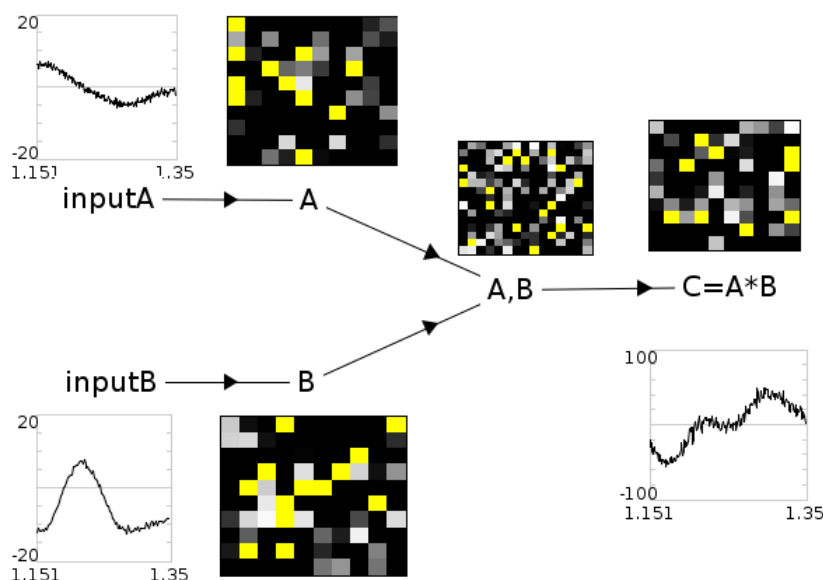


Рисунок 1 — Пример построения нейронной сети средствами визуального моделирования в пакете Nengo [35]

Таким образом, имеющиеся в Nengo инструменты позволяют создать огромное разнообразие моделей, в их числе *Spaun* — на сегодняшний день крупнейшая в мире функциональная модель головного мозга [38]. Модель *Spaun* отличается от других моделей мозга, таких как *IBM Watson*, тем, что она разработана, в первую очередь, для того, чтобы моделировать поведение реального мозга. Нейроны модели связаны между собой реалистичным с точки зрения современной физиологии способом, а части математической модели взаимодействуют между собой так, как по мнению современной науки взаимодействуют между собой части реального головного мозга.

1.3. Библиотеки, реализующие некоторые типы и методы машинного обучения

Существует ряд библиотек, вообще говоря, не ориентированных на решение задач машинного обучения, но реализующих некоторые его методы в качестве вспомогательных. Поэтому подобные библиотеки отнесены нами к специализированным.

[Neuroph](#) [39] — кроссплатформенный пакет для объектно-ориентированного моделирования нейронных сетей, написанный на Java. Основной областью применения пакета является распознавание образов. *Neuroph* представляет собой библиотеку Java-классов (таким образом, созданные в нем нейронные сети можно непосредственно использовать в программах на Java), дополненную графической средой *NeurophStudio* (рис. 2) для создания и обучения нейросетей.

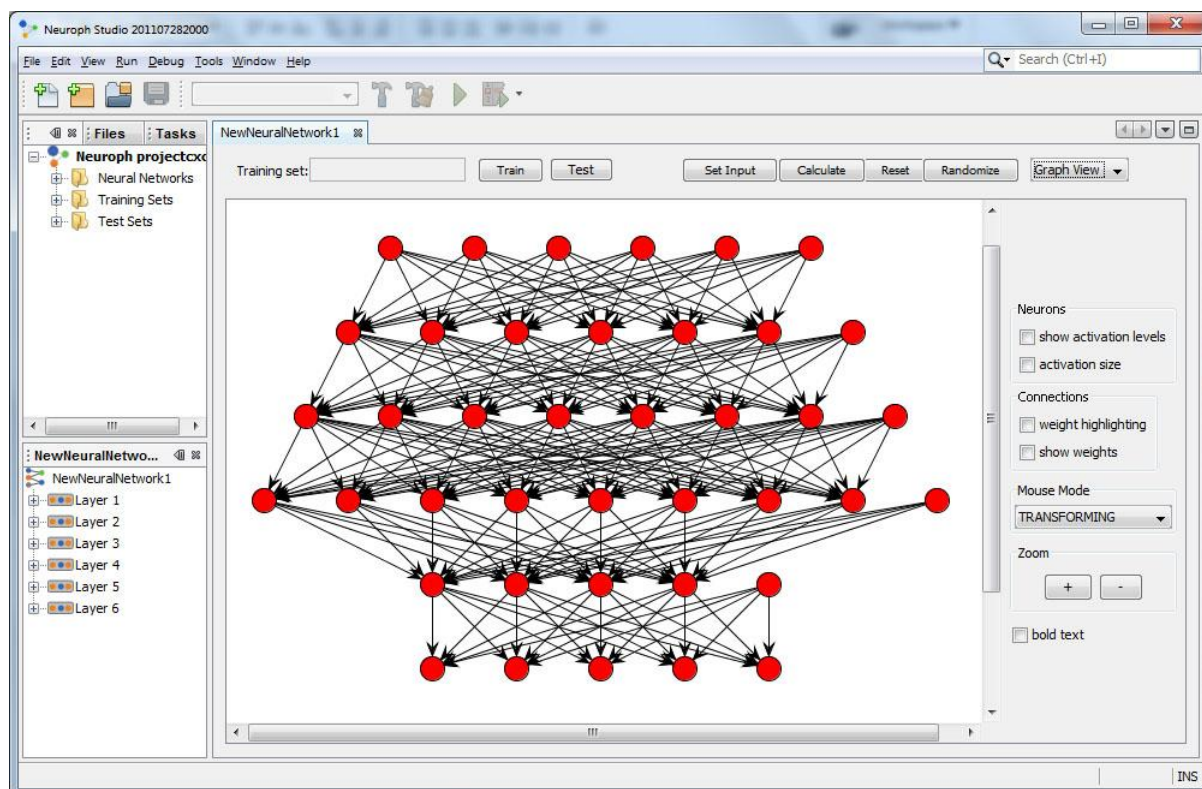


Рисунок 2 — Интерфейс редактора нейронных сетей *NeurophStudio* 2.9 [39]

В *Neuroph* реализованы следующие архитектуры нейронных сетей:

- адалайн (Adaptive Linear Neuron — адаптивный линейный нейрон);
- персептроны, в частности, многослойный персептрон с алгоритмом обратного распространения ошибки;
- сеть Хопфилда;
- двусторонняя ассоциативная память (нейронная сеть Коско);
- сеть Кохонена;
- сеть Хебба;
- сверточная нейронная сеть;
- RBF-сеть.

Из методов машинного обучения в Neuroph реализованы:

- кластеризация методом k-средних;
- метод ближайших соседей (классификация и регрессия).

Существует программный интерфейс, позволяющий использовать возможности Neuroph из пакета Weka.

[OpenCV](#) (Open Source Computer Vision) [40] — реализованная на C++ библиотека алгоритмов компьютерного зрения содержит реализации следующих методов машинного обучения:

- k-ближайших соседей;
- опорных векторов;
- искусственных нейронных сетей.

Существуют полные интерфейсы OpenCV с Python, Java и GNU Octave. В языке R OpenCV запускается с помощью пакета Rcpp. Кроме того, пакет ropencv реализует «обертку» OpenCV в R.

[ROS](#) (Robot Operating System) [41] предлагает коллекцию библиотек и программ для решения задач робототехники. К этим задачам относится, в частности, обучение с подкреплением [42]. В коллекцию ROS входит пакет rosR для вызова функций ROS из R и пакет rospy, реализующий Python API к ROS.

2. Универсальное ПО для решения группы задач машинного обучения

2.1. Библиотеки и фреймворки

[Accord.NET Framework](#) [43] — фреймворк для решения задач машинного обучения на программной платформе .NET, дополненный библиотеками по обработке изображений и звука. Разработан на C#. Предназначен для создания программ, использующих возможности компьютерного зрения, компьютерной обработки звука, обработки сигналов и различных статистических методов.

Возможности, предоставляемые Accord.NET Framework разделены разработчиками на несколько групп:

- классификация (в частности, метод опорных векторов, логистическая регрессия, нейронные сети);

- регрессия (в частности, многопараметрическая линейная регрессия; регрессия, основанная на методе опорных векторов; логистическая регрессия (softmax));
- кластеризация (в частности, методом k-средних);
- распределения (более 40 видов);
- проверка гипотез;
- ядерные методы;
- обработка изображений;
- обработка звука и сигналов;
- методы компьютерного зрения.

[Datumbox Machine Learning Framework](#) [44] — фреймворк, реализованный на Java, ориентированный на быструю разработку приложений для машинного обучения и различных статистических приложений. Кроме того, фреймворк [44] можно использовать в качестве библиотеки, предоставляющей классы для решения следующих задач машинного обучения: регрессия, классификация, кластеризация, понижения размерности, тематическое моделирование (topic modeling), выделение признаков (feature selection) и создание рекомендательных систем. Вот некоторые из реализованных в Datumbox Machine Learning Framework алгоритмов:

- линейный дискриминантный анализ;
- метод максимальной энтропии;
- наивный байесовский классификатор;
- метод опорных векторов;
- бутстрэп-агрегация (бэггинг);
- Adaboost (адаптивный бустинг);
- метод k-средних;
- иерархическая кластеризация;
- смешанные модели процесса Дирихле;
- Softmax-регрессия;
- ординальная логистическая регрессия;
- линейная регрессия;
- пошаговая регрессия;
- анализ главных компонент.

[Deeplearning4j](#) [45] — библиотека, которая реализует алгоритмы глубокого обучения (deep learning algorithms), в частности, методы регрессии и нейронные сети. Все алгоритмы имеют распределенные параллельные версии, интегрированные с [Apache Hadoop](#) [46] и [Spark](#) [47].

Виды нейронных сетей, поддерживаемых Deeplearning4j:

- ограниченные машины Больцмана;
- сверточные нейронные сети;
- рекуррентные сети;

- рекурсивные и глубокие автоассоциаторы (автоэнкодеры);
- сети глубокого доверия (deep belief neural networks).

Библиотека `Deeplearning4j` написана на Java, работает на Java Virtual Machine и, таким образом, совместима с языками Scala и Clojure. Для работы с матрицами и выполнения операций линейной алгебры `Deeplearning4j` использует возможности библиотеки [ND4J](#) [48]. Интерфейс ND4S очень похож на интерфейс Python библиотеки [NumPy](#) [49].

[Dlib](#) [50] — кросс-платформенная библиотека общего назначения, написанная на C++. Один из ее компонентов — [Dlib-ml](#) [51] — реализует ряд методов машинного обучения, в частности, методы регрессии и опорных векторов, а также нейронные сети (например, многослойные перцептроны и сети радиально-базисных функций). `Dlib-ml` предполагается использовать для решения задач машинного обучения на языке C++. Поддерживаются интерфейсы для использования `Dlib` из Java и R.

[Encog](#) [52] — фреймворк, посвященный машинному обучению, доступный на языках Java, .Net и C++. Фреймворк поддерживает метод опорных векторов, искусственные нейронные сети (в частности, сети Кохонена), генетические алгоритмы и генетическое программирование, а кроме того, байесовские сети и скрытые марковские модели. Большинство алгоритмов обучения в `Encog` являются многопоточными и масштабируются на многоядерных аппаратных средствах. `Encog` может также использовать GPU для дальнейшего сокращения времени обработки данных. Фреймворк содержит графический интерфейс пользователя.

[GoLearn](#) [53] — библиотека по машинному обучению для языка Go (и реализованная на Go). Язык Go — появившийся в 2009 г. компилируемый, многопоточный язык программирования, разработанный компанией Google. В настоящее время библиотека `GoLearn` поддерживает следующие методы машинного обучения:

- метод опорных векторов;
- линейная регрессия;
- метод k-ближайших соседей (классификация и регрессия);
- нейронные сети;
- наивный байесовский классификатор;
- решающие деревья;
- random forest.

[H2O](#) [54] — пакет программ, предназначенный для анализа данных большого объема (big-data). Используется для изучения и анализа данных в облачных вычислительных системах, в `Hadoop` [46], а также в обычных операционных системах. `H2O` может запускаться непосредственно из R (пакет `h2o`). Реализует ряд методов машинного обучения, в частности:

- обобщенные линейные модели;
- решающие деревья (деревья принятия решений);
- кластеризация методом k-средних;

- методы выявления аномалий;
- методы глубокого обучения;
- метод Gradient Boosting Machines для решения задач регрессии;
- наивный байесовский классификатор;
- методы оптимизации.

[Intel DAAL](#) (Data Analytics Acceleration Library) [55] — библиотека реализует статистические методы, методы интеллектуального анализа данных, теории распознавания образов, приложения искусственного интеллекта и методы машинного обучения. Алгоритмы DAAL поддерживают решение задач кластеризации, классификации и регрессии и могут выполняться в режимах пакетной, онлайн-овой и распределенной обработки данных. DAAL не использует сторонних библиотек. Для выполнения математических операций используется библиотека Intel MKL (Math Kernel Library) [56], что обеспечивает высокую производительность. DAAL поддерживает кроссплатформенную разработку приложений на языках C++ и Java. Может быть использована совместно с Hadoop, Spark, R и MATLAB. Библиотека доступна в составе Parallel Studio XE, а также отдельно — с коммерческой и бесплатной (community) лицензией (с закрытым исходным кодом).

[JSAT](#) (Java Statistical Analysis Tool) [57] — библиотека для решения задач машинного обучения. Отличительной чертой JSAT является самодостаточность — она написана на чистой Java, без использования сторонних библиотек. Значительная часть кода JSAT поддерживает параллельное выполнение.

В JSAT реализовано большое число алгоритмов машинного обучения, разделенных разработчиками [57] на следующие группы:

- преобразование данных;
- прогнозирование (среди прочего, линейная и логистическая регрессии);
- ядерные алгоритмы;
- деревья принятия решений;
- метод ближайших соседей (сюда отнесены, в частности, сети Кохонена);
- метаалгоритмы (различные виды бустинга).

[JuliaStats](#) [58] — библиотека статистических методов и методов машинного обучения, реализованная на языке Julia. Язык Julia [59] — еще один молодой язык программирования (как и Go он появился в 2009 г.) с динамической типизацией, предназначенный для высокопроизводительных математических вычислений. Синтаксис Julia схож с синтаксисом других языков для научных расчетов (например, MATLAB). В стандартный комплект языка входит JIT-компилятор, благодаря чему, по утверждению авторов языка, приложения, написанные на Julia, не уступают в производительности приложениям, написанным на статически компилируемых языках вроде C или C++.

В настоящее время JuliaStats включает в себя 15 пакетов, в том числе, пакеты для кластеризации и регрессии:

- StatsBase — базовые функции статистики;

- DataArrays — обработка массивов, содержащих пропуски в данных;
- DataFrames — функции для работы с табличными данными;
- Distributions — функции распределения вероятностей;
- MultivariateStats — многомерный статистический анализ;
- HypothesisTests — проверка гипотез;
- MLBase — инструменты для машинного обучения;
- Distances — функции вычисления различных видов расстояний между векторами;
- KernelDensity — ядерная оценка плотности;
- Clustering — алгоритмы кластеризации;
- GLM — обобщенные линейные модели;
- NMF — неотрицательная матричная факторизация;
- RegERMs — минимизация регуляризованного эмпирического риска;
- MCMC — методы Монте-Карло по схеме марковских цепей;
- TimeSeries — анализ временных рядов.

Пакет [RCall.jl](#) [60] позволяет вызывать R из Julia, а пакет R [rjulia](#) [61] — вызывать Julia из R.

[MOA](#) (Massive Online Analysis) [62] — фреймворк, предназначенный для анализа потоков данных (data stream mining) и машинного обучения (регрессия, кластеризация, нейронные сети). Реализован на Java. Имеет как интерфейс командной строки, так и графический интерфейс пользователя. Существует несколько пакетов для работы с MOA из R, а также пакет moa для взаимодействия с Python.

[OpenNN](#) [63] — библиотека, состоящая из C++-классов, позволяющих реализовать нейронные сети и ряд методов обучения с учителем. По словам разработчиков, OpenNN предназначена для продвинутых пользователей с хорошими навыками работы в C++. OpenNN содержит алгоритмы машинного обучения и интеллектуального анализа данных в виде набора классов. Эти классы могут быть встроены в другие программные инструменты, используя прикладной программный интерфейс OpenNN. В связи с этим, графический интерфейс пользователя отсутствует, но некоторые методы поддерживают специфические инструменты визуализации. В целях повышения производительности библиотеку OpenNN можно использовать совместно с OpenMP для реализации многопроцессорных расчетов. Среди областей применения OpenNN: регрессия, ассоциация, распознавание образов (классификация) и предсказание временных рядов (прогнозирование).

[scikit-learn](#) [64] — основная библиотека по машинному обучению, используемая в языке Python. В библиотеке реализовано подавляющее большинство видов и методов машинного обучения. Библиотека имеет обширную документацию, ее поддерживает несколько сотен разработчиков. scikit-learn опирается на научные библиотеки NumPy и [SciPy](#) [49, 65], а также использует библиотеки LIBSVM и LIBLINEAR [14, 15].

[Shogun](#) [66] — пакет, реализующий, среди прочего, методы опорных векторов, регрессии, ряд алгоритмов кластеризации и разновидностей нейронных сетей. Ядро пакета

написано на C++. Существуют интерфейсы для работы с MATLAB, GNU Octave, Python, R, Java, Lua, Ruby и C#.

Другие возможности Shogun:

- графический интерфейс пользователя;
- бинарная и многоклассовая классификация;
- регрессия;
- обучение с частичным привлечением учителя;
- многозадачное обучение (multitask learning);
- параллелизация кода;
- визуализация результатов;
- обработка изображений.

[Torch](#) [67] представляет собой одновременно фреймворк для машинного обучения и предметно-ориентированный скриптовый язык программирования, основанный на Lua. Torch предлагает реализацию методов регрессии, опорных векторов, кластеризации и др. Torch обеспечивает поддержку вычислений с GPU и интерфейс с языком C. Более подробная информация об этом фреймворке содержится в работе [68].

Другие возможности Torch:

- поддержка N-мерных массивов;
- функции для работы с матрицами, выделения части массива;
- функции линейной алгебры;
- функции численной оптимизации;
- наличие версий под iOS и Android.

Torch состоит из множества пакетов, среди которых к области машинного обучения относятся, в частности:

- nn — библиотека моделирования нейронных сетей;
- nngraph — графическое представление нейронных сетей, моделируемых с помощью nn;
- dp — библиотека по глубокому машинному обучению;
- dpnn — «глубокие» расширения для библиотеки nn;
- rnn — рекуррентные нейронные сети;
- optim — методы оптимизации;
- unsup — пакет поддержки методов обучения без учителя (модули совместимы с nn);
- rbm_toolbox — ограниченные машины Больцмана;
- svm — метод опорных векторов;
- OpenGM — обертка Lua над одноименной библиотекой моделирования графов, реализованной на C++;
- LuaSHKit — обертка Lua над библиотекой локально-чувствительного хеширования SHKit;

- kernel smoothing — функции сглаживания;
- component analysis — анализ главных компонент (PCA, Whitened PCA, LDA, LPP, NPP, FastICA);
- reinforcement learning — обучение с подкреплением. Реализует примеры [42].

[Waffles](#) [69] — набор инструментов, предназначенных для решения задач машинного обучения и анализа данных с интерфейсом командной строки. Предлагаемые инструменты реализованы как «обертки» над библиотекой классов на языке C++, так что Waffles можно использовать и как библиотеку. Цель Waffles, по словам разработчиков, заключается в предоставлении пользователю максимально простого инструмента для использования в автоматизированных экспериментах. Waffles автоматически компенсирует недостающие значения, применяет необходимые фильтры для преобразования данных к требуемому типу. Многие пакеты для машинного обучения обеспечивают сходные возможности, но требуют от пользователя явным образом настроить фильтры данных и преобразования типов, чтобы сделать данные совместимыми с определенным алгоритмом обучения.

[Weka](#) (Waikato Environment for Knowledge Analysis) [70, 71] — коллекция программ для машинного обучения и анализа данных, реализованных на Java в Университете Вайкато (Новая Зеландия). Weka поддерживает методы решения стандартных задач анализа данных, таких как предварительная обработка данных, кластеризация, классификация, регрессия и снижение размерности. Пакет RWeka реализует интерфейс между Weka и R, а `python-weka-wrapper` — Python «обертку» для Weka.

2.2. Программы

Ниже рассмотрены программы для машинного обучения, которые также можно использовать в качестве библиотек.

[GNU Octave](#) [72] представляет собой язык программирования для научных и инженерных расчетов, близкий к MATLAB, а также среду для выполнения расчетов. Возможности GNU Octave в области машинного обучения обусловлены поддержкой библиотек LIBSVM, LIBLINEAR и пакета Shogun. Полный список поддерживаемых пакетов расширений находится по [адресу](#) [73]. Интерфейс между Octave и R реализован в пакете RcppOctave.

[Orange](#) [74] позволяет решать задачи машинного обучения и анализа данных средствами визуального программирования (рис. 3). Orange написан на Python, использует научные библиотеки этого языка (NumPy [49], SciPy [65]), а также библиотеку scikit-learn [64], которая и определяет возможности программы в области машинного обучения.

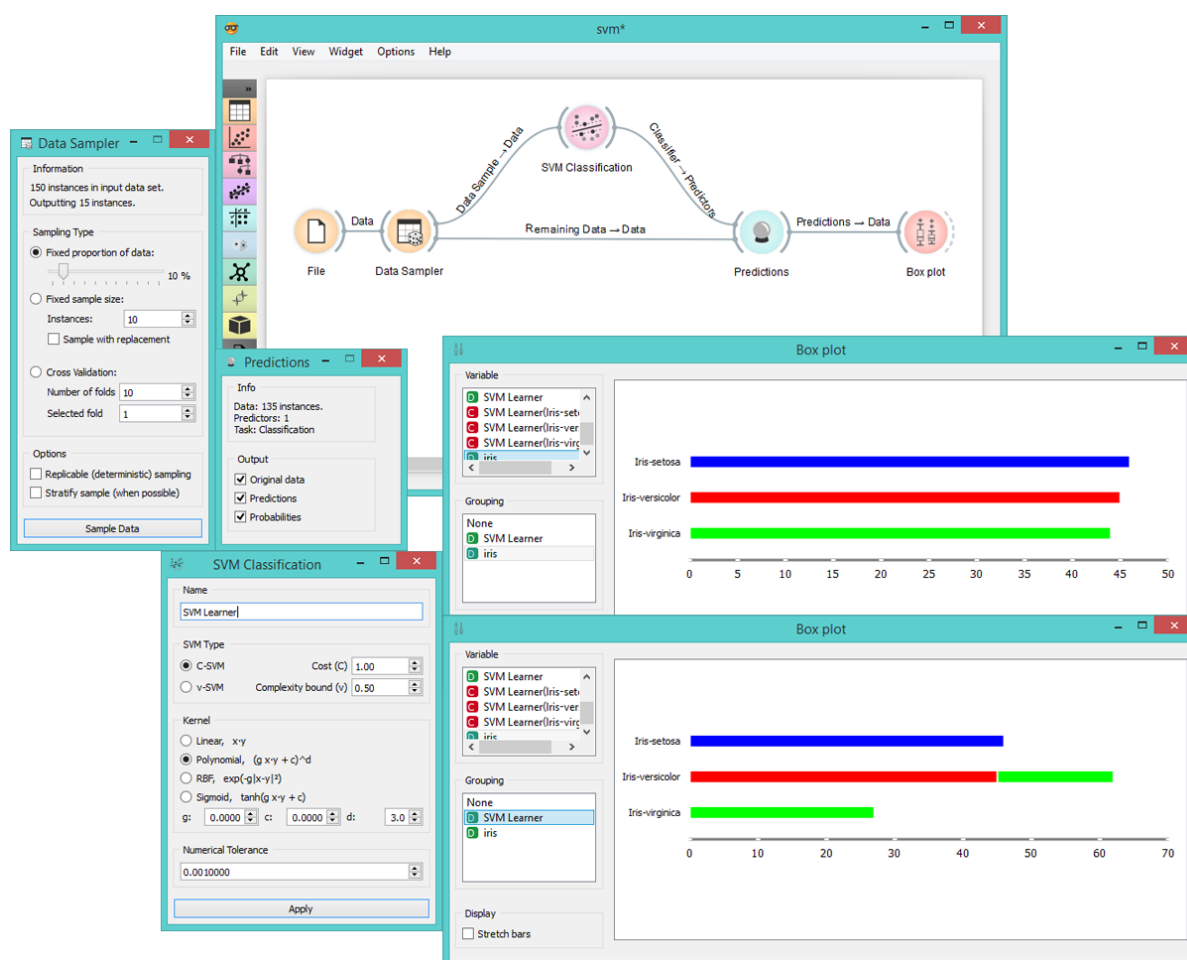


Рисунок 3 — Построение классификатора, основанного на методе опорных векторов в Orange [74]

[R](#) [75] — язык программирования и среда для анализа данных. Официальная документация к R и дополнительные пакеты распространяются через сеть ftp- и web-серверов, называемую CRAN (Comprehensive R Archive Network) [76]. В настоящее время R имеет более 7000 пакетов расширений, позволяющих решить практически любую задачу из области статистики, причем, как правило, несколькими способами. Одних только пакетов, посвященных машинному обучению, на веб-странице [77] приведено более 70.

Язык R широко используется специалистами в области анализа данных и машинного обучения, и в настоящее время, согласно рейтингу [IEEE Spectrum](#) [78], является самым популярным в мире предметно-ориентированным языком программирования.

[RapidMiner](#) [79] — программная платформа, предоставляющая интегрированную среду для решения задач машинного обучения, интеллектуального анализа данных, анализа текстов и бизнес-аналитики. Основное ядро программы распространяется свободно с открытым исходным кодом. В то же время существует несколько расширенных версий программы, распространяющихся на коммерческой основе. RapidMiner поддерживает все этапы работы с данными — от предварительной их обработки до визуализации результатов и имеет графический интерфейс для визуальной компоновки процесса обработки данных (см. рис. 4).

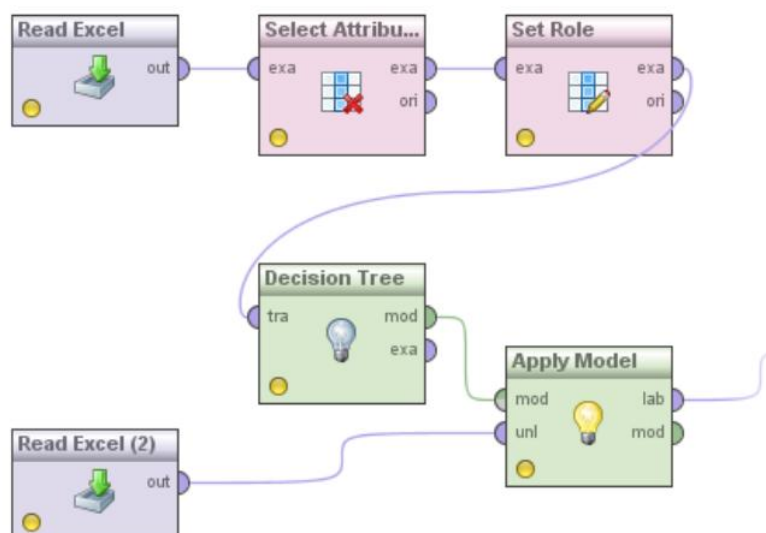


Рисунок 4 — Визуальная компоновка процесса обработки данных в RapidMiner [79]

Программная платформа RapidMiner написана на Java и способна использовать алгоритмы библиотеки Weka, а также обмениваться данными с пакетом R. Среди реализованных в RapidMiner методов:

- метод k-ближайших соседей;
- наивный байесовский классификатор;
- решающие деревья;
- random forest;
- нейронные сети;
- перцептрон;
- линейная и полиномиальная регрессии;
- гауссовские процессы;
- метод опорных векторов;
- метод k-средних.

По состоянию на 2014 г. RapidMiner является одной из наиболее популярных программ в области машинного обучения и анализа данных, имея свыше 250 000 зарегистрированных пользователей [80].

2.3. Библиотеки для реализации масштабируемых алгоритмов машинного обучения

[Apache Mahout](#) [81] — библиотека масштабируемых алгоритмов машинного обучения, разрабатываемая Apache Software Foundation. Библиотека, в первую очередь, ориентирована на решение задач коллаборативной фильтрации, кластеризации и классификации. В настоящее время большинство алгоритмов Mahout реализованы на базе Hadoop [46] с использованием модели [MapReduce](#) [82], однако в будущем предполагается не ограничиваться возможностями Hadoop. Помимо этого, Mahout содержит Java-библиотеки для решения задач линейной алгебры и статистики.

[Jubatus](#) [83] — фреймворк и библиотека для распределенного машинного обучения. Jubatus состоит из следующих компонентов:

- онлайн-библиотека для решения задач машинного обучения: классификации, регрессии, кластеризации, выработки рекомендаций, анализу графов, обнаружению аномалий;
- конвертер вектора признаков (feature vector converter): предварительная обработка данных и выделение признаков;
- фреймворк для распределенного отказоустойчивого онлайн-машинного обучения.

У Jubatus много общего с проектами Hadoop [46] и Mahout [81] — все три представляют собой масштабируемое ПО, способное работать на стандартном оборудовании. Однако Hadoop не содержит многих современных алгоритмов машинного обучения, а основанный на Hadoop проект Mahout не имеет алгоритмов для онлайн-обработки потоков данных. Последнее как раз и является сильной стороной Jubatus.

Возможен вызов Jubatus из Python при помощи одноименного пакета, а также из C++, Java и Ruby. Основная документация к Jubatus — на японском языке.

[MLlib](#) [84] — фреймворк для распределенного решения задач машинного обучения, основанный на Spark [47]. Последний представляет собой фреймворк, предназначенный для создания распределенных приложений. В отличие от Hadoop, реализующего двухуровневую концепцию MapReduce с дисковым хранилищем, Spark использует примитивы в оперативной памяти, благодаря чему позволяет получать многократный выигрыш в скорости работы в определенных приложениях. Spark предоставляет интерфейс из более чем 80 высокоуровневых операторов, позволяющий создавать параллельные приложения на Scala, Python и R.

[Vowpal Wabbit](#) [85] — библиотека для создания масштабируемых алгоритмов машинного обучения, реализованная на C++ с помощью библиотеки Boost. Существует пакет-«обертка» этой библиотеки для языка Python — Wabbit Wappa.

3. Сравнительный анализ ПО с учетом особенностей реализации программного комплекса автоматизированной количественной оценки метакомпетенций учащихся

В ходе информационного поиска проанализировано более 40 единиц ПО (библиотеки, фреймворки, программы и т. п.), не считая пакетов расширений языков R и Python, а также библиотек для реализации высокопроизводительных алгоритмов.

Представляется, что последний из указанных видов ПО, равно как и специализированные библиотеки, имеет смысл использовать только в том случае, когда необходимо повысить производительность какой-либо части системы. К тому же, специализированные библиотеки, как правило, уже используются в более высокоуровневом универсальном ПО. Поэтому в дальнейшем сосредоточимся на анализе последнего.

Универсальное ПО включает в себя большую группу инструментов, реализованных на C++ и предполагающих решение задач машинного обучения с помощью этого языка программирования. Однако C++ достаточно сложен в освоении и при этом его возможные преимущества (например, более высокая производительность разработанных на нем программ) на этапе создания экспериментального образца программного комплекса не существенны. Использование языка Julia представляется весьма перспективным в силу достаточной простоты его освоения и высокой производительности, однако до реализации значительной части методов машинного обучения на нем еще далеко. Самая большая группа универсального ПО состоит из инструментов, которые предоставляют интерфейсы к большому числу других языков программирования (табл. 1).

Таблица 1 — Поддержка языками программирования интерфейсов с пакетами машинного обучения

Язык	Число интерфейсов со сторонними пакетами машинного обучения
C++	11
C#	3
GNU Octave	4
Go	1
Java	10
Julia	1
Lua	2
Python	10
R	11
Ruby	1

Среди этого ПО особо выделяются языки Java, Python и R как реализующие многие методы машинного обучения собственными средствами и позволяющие использовать возможности значительного числа стороннего ПО. Два последних языка — Python и R — более просты в освоении, имеют интерфейсы к большому числу сторонних библиотек, хорошо дополняют друг друга по отношению к решению задач машинного обучения, способны вызывать программы друг друга.

Рассмотрим Python и R более подробно. Первый из них обладает обширной и хорошо документированной библиотекой `scikit-learn`, второй — богатой коллекцией специализированных библиотек для машинного обучения.

Язык R создавался специально для статистического анализа, поэтому он имеет некоторые специфические типы данных, удобные для выполнения такого анализа, например, таблицы (`data frames`). Основным типом данных в R является вектор. Операции с векторами и матрицами выполняются в R поэлементно сразу над всем набором данных (будь то вектор, матрица, таблица и т. п.). Такие векторизованные вычисления весьма эффективны, поэтому если нужно совершить действия над элементами вектора (матрицы и т.д.) предпочтительнее использовать векторизацию, а не циклы. Базовые возможности R описаны в книге [86].

Python представляет собой язык программирования общего назначения. Для его полноценного использования в задачах машинного обучения необходимо установить пакеты:

- NumPy и SciPy — для операций с матрицами, линейной алгебры и базовых функций статистики, позволяющие совершать векторизованные вычисления, как в R или Matlab;

- [pandas](#) — для операций таблицами (data frames);
- [statsmodels](#) — дополнение к SciPy для выполнения статистических расчетов;
- [matplotlib](#) — для визуализации результатов.

Строго говоря, возможности pandas в части операций с таблицами существенно шире базовых возможностей R, поэтому в последнем нужно дополнительно установить пакет [dplyr](#). Пакет pandas подробно описан в [87].

Возможности Python в части визуализации даже после установки matplotlib остаются существенно ниже возможностей R.

Краткий список пакетов R, полезных при решении задач машинного обучения, приведен ниже.

- [activelearning](#) — активное обучение.
- [caret](#) — подборка функций для обучения и построения классификационных и регрессионных моделей.
- [class](#) — методы классификации, включая метод k-ближайших соседей и карты Кохонена.
- [cluster](#) — методы кластеризации.
- [CORElearn](#) — классификация, регрессия и оценивание. Пакет представляет собой коллекцию функций, реализованных на C++ и снабженных интерфейсом с R. Некоторые алгоритмы имеют параллельные реализации.
- [e1071](#) — большая коллекция функций для решения задач классификации и кластеризации, в частности, метод опорных векторов.
- [glmnet](#) — обобщенная линейная модель, в частности, линейная и логистическая регрессии.
- [kernlab](#) — методы машинного обучения для классификации, регрессии, кластеризации и др., в частности, метод опорных векторов.
- [klaR](#) — функции для классификации и визуализации.
- [kohonen](#) — обучение с учителем и обучение без учителя сетей Кохонена ([пример использования](#)). Другой подобный пакет: [som](#).
- [MDPtoolbox](#) (Markov Decision Processes toolbox) — обучение с подкреплением.
- [nnet](#) — реализация, обучение и визуализация нейронных сетей.
- [neuralnet](#) — обучение нейронных сетей.
- [NeuralNetTools](#) — моделирование и визуализация нейронных сетей.
- [Rmixmod](#) — обучение с учителем, без учителя и с частичным привлечением учителя. Имеется интерфейс к библиотеке [Mixmod](#).

- [RWeka](#) — интерфейс с Weka. Среди прочего, реализует методы многовариантного обучения.

- [upclass](#) — классификация с частичным привлечением учителя.

В CRAN существует тематический указатель пакетов — [CRAN Task Views](#) [88]. Причем, хотя в разделе «Machine Learning & Statistical Learning» [77] приведено более 70 пакетов, этим возможности R в части машинного обучения не ограничиваются: некоторая часть пакетов отнесена к другим темам, например, к «Cluster Analysis & Finite Mixture Models», а часть методов входит в базовые возможности R. Поиск нужных пакетов в CRAN удобно выполнять с помощью сервиса [Rdocumentation](#) [89].

Одним из преимуществ R является наличие в нем пакета [Shiny](#) [90], который позволяет создавать на основе кода R интерактивные веб-страницы. Предоставляемый компанией RStudio сервер [Shiny Server](#) [91] используется для размещения и обслуживания приложений Shiny. Shiny Server поставляется в двух версиях: свободный сервер с открытым исходным кодом; платный профессиональный сервер с более широким набором функций.

Средствами R можно реализовать все методы машинного обучения, которые предполагается использовать в рассматриваемой нами задаче (причем обычно — несколькими способами), кроме спайковых нейронных сетей. Причины этого неясны, поскольку для моделирования других видов нейронных сетей существует большое число пакетов. Модели единичных нейронов, например, модель Ходжкина-Хаксли в R также [реализованы](#) [92].

В Python кроме scikit-learn существует ряд следующих пакетов, посвященных задачам машинного обучения:

- [Brian](#) — моделирование спайковых нейронных сетей;

- [DEAP](#) (Distributed Evolutionary Algorithms in Python) — распределенные версии эволюционных алгоритмов;

- [MISVM](#) — реализации различных вариантов методов опорных векторов применительно к многовариантному обучению;

- [PyBrain](#) — содержит алгоритмы для нейронных сетей, обучения с подкреплением, обучения без учителя, а также ряд эволюционных алгоритмов;

- [Pyevolve](#) — реализация генетических алгоритмов на чистом Python;

- [PyMTL](#) (Python library for Multi-task learning) — модуль для мультизадачного обучения, построенный на базе scikit-learn и SciPy / NumPy;

- [PyMVPA](#) (MultiVariate Pattern Analysis (MVPA) in Python) — пакет, призванный объединить возможности scikit-learn, Shogun и др. для решения задач статистического обучения на больших массивах данных. Среди прочего, реализует сети Кохонена;

- [PyNN](#) — пакет для реализации нейронных сетей.

О других пакетах моделирования нейронных сетей можно узнать из обзора [93]. Обновляемый список пакетов, посвященных искусственному интеллекту и, в частности, машинному обучению, можно найти на веб-странице [94].

Python позволяет, по-видимому, реализовать все методы машинного обучения, которые предполагается использовать в рассматриваемой нами задаче. В R одну и ту же зада-

чу, как правило, позволяет решить несколько пакетов, тогда как в Python доминирует один пакет — scikit-learn. В то же время, в Python есть уникальные специализированные пакеты, например, Brian, посвященный спайковым нейронным сетям.

Оба языка обладают обширными сообществами пользователей. Однако доля пользователей R, занимающихся вопросами машинного обучения, гораздо больше, чем в Python вследствие специализации языка R. Так, веб-страниц, посвященных реализации методов машинного обучения на R в десятки раз больше, чем страниц, посвященных решению тех же задач на Python. Следует отметить, что в последние несколько лет (примерно с 2010 года) возможности Python-а в области машинного обучения существенно возросли. Однако до сих пор Python выступает в роли догоняющего, повторяя, за редкими исключениями, существующие в R решения.

Все вышеизложенное приводит к выводу, что выбор R или Python в качестве инструмента для реализации машинного обучения, если только речь не идет о какой-то узко-специальной задаче, диктуется скорее привычками и подготовкой конкретного специалиста, чем достоинствами/недостатками этих языков.

Отметим, что программы на Python вызываются из R [95] и наоборот [96]. Обмен информацией между Python и R подробно описан в [97].

Заключение

Проведен обзор свободно распространяемого ПО, реализующего методы машинного обучения, которые предполагается использовать при разработке комплекса научно-технических решений, направленных на создание системы количественной оценки метапредметных, метакогнитивных и метакреативных компетенций учащихся.

Выделено специализированное ПО, реализующее один, или несколько методов машинного обучения, и универсальное ПО для решения группы задач машинного обучения.

Проведен сравнительный анализ рассмотренного ПО с точки зрения целесообразности его использования в экспериментальном образце программного комплекса автоматизированной количественной оценки метапредметных, метакогнитивных и метакреативных навыков и умений учащихся на основе анализа поведения пользователей в различных обучающих средах.

По результатам анализа в качестве наиболее подходящих инструментов выделены языки R и Python, с одной стороны, реализующие многие методы машинного обучения собственными средствами, а с другой стороны, позволяющие использовать возможности значительного числа стороннего ПО. Для повышения производительности к ним могут быть подключены специализированные библиотеки, более эффективно реализующие тот или иной алгоритм, в частности, библиотеки масштабируемых алгоритмов машинного обучения.

Проанализированная информация о рассмотренном ПО представлена в структурированном виде, предполагающем возможность ее использования при решении других задач,

требующих применения свободно распространяемых программных средств, реализующих различные методы машинного обучения.

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации (Соглашение о предоставлении субсидии № 14.577.21.0135, уникальный идентификатор прикладных научных исследований (проекта) RFMEFI57714X0135).

Список литературы

1. Печников В.П., Добряков А.А. Тренинг и оценка качества креативных способностей с помощью ЭВМ // Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 01. С. 221–231. DOI: [10.7463/0115.0754865](https://doi.org/10.7463/0115.0754865)
2. Гаврилина Е.А., Захаров М.А., Карпенко А.П., Смирнова Е.В. Онтологический подход к тестированию уровня владения обучающимся метапредметными понятиями // Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 02. С. 136–149. DOI: [10.7463/0215.0756631](https://doi.org/10.7463/0215.0756631)
3. Цветков Ю.Б. Особенности проектирования учебных целей дисциплин инженерных образовательных программ // Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 03. С. 331–344. DOI: [10.7463/0315.0761285](https://doi.org/10.7463/0315.0761285)
4. Гаврилина Е.А., Захаров М.А., Карпенко А.П. Количественная оценка метакомпетенций учащихся на основе методов машинного обучения // Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 04. С. 228–253. DOI: [10.7463/0415.0764221](https://doi.org/10.7463/0415.0764221)
5. Coelho L.P., Richert W. Building Machine Learning Systems with Python. Packt Publishing, 2015. 302 p.
6. Machine learning // Wikipedia, the free encyclopedia: internet-portal. Available at: https://en.wikipedia.org/wiki/Machine_learning , accessed 02.10.2015.
7. Software // Machine Learning Open Source Software. Available at: <http://www.mloss.org/software/> , accessed 02.10.2015.
8. Misiti J. Awesome Machine Learning // GitHub: public repository. Available at: <https://github.com/josephmisiti/awesome-machine-learning> , accessed 02.10.2015.
9. Machine Learning Libraries // Machine Learning Newsletter: website. Available at: <http://www.mln.io/resources/periodic-table/> , accessed 02.10.2015.
10. gretl // Gnu Regression, Econometrics and Time-series Library : official website. Available at: <http://gretl.sourceforge.net/> , accessed 05.10.2015.
11. libnabo // GitHub: public repository. Available at: <https://github.com/ethz-asl/libnabo> , accessed 05.10.2015.
12. Eigen // Eigen C++ template library: official website. Available at: <http://eigen.tuxfamily.org/> , accessed 05.10.2015.
13. Boost // Boost C++ Libraries: official website. Available at: <http://www.boost.org/> , accessed 05.10.2015.

14. LIBSVM // LIBSVM -- A Library for Support Vector Machines: official website. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm> , accessed 06.10.2015.
15. LIBLINEAR // LIBLINEAR -- A Library for Large Linear Classification: official website. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm> , accessed 06.10.2015.
16. LIBSVM Tools // LIBSVM Tools: official website. Available at: <http://ntucsu.csie.ntu.edu.tw/~cjlin/libsvmtools/> , accessed 06.10.2015.
17. GPU-accelerated LIBSVM // Multimedia Knowledge and Social Media Analytics Laboratory: official website. Available at: <http://mklab.itl.gr/project/GPU-LIBSVM> , accessed 06.10.2015.
18. Mixmod // MIXMOD High Performance Model-Based Cluster and Discriminant Analysis: official website. Available at: <http://www.mixmod.org/?lang=en> , accessed 06.10.2015.
19. Mixmod Library // MIXMOD High Performance Model-Based Cluster and Discriminant Analysis: official website. Available at: http://www.mixmod.org/rubrique.php3?id_rubrique=24&lang=en , accessed 06.10.2015.
20. mlpack // mlpack : official website. Available at: <http://mlpack.org/> , accessed 06.10.2015.
21. Armadillo // Armadillo C++ linear algebra library: official website. Available at: <http://arma.sourceforge.net/> , accessed 06.10.2015.
22. Curtin R.R., Cline J.R., Slagle N.P., March W.B., Ram P., Mehta N.A., Gray A.G. MLPACK: A Scalable C++ Machine Learning Library // Journal of Machine Learning Research. 2013. Vol. 14. P. 801–805.
23. Machine Learning with Javascript // Ganesh Iyer's Blog. Available at: <http://ganeshiyer.net/blog/2013/10/22/machine-learning-with-javascript/> , accessed 06.10.2015.
24. Lyric: Linear Regression library in pure Javascript // GitHub: public repository. Available at: <https://github.com/flurry/Lyric> , accessed 06.10.2015.
25. ConvNetJS: Deep Learning in your browser // ConvNetJS: official website. Available at: <http://cs.stanford.edu/people/karpathy/convnetjs/> , accessed 06.10.2015.
26. The javascript architecture-free neural network library for node.js and the browser // Synaptic – The JavaScript neural network library: official website. Available at: <http://synaptic.juancazala.com/> , accessed 06.10.2015.
27. Node.js // Node.js: official website. Available at: <https://nodejs.org/en> / , accessed 06.10.2015.
28. Spiking neural network // Wikipedia, the free encyclopedia: internet-portal. Available at: https://en.wikipedia.org/wiki/Spiking_neural_network , accessed 06.10.2015.
29. GENeral NEural SIMulation System // GENESIS-Sim.Org: official website. Available at: <http://genesis-sim.org/> , accessed 06.10.2015.
30. NEURON // NEURON: for empirically-based simulations of neurons and networks of neurons: official website. Available at: <http://www.neuron.yale.edu/neuron> / , accessed 06.10.2015.

31. Brian // The Brian spiking neural network simulator: official website. Available at: <http://briansimulator.org/> , accessed 06.10.2015.
32. Paugam-Moisy H., Bohte S. Computing with Spiking Neuron Networks // Website Sander Bohte. Available at: http://homepages.cwi.nl/~sbohte/publication/paugam_moisy_bohte_SNNChapter.pdf , accessed 06.10.2015.
33. Brette R., et al. Simulation of networks of spiking neurons: A review of tools and strategies // PMC: US National Library of Medicine National Institutes of Health: website. Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2638500/> , accessed 06.10.2015.
34. NEST // NEST: The Neural Simulation Tool: official website. Available at: <http://www.nest-simulator.org/> , accessed 06.10.2015.
35. Nengo // The Nengo Neural Simulator: official website. Available at: <http://nengo.ca/> , accessed 06.10.2015.
36. Neural Engineering Framework // CNRGLab @ UWaterloo | Research: official website. Available at: <http://compneuro.uwaterloo.ca/research/nef.html> , accessed 06.10.2015.
37. Eliasmith C., Anderson Ch.H. Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems. MIT Press, 2003. 377 p.
38. Eliasmith C. How to Build a Brain: A Neural Architecture for Biological Cognition. Oxford University Press, 2013. 480 p.
39. Neuroph // Neuroph Java Neural Network Framework: official website. Available at: <http://neuroph.sourceforge.net/> , accessed 06.10.2015.
40. OpenCV // OpenCV: Open Source Computer Vision: official website. Available at: <http://opencv.org/> , accessed 06.10.2015.
41. ROS // ROS.Org. Robot Operating System: official website. Available at: <http://wiki.ros.org/> , accessed 06.10.2015.
42. Саттон Р.С., Барто Э.Г. Обучение с подкреплением: пер. с англ. М.: Бином. Лаборатория знаний, 1999. 400 с.
43. Accord.NET Framework // Accord.NET Machine Learning Framework: official website. Available at: <http://accord-framework.net/> , accessed 06.10.2015.
44. Datumbox Machine Learning Framework // Datumbox: official website. Available at: <http://www.datumbox.com/machine-learning-framework/> , accessed 06.10.2015.
45. Deeplearning4j – Open-source, distributed deep learning for the JVM // DL4J Deep Learning for Java: official website. Available at: <http://deeplearning4j.org/> , accessed 07.10.2015.
46. Welcome to Apache™ Hadoop®! // Hadoop: official website. Available at: <https://hadoop.apache.org/> , accessed 07.10.2015.
47. Apache Spark™ - Lightning-fast cluster computing // Apache Spark: official website. Available at: <http://spark.apache.org> , accessed 07.10.2015.
48. ND4J: N-Dimensional Arrays for Java // ND4J: official website. Available at: <http://nd4j.org/> , accessed 07.10.2015.

49. NumPy // NumPy: official website. Available at: <http://www.numpy.org/> , accessed 07.10.2015.
50. Dlib // Dlib C++ Library: official website. Available at: <http://dlib.net/> , accessed 07.10.2015.
51. King D.E. Dlib-ml: A Machine Learning Toolkit // Journal of Machine Learning Research. 2003. Vol. 10. P. 1755–1758.
52. Encog Machine Learning Framework // Heaton Research: official website. Available at: <http://www.heatonresearch.com/encog/> , accessed 07.10.2015.
53. Machine Learning for Go // GitHub: public repository. Available at: <https://github.com/sjwhitworth/golearn> , accessed 07.10.2015.
54. H2O.ai - Why H2O? // H2O.ai – Fast Scalable Machine Learning: official website. Available at: <http://h2o.ai/product/> , accessed 07.10.2015.
55. Intel® Data Analytics Acceleration Library // Intel® Developer Zone: official website. Available at: <https://software.intel.com/en-us/forums/intel-data-analytics-acceleration-library> , accessed 07.10.2015.
56. Intel® Math Kernel Library (Intel® MKL) // Intel® Developer Zone: official website. Available at: <https://software.intel.com/en-us/intel-mkl> , accessed 07.10.2015.
57. Java Statistical Analysis Tool, a Java library for Machine Learning // GitHub: public repository. Available at: <https://github.com/EdwardRaff/JSAT> , accessed 07.10.2015.
58. JuliaStats.org // JuliaStats : official website. Available at: <http://juliastats.github.io/> , accessed 07.10.2015.
59. The Julia Language // Julia: official website. Available at: <http://julialang.org/> , accessed 07.10.2015.
60. Embedded R within Julia // GitHub: public repository. Available at: <https://github.com/JuliaStats/RCall.jl> , accessed 07.10.2015.
61. R package which integrating R and Julia // GitHub: public repository. Available at: <https://github.com/armgong/RJulia> , accessed 07.10.2015.
62. MOA Massive Online Analysis // Real Time Analytics for Data Streams: official website. Available at: <http://moa.cms.waikato.ac.nz/> , accessed 07.10.2015.
63. OpenNN // Artnikis - Predictive Analytics, Deep Learning, Data Mining, Big Data: official website. Available at: <http://www.artelnics.com/opennn> , accessed 07.10.2015.
64. scikit-learn: machine learning in python // scikit-learn: official website. Available at: <http://scikit-learn.org/stable/> , accessed 07.10.2015.
65. SciPy.org // SciPy.org: official website. Available at: <http://www.scipy.org/> , accessed 07.10.2015.
66. Shogun // The SHOGUN Machine Learning Toolbox: official website. Available at: <http://www.shogun-toolbox.org/> , accessed 07.10.2015.
67. Torch // Scientific Computing for LuaJIT: official website. Available at: <http://torch.ch/> , accessed 07.10.2015.

68. Collobert R., Kavukcuoglu K., Farabet C. Torch7: A Matlab-like Environment for Machine Learning // Ronan Collobert: homepage. Available at: http://ronan.collobert.com/pub/matos/2011_torch7_nipsw.pdf , accessed 07.10.2015.
69. Waffles. A Machine Learning Toolkit: official website. Available at: <http://waffles.sourceforge.net/> , accessed 07.10.2015.
70. Weka3 - Data Mining with Open Source Machine Learning Software in Java // Weka: the University of Waikato: official website. Available at: <http://www.cs.waikato.ac.nz/~ml/weka/> , accessed 07.10.2015.
71. Hall M., Witten I., Frank E. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, 2011. 629 p.
72. GNU Octave // GNU Octave: official website. Available at: <http://www.gnu.org/software/octave/> , accessed 07.10.2015.
73. Packages Octave // GNU Octave Wiki: official website. Available at: <http://wiki.octave.org/Packages> , accessed 07.10.2015.
74. Orange Data Mining // Orange: official website. Available at: <http://orange.biolab.si/> , accessed 07.10.2015.
75. R: The R Project for Statistical Computing // The R Project: official website. Available at: <https://www.r-project.org/> , accessed 07.10.2015.
76. CRAN - Mirrors // The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/mirrors.html> , accessed 07.10.2015.
77. CRAN Task View: Machine Learning & Statistical Learning // The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/web/views/MachineLearning.html> , accessed 07.10.2015.
78. Cass S. The 2015 Top Ten Programming Languages. Posted 20.07.2015 // IEEE Spectrum: website. Available at: <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages> , accessed 15.10.2015.
79. RapidMiner // RapidMiner - #1 Open Source Predictive Analytics Platform: official website. Available at: <https://rapidminer.com/products/studio/> , accessed 07.10.2015.
80. RapidMiner Press Releases, Boston - June 17, 2014 // RapidMiner - #1 Open Source Predictive Analytics Platform: official website. Available at: <https://rapidminer.com/news-posts/rapidminer-acquires-radoop/> , accessed 07.10.2015.
81. Apache Mahout: Scalable Machine Learning and Data Mining // Apache Mahout™ : official website. Available at: <http://mahout.apache.org/> , accessed 08.10.2015.
82. MapReduce // Wikipedia, the free encyclopedia: internet-portal. Available at: <https://ru.wikipedia.org/wiki/MapReduce> , accessed 08.10.2015.
83. Jubatus // Jubatus : Distributed Online Machine Learning Framework: official website. Available at: <http://jubat.us/en/> , accessed 08.10.2015.
84. MLib | Apache Spark // Apache Spark: official website. Available at: <http://spark.apache.org/mlib/> , accessed 08.10.2015.

85. Vowpal Wabbit // GitHub: public repository. Available at:
https://github.com/JohnLangford/vowpal_wabbit/wiki , accessed 08.10.2015.
86. Кабаков Р.И. R в действии. Анализ и визуализация данных в программе R. М.: ДМК Пресс, 2014. 588 с.
87. Маккинни У. Python и анализ данных: пер. с англ. М.: ДМК Пресс, 2015. 482 с.
88. CRAN Task Views // The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/web/views/> , accessed 08.10.2015.
89. Search all R packages and function manuals // Rdocumentation: website. Available at: <http://www.rdocumentation.org/> , accessed 08.10.2015.
90. shiny: Web Application Framework for R // The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/web/packages/shiny/index.html> , accessed 15.10.2015.
91. RStudio - Shiny Server // RStudio: official website. Available at:
<https://www.rstudio.com/products/shiny/shiny-server/> , accessed 15.10.2015.
92. Hodgkin-Huxley model in R // mages' blog. Available at:
<http://www.magesblog.com/2012/06/hodgkin-huxley-model-in-r.html> , accessed 08.10.2015.
93. What is the best neural network library for Python? // Quora: internet-portal. Available at:
<https://www.quora.com/What-is-the-best-neural-network-library-for-Python> , accessed 08.10.2015.
94. Python For Artificial Intelligence // Python Wiki: internet-portal. Available at:
<https://wiki.python.org/moin/PythonForArtificialIntelligence> , accessed 08.10.2015.
95. Calling Python from R with rPython // R-bloggers: internet-portal. Available at:
<http://www.r-bloggers.com/calling-python-from-r-with-rpython/> , accessed 08.10.2015.
96. Accessing R from Python using RPy2 // R-bloggers: internet-portal. Available at:
<http://www.r-bloggers.com/accessing-r-from-python-using-rpy2/> , accessed 08.10.2015.
97. R/SPlus - Python Interface // The Omega Project for Statistical Computing: official website
Available at: <http://www.omegahat.org/RSPython/> , accessed 08.10.2015.

Software Tools for Automated Quantitative Assessment of the Students' Meta-competence. Overview

10, October 2015

DOI: 10.7463/1015.0821623

P.P. Belonozhko^{1,*}, V.V. Belous¹,
A.P. Karpenko¹, D.A. Khramov²

UDK: 519.6

¹Moscow State Technical University n.a. N.E. Bauman,
Moscow, Russia

²Institute of Technical Mechanics of the National Academy of Sciences of Ukraine
and National Space Agency of Ukraine,
Dnepropetrovsk, Ukraine

[*byelonozhko@mail.ru](mailto:byelonozhko@mail.ru)

The paper concerns a problem of elaborating a set of the scientific and technical solutions aimed at creating a system of automated quantitative assessment of students' meta-subject, meta-cognitive, and meta-creative skills and abilities (meta-competence). It provides overview of free software that implements computer-aided learning methods to be used. There is an emphasis on specialized and universal software. The paper gives recommendations on the choice of tools to implement a pilot sample of a software package. Materials reviewed may also be used to solve a wide range of theoretical and applied tasks regarding the need to implement various computer-aided learning methods.

References

1. Pechnikov V.P., Dobryakov A.A. Computer-Aided Training and Quality Assessment of Creative Abilities. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2015, no. 1, pp. 221–231. DOI: [10.7463/0115.0754865](https://doi.org/10.7463/0115.0754865) (in Russian).
2. Gavrilina E.A., Zakharov M.A., Karpenko A.P., Smirnova E.V. An Ontological Approach to Test Student's Competences in Meta-Subject Concepts. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2015, no. 2, pp. 136–149. DOI: [10.7463/0215.0756631](https://doi.org/10.7463/0215.0756631) (in Russian).

3. Tsvetkov Yu.B. Special Aspects of Learning Objectives Design for Disciplines in Engineering Education. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2015, no. 3, pp. 331–344. DOI: [10.7463/0315.0761285](https://doi.org/10.7463/0315.0761285) (in Russian).
4. Gavrilina E.A., Zakharov M.A., Karpenko A.P. A Quantitative Assessment of Student's Metacompetences Based on Machine Learning Methods. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2015, no. 4, pp. 228–253. DOI: [10.7463/0415.0764221](https://doi.org/10.7463/0415.0764221) (in Russian).
5. Coelho L.P., Richert W. *Building Machine Learning Systems with Python*. Packt Publishing, 2015. 302 p.
6. Machine learning. Wikipedia, the free encyclopedia: internet-portal. Available at: https://en.wikipedia.org/wiki/Machine_learning , accessed 02.10.2015.
7. Software. Machine Learning Open Source Software: internet-portal. Available at: <http://www.mloss.org/software/> , accessed 02.10.2015.
8. Misiti J. *Awesome Machine Learning*. GitHub: public repository. Available at: <https://github.com/josephmisiti/awesome-machine-learning> , accessed 02.10.2015.
9. Machine Learning Libraries. Machine Learning Newsletter: website. Available at: <http://www.mln.io/resources/periodic-table/> , accessed 02.10.2015.
10. gretl. Gnu Regression, Econometrics and Time-series Library : official website. Available at: <http://gretl.sourceforge.net/> , accessed 05.10.2015.
11. libnabo. GitHub: public repository. Available at: <https://github.com/ethz-asl/libnabo> , accessed 05.10.2015.
12. Eigen. Eigen C++ template library: official website. Available at: <http://eigen.tuxfamily.org/> , accessed 05.10.2015.
13. Boost. Boost C++ Libraries: official website. Available at: <http://www.boost.org/> , accessed 05.10.2015.
14. LIBSVM. LIBSVM -- A Library for Support Vector Machines: official website. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm> , accessed 06.10.2015.
15. LIBLINEAR. LIBLINEAR -- A Library for Large Linear Classification: official website. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm> , accessed 06.10.2015.
16. LIBSVM Tools. LIBSVM Tools: official website. Available at: <http://ntucsu.csie.ntu.edu.tw/~cjlin/libsvmtools/> , accessed 06.10.2015.
17. GPU-accelerated LIBSVM. Multimedia Knowledge and Social Media Analytics Laboratory: official website. Available at: <http://mklab.iti.gr/project/GPU-LIBSVM> , accessed 06.10.2015.
18. Mixmod. MIXMOD High Performance Model-Based Cluster and Discriminant Analysis: official website. Available at: <http://www.mixmod.org/?lang=en> , accessed 06.10.2015.
19. Mixmod Library. MIXMOD High Performance Model-Based Cluster and Discriminant Analysis: official website. Available at: http://www.mixmod.org/rubrique.php3?id_rubrique=24&lang=en , accessed 06.10.2015.

20. mlpack. mlpack : official website. Available at: <http://mlpack.org/> , accessed 06.10.2015.
21. Armadillo. Armadillo C++ linear algebra library: official website. Available at: <http://arma.sourceforge.net/> , accessed 06.10.2015.
22. Curtin R.R., Cline J.R., Slagle N.P., March W.B., Ram P., Mehta N.A., Gray A.G. MLPACK: A Scalable C++ Machine Learning Library. *Journal of Machine Learning Research*, 2013, vol. 14, pp. 801–805.
23. Machine Learning with Javascript. Ganesh Iyer's Blog. Available at: <http://ganeshiyer.net/blog/2013/10/22/machine-learning-with-javascript/> , accessed 06.10.2015.
24. Lyric: Linear Regression library in pure Javascript. GitHub: public repository. Available at: <https://github.com/flurry/Lyric> , accessed 06.10.2015.
25. ConvNetJS: Deep Learning in your browser. ConvNetJS: official website. Available at: <http://cs.stanford.edu/people/karpathy/convnetjs/> , accessed 06.10.2015.
26. The javascript architecture-free neural network library for node.js and the browser. Synaptic – The JavaScript neural network library: official website. Available at: <http://synaptic.juancazala.com/> , accessed 06.10.2015.
27. Node.js. Node.js: official website. Available at: <https://nodejs.org/en/> , accessed 06.10.2015.
28. Spiking neural network. Wikipedia, the free encyclopedia: internet-portal. Available at: https://en.wikipedia.org/wiki/Spiking_neural_network , accessed 06.10.2015.
29. GEneral NEural SIMulation System. GENESIS-Sim.Org: official website. Available at: <http://genesis-sim.org/> , accessed 06.10.2015.
30. NEURON. NEURON: for empirically-based simulations of neurons and networks of neurons: official website. Available at: <http://www.neuron.yale.edu/neuron> / , accessed 06.10.2015.
31. Brian. The Brian spiking neural network simulator: official website. Available at: <http://briansimulator.org/> , accessed 06.10.2015.
32. Paugam-Moisy H., Bohte S. Computing with Spiking Neuron Networks. Website Sander Bohte. Available at: http://homepages.cwi.nl/~sbohte/publication/paugam_moisy_bohte_SNNChapter.pdf , accessed 06.10.2015.
33. Brette R. et al. *Simulation of networks of spiking neurons: A review of tools and strategies*. PMC: US National Library of Medicine National Institutes of Health: website. Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2638500/> , accessed 06.10.2015.
34. NEST. NEST: The Neural Simulation Tool: official website. Available at: <http://www.nest-simulator.org/> , accessed 06.10.2015.
35. Nengo. The Nengo Neural Simulator: official website. Available at: <http://nengo.ca/> , accessed 06.10.2015.

36. Neural Engineering Framework. CNRGlub @ UWaterloo | Research: official website. Available at: <http://compneuro.uwaterloo.ca/research/nef.html> , accessed 06.10.2015.
37. Eliasmith C., Anderson Ch.H. *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. MIT Press, 2003. 377 p.
38. Eliasmith C. *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford University Press, 2013. 480 p.
39. Neuroph. Neuroph Java Neural Network Framework: official website. Available at: <http://neuroph.sourceforge.net/> , accessed 06.10.2015.
40. OpenCV. OpenCV: Open Source Computer Vision: official website. Available at: <http://opencv.org/> , accessed 06.10.2015.
41. ROS. ROS.Org. Robot Operating System: official website. Available at: <http://wiki.ros.org/> , accessed 06.10.2015.
42. Sutton R.S., Barto A.G. *Reinforcement Learning: An Introduction*. A Bradford Book. MIT Press, Cambridge, 2002. (Russ. ed.: Sutton R.S., Barto A.G. *Обучение с подкреплением*. Moscow, BINOM. Laboratoriya znaniy Publ., 2012. 399 p.).
43. Accord.NET Framework. Accord.NET Machine Learning Framework: official website. Available at: <http://accord-framework.net/> , accessed 06.10.2015.
44. Datumbox Machine Learning Framework. Datumbox: official website. Available at: <http://www.datumbox.com/machine-learning-framework/> , accessed 06.10.2015.
45. Deeplearning4j – Open-source, distributed deep learning for the JVM. DL4J Deep Learning for Java: official website. Available at: <http://deeplearning4j.org/> , accessed 07.10.2015.
46. Welcome to Apache™ Hadoop®!. Hadoop: official website. Available at: <https://hadoop.apache.org/> , accessed 07.10.2015.
47. Apache Spark™ - Lightning-fast cluster computing. Apache Spark: official website. Available at: <http://spark.apache.org> , accessed 07.10.2015.
48. ND4J: N-Dimensional Arrays for Java. ND4J: official website. Available at: <http://nd4j.org/> , accessed 07.10.2015.
49. NumPy. NumPy: official website. Available at: <http://www.numpy.org> / , accessed 07.10.2015.
50. Dlib. Dlib C++ Library: official website. Available at: <http://dlib.net/> , accessed 07.10.2015.
51. King D.E. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 2003, vol. 10, pp. 1755–1758.
52. Encog Machine Learning Framework. Heaton Research: official website. Available at: <http://www.heatonresearch.com/encog/> , accessed 07.10.2015.
53. Machine Learning for Go. GitHub: public repository. Available at: <https://github.com/sjwhitworth/golearn> , accessed 07.10.2015.
54. H2O.ai - Why H2O?. H2O.ai – Fast Scalable Machine Learning: official website. Available at: <http://h2o.ai/product/> , accessed 07.10.2015.

55. Intel® Data Analytics Acceleration Library. Intel® Developer Zone: official website. Available at: <https://software.intel.com/en-us/forums/intel-data-analytics-acceleration-library> , accessed 07.10.2015.
56. Intel® Math Kernel Library (Intel® MKL). Intel® Developer Zone: official website. Available at: <https://software.intel.com/en-us/intel-mkl> , accessed 07.10.2015.
57. Java Statistical Analysis Tool, a Java library for Machine Learning. GitHub: public repository. Available at: <https://github.com/EdwardRaff/JSAT> , accessed 07.10.2015.
58. JuliaStats.org. JuliaStats: official website. Available at: <http://juliastats.github.io/> , accessed 07.10.2015.
59. The Julia Language. Julia: official website. Available at: <http://julialang.org/> , accessed 07.10.2015.
60. Embedded R within Julia. GitHub: public repository. Available at: <https://github.com/JuliaStats/RCall.jl> , accessed 07.10.2015.
61. R package which integrating R and Julia. GitHub: public repository. Available at: <https://github.com/armgong/RJulia> , accessed 07.10.2015.
62. MOA Massive Online Analysis. Real Time Analytics for Data Streams: official website. Available at: <http://moa.cms.waikato.ac.nz/> , accessed 07.10.2015.
63. OpenNN. Artniks - Predictive Analytics, Deep Learning, Data Mining, Big Data: official website. Available at: <http://www.artelnics.com/opennn> , accessed 07.10.2015.
64. scikit-learn: machine learning in python. scikit-learn : official website. Available at: <http://scikit-learn.org/stable/> , accessed 07.10.2015.
65. SciPy.org. SciPy.org: official website. Available at: <http://www.scipy.org/> , accessed 07.10.2015.
66. Shogun. The SHOGUN Machine Learning Toolbox: official website. Available at: <http://www.shogun-toolbox.org/> , accessed 07.10.2015.
67. Torch. Scientific Computing for LuaJIT: official website. Available at: <http://torch.ch/> , accessed 07.10.2015.
68. Collobert R., Kavukcuoglu K., Farabet C. Torch7: A Matlab-like Environment for Machine Learning. Ronan Collobert homepage. Available at: http://ronan.collobert.com/pub/matos/2011_torch7_nipsw.pdf , accessed 07.10.2015.
69. Waffles. A Machine Learning Toolkit: official website. Available at: <http://waffles.sourceforge.net/> , accessed 07.10.2015.
70. Weka3 - Data Mining with Open Source Machine Learning Software in Java. Weka: the University of Waikato: official website. Available at: <http://www.cs.waikato.ac.nz/~ml/weka/> , accessed 07.10.2015.
71. Hall M., Witten I., Frank E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2011. 629 p.
72. GNU Octave. GNU Octave: official website. Available at: <http://www.gnu.org/software/octave/> , accessed 07.10.2015.

73. Packages Octave. GNU Octave Wiki: official website. Available at: <http://wiki.octave.org/Packages> , accessed 07.10.2015.
74. Orange Data Mining. Orange: official website. Available at: <http://orange.biolab.si/> , accessed 07.10.2015.
75. R: The R Project for Statistical Computing. The R Project: official website. Available at: <https://www.r-project.org/> , accessed 07.10.2015.
76. CRAN - Mirrors. The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/mirrors.html> , accessed 07.10.2015.
77. CRAN Task View: Machine Learning & Statistical Learning. The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/web/views/MachineLearning.html> , accessed 07.10.2015.
78. Cass S. *The 2015 Top Ten Programming Languages*. Posted 20.07.2015. IEEE Spectrum: website. Available at: <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages> , accessed 15.10.2015.
79. RapidMiner. RapidMiner - #1 Open Source Predictive Analytics Platform: official website. Available at: <https://rapidminer.com/products/studio/> , accessed 07.10.2015.
80. RapidMiner Press Releases, Boston - June 17, 2014. RapidMiner - #1 Open Source Predictive Analytics Platform: official website. Available at: <https://rapidminer.com/news-posts/rapidminer-acquires-radoop/> , accessed 07.10.2015.
81. Apache Mahout: Scalable Machine Learning and Data Mining. Apache Mahout™ : official website. Available at: <http://mahout.apache.org/> , accessed 08.10.2015.
82. MapReduce. Wikipedia, the free encyclopedia: internet-portal. Available at: <https://ru.wikipedia.org/wiki/MapReduce> , accessed 08.10.2015.
83. Jubatus. Jubatus : Distributed Online Machine Learning Framework: official website. Available at: <http://jubat.us/en/> , accessed 08.10.2015.
84. MLib | Apache Spark. Apache Spark: official website. Available at: <http://spark.apache.org/mllib> / , accessed 08.10.2015.
85. Vowpal Wabbit. GitHub: public repository. Available at: https://github.com/JohnLangford/vowpal_wabbit/wiki , accessed 08.10.2015.
86. Kabakov R.I. *R v deistvii. Analiz i vizualizatsiya dannykh v programme R* [R in action. Analysis and visualization of data in the R program]. Moscow, DMK Press, 2014. 588 p. (in Russian).
87. McKinney W. *Python for data analysis*. O'Reilly Media, 2012. 466 p. (Russ. ed.: McKinney W. *Python i analiz dannykh*. Moscow, DMK Press, 2015. 482 p.).
88. CRAN Task Views. The Comprehensive R Archive Network: official website. Available at: <https://cran.r-project.org/web/views/> , accessed 08.10.2015.
89. Search all R packages and function manuals. Rdocumentation: website. Available at: <http://www.rdocumentation.org/> , accessed 08.10.2015.

90. shiny: Web Application Framework for R. The Comprehensive R Archive Network : official website. Available at: <https://cran.r-project.org/web/packages/shiny/index.html> , accessed 15.10.2015.
91. RStudio - Shiny Server. RStudio: official website. Available at: <http://www.rstudio.com/products/shiny/shiny-server/> , accessed 15.10.2015.
92. Hodgkin-Huxley model in R. mages' blog. Available at: <http://www.magesblog.com/2012/06/hodgkin-huxley-model-in-r.html> , accessed 08.10.2015.
93. What is the best neural network library for Python?. Quora : internet-portal. Available at: <https://www.quora.com/What-is-the-best-neural-network-library-for-Python> , accessed 08.10.2015.
94. Python for Artificial Intelligence. Python Wiki: internet-portal. Available at: [https://wiki.python.org/moin/ PythonForArtificialIntelligence](https://wiki.python.org/moin/PythonForArtificialIntelligence) , accessed 08.10.2015.
95. Calling Python from R with rPython. R-bloggers: internet-portal. Available at: <http://www.r-bloggers.com/calling-python-from-r-with-rpython/> , accessed 08.10.2015.
96. Accessing R from Python using RPy2. R-bloggers: internet-portal. Available at: <http://www.r-bloggers.com/accessing-r-from-python-using-rpy2/> , accessed 08.10.2015.
97. R/SPlus - Python Interface. The Omega Project for Statistical Computing: official website Available at: <http://www.omegahat.org/RSPython/> , accessed 08.10.2015.