УДК 004.4

Степаненко Д.Б.
4 курс, институт интеллектуальных кибернетических систем
Национальный исследовательский ядерный университет «МИФИ»
Россия, г. Москва
Кокорев Д.С.
4 курс, институт интеллектуальных кибернетических систем
Национальный исследовательский ядерный университет «МИФИ»
Россия, г. Москва

SCIKIT-LEARN: МАШИННОЕ ОБУЧЕНИЕ В PYTHON

Аннотация: *Scikit-learn является известным программным модулем языка Python, содержащим широкий набор самых разнообразных алгоритмов машинного обучения, позволяющих решать самые нестандартные прикладные задачи. Данный модуль предназначен, в первую очередь, для людей, не являющихся специалистами в области анализа данных и машинного обучения, о чём свидетельствует сравнительная простота "говорящих" названий всех структурных элементов модуля, удобство их использования в совокупности, а также их комбинации с элементами (объектами и методами) других программных модулей языка Python. Scikit-learn распространяется под лицензией BSD, что позволяет использовать его как для академической деятельности, так и для коммерческой. Исходный код и документация данного модуля может быть найдена на портале http://scikit-learn.org/stable.*

Ключевые слова: *Python, машинное обучение, обучение с учителем, обучение без учителя, анализ данных, подбор модели.*

Annotation: *Scikit-learn is a well-known Python software module containing a wide range of the most diverse machine learning algorithms, allowing to solve*

unconventional and sophisticated problems. This module is intended primarily for people who are not specialists in the field of data analysis and machine learning, as it mentioned by the rather simple names of all structural elements of the module and the convenience of their use in aggregate, as well as their combinations with elements (objects and methods) of other Python program modules. Scikit-learn is distributed under the BSD license, which makes it possible to use it for academic and commercial purposes. The source code and documentation of this module can be found on http://scikit-learn.org/stable.

Key words: *Python, machine learning, supervised learning, unsupervised learning, data analysis, model selection.*

1. Введение

Язык программирования Python зарекомендовал себя как один из самых популярных языков для проведения научных рассчётов в рамках разного рода исследований. Благодаря своей универсальности и большому количеству библиотек он является одним из лучших для алгоритмической разработки и исследовательского анализа данных по сравнению с другими языками программирования [1]. Тем не менее, как язык общего назначения, он все чаще используется не только в академических целях, но и в промышленных.

Scikit-learn использует богатую среду для реализации многих известных алгоритмов машинного обучения, поддерживая простой в использовании интерфейс, тесно интегрированный с языком Python. Это отвечает растущей потребности в анализе статистических данных в программной и интернет-индустрии, а также в областях, не относящихся к информатике, таких как биология или физика. Scikit-learn отличается от других инструментов машинного обучения следующим:

- распространяется под лицензией BSD;
- включает скомпилированный код для повышения эффективности, в отличие от MDP и pybrain;

- для облегчения распространения зависит только от numpy и scipy, в отличие от pymvpa, который имеет необязательные зависимости, такие как R;
- фокусируется на императивном программировании, в отличие от pybrain, который использует структуру потока данных.

Несмотря на то, что пакет в основном написан на Python, он также включает две библиотеки C++: LibSVM и LibLinear, которые предоставляют реализации SVM и обобщенных линейных моделей. Бинарные пакеты доступны на большом наборе платформ, включая Windows и любые платформы POSIX. Кроме того, благодаря своей либеральной лицензии он распространяется как часть крупных бесплатных программных дистрибутивов, таких как Ubuntu, Debian, Mandriva, NetBSD и Macports и в коммерческих дистрибутивах, таких как «Enthought Python Distribution».

2. Обзор проекта

Целью проекта является обеспечение стабильных реализаций вместо предоставления дополнительного функционала и возможностей. Высокое качество кода достигается с помощью модульных тестов — покрытие тестов составляет около 81%, а также с помощью использования инструментов статического анализа, таких как pyflakes и pep8. Наконец, разработчики Scikit-learn используют последовательный порядок именования для функций и параметров в соответствии с принципами построения кода на языке Python и документацией стиля numpy.

Разработчики Scikit-learn для обеспечения удобства процесса совместной разработки используют инструменты совместной работы, такие как git и github.

В качестве документации для модуля Scikit-learn представлено руководство пользователя объёмом около 300 страниц, включающее описательную

документацию, учебные пособия, инструкции по установке, а также более 60 примеров практического использования библиотеки.

3. Основные технологии

В качестве основных технологий для реализации модуля Scikit-learn были использованы библиотеки Numpy и Scipy, а также язык программирования, упрощающий написание модулей C\C++ кода для Python.

Библиотека Numpy используется для реализации базовой структуры данных, используемая для параметров модели. Входные данные представлены в виде массивов numpy, поэтому они легко интегрируются с другими библиотеками Python. Библиотека numpy также обеспечивает поддержку основных арифметических и матричных операций.

Библиотека Scipy поддерживает эффективные алгоритмы линейной алгебры, специальные функции, обработку сигналов и изображений. Scipy имеет привязку ко многим стандартным числовым пакетам на основе языка программирования Fortran, таких как LAPACK, что является важным аспектом для упрощения установки.

Язык Cython является объединением языков C и Python. Код Cython преобразуется в C/C++ код для последующей компиляции и далее может использоваться как расширение Python или как независимое приложение со встроенной библиотекой выполнения Cython.

4. Особенности разработки программного модуля с использованием модуля Scikit-learn

Для использования внешних по отношению к Scikit-learn объектов в совокупности с имеющимися в модуле не рекомендуется использовать наследование — вместо этого, как правило, используется совместимый интерфейс, которому было уделено огромное внимание при разработке [2].

Основным объектом рассматриваемого модуля является estimator, который реализует метод fit, отвечающий за настройку модели на предоставленной тестовой выборке. Классы модели, находящиеся в классе задач обучения с учителем, например, SVM (от англ. *support vector machine — метод опорных векторов*), также реализуют метод predict, обеспечивающий составление некоторого рода прогноза по тестовой выборке. Кроме того, некоторые наследники класса estimator, т.н. transformer-классы (как, например, PCA (от англ. *principal component analysis — метод главных компонент*)) имеют метод transform, который позволяет менять формат входных данных модели. Достаточно странным представляется рассуждение о моделях и методах машинного обучения без упоминания о необходимости подсчёта значения выбранной метрики качества получаемых результатов — класс estimator в общем случае реализует метод score: он позволяет “на лету” посчитать показатель выбранной метрики качества. Другим важным классом Scikit-learn является cross-validation iterator, предоставляющий разнообразные методы *скользящего контроля*, среди которых можно выделить наиболее востребованные: *K-Fold*, *leave-one-out* или *поэтапный скользящий контроль*.

Данный программный модуль может вычислять показатели качества работы класса estimator, автоматически подбирать параметры скользящего контроля, а также распределять вычисления по нескольким потокам. Подобный функционал достигается во многом благодаря наличию класса GridSearchCV, который, как правило, служит “обёрткой” для других классов — например, при вызове вышеописанного метода fit, произойдёт не только настройка параметров модели на предоставленную тестовую выборку, но и автоматическая максимизация выбранной метрики качества (метод score) за счёт варьирования данных параметров. “CV” в названии класса указывает на специфику его функциональных возможностей: “cross-validated” или “прошедшее процедуру скользящего контроля”. Также во многих случаях полезным оказывается объект pipeline, позволяющий создавать некоторый

симбиоз (именно в каноническом для биологии смысле — в значении взаимовыгодного существования) классов типа transformers и estimator-классов для выполнения более сложных операций с данными, например, с целью понижения размерности входных данных перед настройкой модели, что, вообще говоря, часто может сыграть решающую роль в вопросах получения модели, адекватно описывающей исследуемый объект [3].

5. Удобство или эффективность — существует ли компромисс?

В то время как разработчики Scikit-learn концентрируются, в первую очередь, на вопросах удобства использования — создавая в качестве инструмента для решения задач машинного обучения программный модуль с интерфейсом, являющимся расширением языка программирования высокого уровня под названием Python, — возникает проблема низкой производительности подобных решений, однако и эту проблему авторам модуля удалось решить: в Scikit-learn отсутствует излишнее копирование, что позволяет сократить количество хранимых данных до 40% в сравнении с предыдущими библиотеками (например, libsvm); кроме того, равное количество информации занимает меньше места в памяти компьютера при использовании данного модуля, чем при использовании его предшественников, что позволяет успешно использовать его и для работы с большим объёмом данных; наконец, в завершении разговора об оптимизации представления информации при использовании модуля, следует отметить наличие возможности автоматического удаления неиспользуемых данных, вместо безрассудного их пересчёта.

Итак, вычислительные возможности данного модуля при сравнении по конкретным показателям качества для методов решения определённых типовых задач машинного обучения вполне может составить конкуренцию для аналогичных библиотек на более низкоуровневых языка программирования, например C или Fortran, что, при наличии вышеописанных

удобств при использовании Scikit-learn, делает данный модуль одним из фаворитов на сегодняшний день в конкурентной борьбе за титул лучшего инструмента для решения задач машинного обучения.

6. Заключение

Программный модуль Scikit-learn предоставляет широкий диапазон функциональных возможностей для решения задач машинного обучения: и для обучения с учителем, и для обучения без учителя. Модуль опирается на вполне сформировавшуюся научно-исследовательскую экосистему языка Python, что позволяет легко интегрировать необходимый функционал в уже существующие проекты по анализу данных и машинному обучению, а простота и удобство использования данного модуля открывает новые возможности для людей, участвующих в исследованиях самого разного уровня и самых разных направлений, но не являющихся профессионалами в сфере анализа данных.

Использованные источники:

1. Mark Lutz: Programming Python: Powerful Object-Oriented Programming, O'Reilly Media, 2011, 1632 p.
2. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. Sebastopol, O’Reilly Media, Inc., 2017. 568 p.
3. Bowles M. Machine Learning in Python: Essential Techniques for Predictive Analysis. New Jersey, John Wiley & Sons, Inc., 2015. 360 p.