

М. В. Яцковец
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **А. В. Клименко**, канд. техн. наук, доцент

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ПРИ ПРОГНОЗИРОВАНИИ ПРОДАЖ

Прогнозирование продаж является ключевым фактором успешного ведения бизнеса. Это способ предугадать какое количество продукции будет реализовано и какая прибыль будет получена.

Степень важности прогнозирования продаж предопределила разработку множества способов определения будущих продаж, самые популярные среди них: метод экспертных оценок, анализ временных рядов и каузальные методы. Выбор метода прогнозирования зависит от множества факторов – релевантности доступных данных, желаемой степени точности, длительности прогнозируемого периода, ресурсов, соотношения потенциальной и реальной ёмкости рынка.

По мнению многих экспертов наиболее точными и эффективными являются каузальные методы, их основное преимущество – прямая связь с принятием решения. Сущность каузальных методов прогнозирования состоит в установлении математической связи между результирующей и факторными переменными. Необходимым условием применения каузальных методов прогнозирования является наличие большого объема данных для корректного выявления факторов. Если связи между переменными удастся описать математически корректно, то точность каузального прогноза будет достаточно высокой, поэтому при выполнении анализа продаж стоит использовать вычислительную технику и программное обеспечение, в частности, методы машинного обучения и нейросетевого моделирования. Они автоматизируют методы количественного прогнозирования продаж и упрощают ее, задача прогнозирования – очень кропотливая и трудоёмкая работа, требующая знания рынка и понимания процессов формирования спроса, анализа факторов, влияющих на ёмкость рынка, спрос и поведение конкурентов.

Для задачи прогнозирования объемов продаж каузальным методом подходят такие алгоритмы, как случайный лес, градиентный бустинг, метод опорных векторов. Наиболее эффективным алгоритмом считается алгоритм градиентного бустинга. Эффективность этого алгоритма обусловлена тем, что на каждой итерации выстраивается базовый алгоритм, работающий только на части подвыборки. На каж-

дом последующем шаге новое слагаемое вычисляется, опираясь на случайную подвыборку определенного размера.

Бустинг – итерационный алгоритм, реализующий «сильный» классификатор, позволяющий добиться произвольно малой ошибки при обучающей выборке на основании декомпозиции «слабых» классификаторов, каждый из которых лучше, чем угадывание, исходя из этого вероятность правильной классификации 23 больше $0,5$. Сущность заключается в использовании весовой версии одних и тех же обучающих данных вместо выбора случайных подмножества. Ключевое отличие бустинга от бэггинга: выборка для обучения на каждой итерации определяется, исходя из ошибок классификации, возникших на предыдущих итерациях [1].

Входные данные алгоритма:

- $\{(x_i, y_i)\}_{i=1, \dots, n}$ - набор данных;
- M - число итераций;
- $L(y, f)$ – функция потерь с выписанным градиентом;
- $h(x, \theta)$ – семейство функций базовых алгоритмов, с процедурой их обучения;
- $h(x, \theta)$ – дополнительные гиперпараметры (например глубина дерева у деревьев решений);
- $f_0(x)$ – начальное приближение заменено константой γ .

Выходные данные алгоритма:

- $f(x) = \sum f_i(x)$ - итоговая модель градиентного бустинга.

Шаг 1. И константным значением метод градиентного бустинга $f(x) = f_0, f_0 = \gamma, \gamma \in R, f_0 = \arg \min \sum L(y_i, \gamma). n_i = 1$.

Шаг 2. Для каждой итерации $t = 1, \dots, M$ повторять

1. Подсчет псевдо-остатков $rt: rit = -[dL(y_i, f(x_i))df(x_i)]f(x) = f(x), \text{ для } i = 1, \dots, n$.

2. Построение нового базового алгоритма $ht(x)$ как регрессию на псевдоостатках $\{(x_i, rit)\}_{i=1 \dots n}$.

3. Поиск оптимального коэффициента pt при $ht(x)$ относительно исходной функции потерь $pt = \arg \min \sum L(y_i, f(x_i) + p * h(x_i, \theta)). n_i = 1$.

4. Сохранение $f_t(x) = pt * ht(x)$.

5. Обновление текущего приближения $f(x): f(x) \leftarrow f(x) + f_t(x) = \sum f_i(x). ti = 0 \ 23$.

Шаг 3. Составление итоговой модели градиентного бустинга $f(x): f(x) = \sum f_i(x). M i = 0$.

Сложность алгоритма градиентного бустинга составляет в среднем $O(N^2)$, но растет с увеличением количества итераций, так как чем больше итераций, тем больше базовых алгоритмов.

Литература

1. Hastie, T. Chapter 15. Random Forests / The Elements of Statistical Learning: Data Mining, Inference, and Prediction. / T. Hastie, R. Tibshirani, J. Friedman. – Springer-Verlag, 2009. – 746 p.