

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 прогнози-

рования 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 является алгоритм градиентного бустинга. Это обусловлено тем, что алгоритм на каждой итерации строит базовый алгоритм, который действительно эффективен лишь на части подвыборки. На каждом шаге алгоритма новое слагаемое вычисляется, опираясь не на всю обучающую выборку, а лишь на случайную подвыборку фиксированного размера.

Бустинг – итерационный алгоритм, реализующий “сильный” классификатор, который позволяет добиться произвольно малой ошибки обучения (на обучающей выборке) на основе композиции «слабых» классификаторов, каждый из которых лучше, чем просто угадывание, т. е. вероятность правильной классификации > 0.5 . Ключевая идея: использование весовой версии одних и тех же обучающих данных вместо случайного выбора их подмножества. Основное отличие бустинга от бэггинга состоит в том, что обучающая выборка на каждой итерации определяется, исходя из ошибок классификации на предыдущих итерациях.

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

Выходные данные: итоговая модель градиентного бустинга $f(x) = \sum f_i(x)$.

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

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

1. Посчитать псевдо-остатки $r_t; r_{it} = -[dL(y_i, f(x_i))df(x_i)]f(x) = f(x)$, для $i = 1, \dots, n$

2. Построить новый базовый алгоритм $h_t(x)$ как регрессию на псевдоостатках $\{(x_i, r_{it})\}_{i=1 \dots n}$.

3. Найти оптимальный коэффициент p_t при $h_t(x)$ относительно исходной функции потерь $p_t = \arg \min \sum L(y_i, f(x_i) + p * h(x_i, \theta))$. $n_i = 1$.

4. Сохранить $f_t(x) = p_t * h_t(x)$.

5. Обновить текущее приближение $f(x): f(x) \leftarrow f(x) + f_t(x) = \sum f_i(x), t_i = 0 \dots 23$.

Шаг 3. Скомпоновать итоговую модель градиентного бустинга $f(x): f(x) = \sum f_i(x), M_i = 0$.

Конец алгоритма.

Сложность алгоритма градиентного бустинга можно определить как $O(N^2)$, но при этом стоит учитывать, что чем больше итераций, тем больше базовых алгоритмов для голосования и соответственно тем больше сложность алгоритма.

Таким образом, алгоритм градиентного бустинга выполняет M итераций, на каждой из которых происходит обучение базовой модели $L(y_i, \gamma)$, путем подбора оптимальных параметров γ . Следует отметить, что для настройки каждой базовой модели используется функция потерь $\varphi(y, y')$, вообще говоря, отличная от $L(y, y')$.

Список использованных источников

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