УДК 004.021

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

И.И. ИВАНОВ
(***Представлено: канд. физ.-мат. наук, доц. О.В. ГОЛУБЕВА***)

В статье рассматриваются наиболее важные современные алгоритмы машинного обучения. Анализ сфокусирован на помощи в выборе правильного подхода исходя из специфики решаемых задач в рамках предиктивной аналитики.

Введение. Алгоритмы машинного обучения можно описать как обучение целевой функции F, которая наилучшим образом соотносит входные переменные X и выходную переменную Y.

Неизвестно, что из себя представляет функция F. Ведь при её непосредственном наличии, она бы использовалась напрямую, а не предполагалась и уточнялась с помощью различных алгоритмов обучения.

Наиболее распространённой задачей в машинном обучении является предсказание значений Y для новых значений X. Это называется прогностическим моделированием, и главная цель — сделать как можно более точное предсказание.

Основная часть. Случайный лес — очень популярный и эффективный алгоритм машинного обучения. Это разновидность ансамблевого алгоритма, называемого бэггингом.

Бутстрэп является эффективным статистическим методом для оценки какой-либо величины вроде среднего значения. Вы берёте множество подвыборок из ваших данных, считаете среднее значение для каждой, а затем усредняете результаты для получения лучшей оценки действительного среднего значения.

В бэггинге используется тот же подход, но для оценки всех статистических моделей чаще всего используются деревья решений. Тренировочные данные разбиваются на множество выборок, для каждой из которой создаётся модель. Когда нужно сделать предсказание, то его делает каждая модель, а затем предсказания усредняются, чтобы дать лучшую оценку выходному значению.

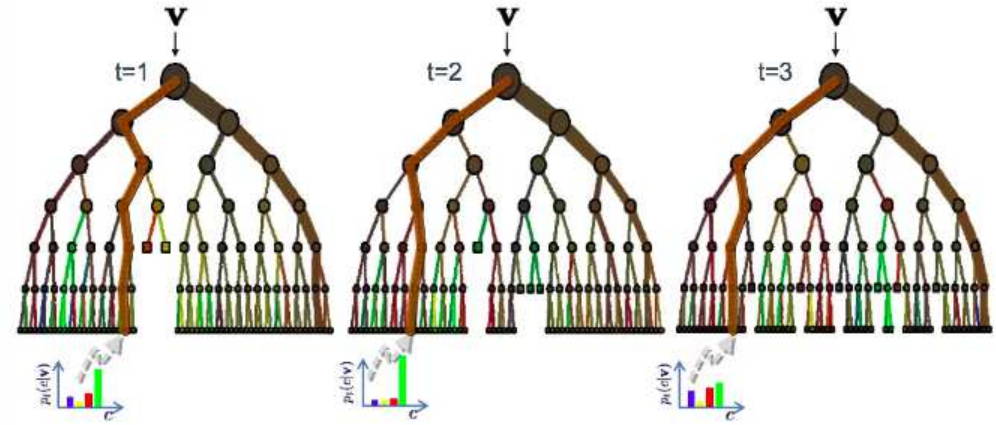

Рисунок 1. – Графическое представление алгоритма случайного леса

В алгоритме случайного леса для всех выборок из тренировочных данных строятся деревья решений. При построении деревьев для создания каждого узла выбираются случайные признаки. В отдельности полученные модели не очень точны, но при их объединении качество предсказания значительно улучшается. [1]

Если алгоритм с высокой дисперсией, например, деревья решений, показывает хороший результат на ваших данных, то этот результат зачастую можно улучшить, применив бэггинг.

Бустинг — это семейство ансамблевых алгоритмов, суть которых заключается в создании сильного классификатора на основе нескольких слабых. Для этого сначала создаётся одна модель, затем другая модель, которая пытается исправить ошибки в первой. Модели добавляются до тех пор, пока тренировочные данные не будут идеально предсказываться или пока не будет превышено максимальное количество моделей.

AdaBoost был первым действительно успешным алгоритмом бустинга, разработанным для бинарной классификации. Именно с него лучше всего начинать знакомство с бустингом. Современные методы вроде стохастического градиентного бустинга основываются на AdaBoost.

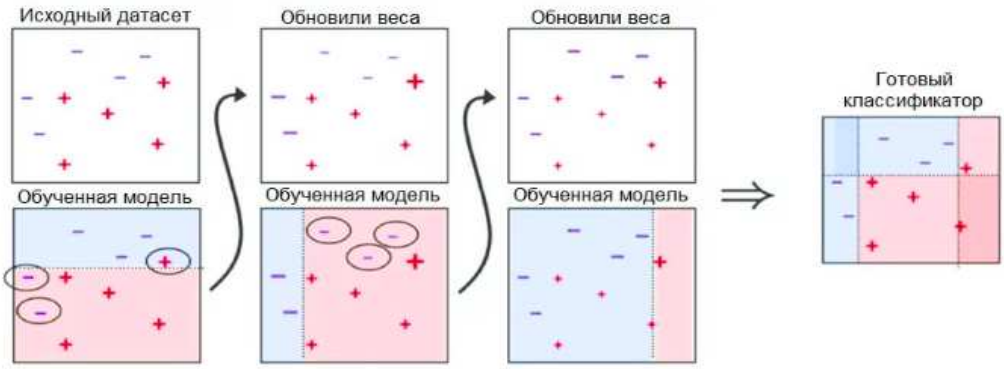

Рисунок 2. – Графическое представление алгоритма AdaBoost

AdaBoost используют вместе с короткими деревьями решений. После создания первого дерева проверяется его эффективность на каждом тренировочном объекте, чтобы понять, сколько внимания должно уделить следующее дерево всем объектам. Тем данным, которые сложно предсказать, даётся больший вес, а тем, которые легко предсказать, — меньший. Модели создаются последовательно одна за другой, и каждая из них обновляет веса для следующего дерева. После построения всех деревьев делаются предсказания для новых данных, и эффективность каждого дерева определяется тем, насколько точным оно было на тренировочных данных.

Так как в этом алгоритме большое внимание уделяется исправлению ошибок моделей, важно, чтобы в данных отсутствовали аномалии.

Метод опорных векторов, вероятно, один из наиболее популярных и обсуждаемых алгоритмов машинного обучения.

Гиперплоскость — это линия, разделяющая пространство входных переменных. В методе опорных векторов гиперплоскость выбирается так, чтобы наилучшим образом разделять точки в плоскости входных переменных по их классу: 0 или 1. В двумерной плоскости это можно представить как линию, которая полностью разделяет точки всех классов. Во время обучения алгоритм ищет коэффициенты, которые помогают лучше разделять классы гиперплоскостью.

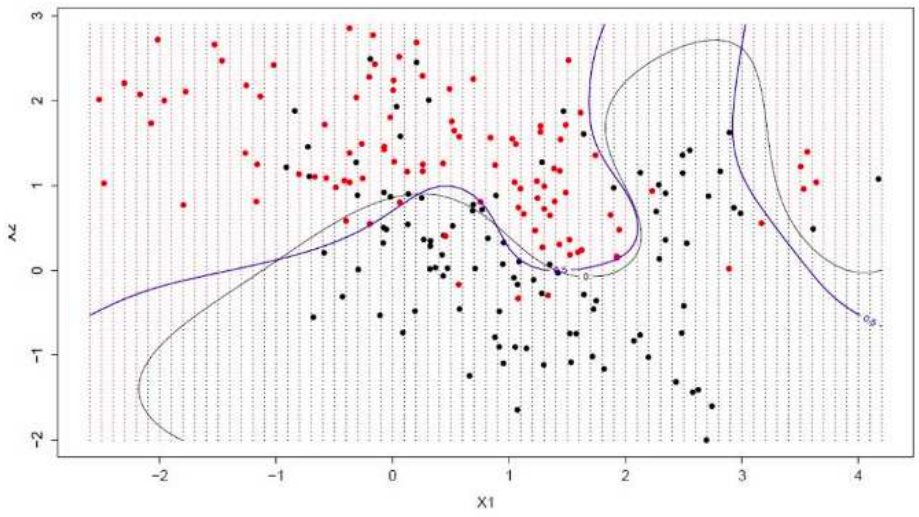

Рисунок 3. – Графическое представление метода опорных векторов в частном двухмерном случае

Расстояние между гиперплоскостью и ближайшими точками данных называется разницей. Лучшая или оптимальная гиперплоскость, разделяющая два класса, — это линия с наибольшей разницей. Только эти точки имеют значение при определении гиперплоскости и при построении классификатора. Эти точки называются опорными векторами. Для определения значений коэффициентов, максимизирующих разницу, используются специальные алгоритмы оптимизации. [2]

Метод опорных векторов, наверное, один из самых эффективных классических классификаторов, на который определённо стоит обратить внимание.

Заключение. Рассмотренные алгоритмы машинного обучения являются одними из самых эффективных на сегодняшний день. Примитивные алгоритмы, исключенные из рассмотрения, могут являться лишь пробным решением с малой точностью.

Метод опорных векторов является выбором многих успешных моделей с высокой точностью классификации в многомерном пространстве. Алгоритм может играть роль твердой базы для большинства задач и является практически универсальным.

Методы бустинга и бутстрэпа же рекомендуются к детальному рассмотрению при специфичности проблемы для решения.

Выбор конкретного алгоритма должен опираться на анализ следующих характерисик решаемой задачи:

- Размер, качество и характер данных;
- Доступное вычислительное время;
- Срочность задачи;
- Конечная специфика применения данных.

ЛИТЕРАТУРА

1. Multiclass Decision Forest [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-forest. – Дата обращения: 03.09.2019.
2. One-Class Support Vector Machine [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/one-class-support-vector-machine. – Дата обращения: 02.09.2019.