

Смесь гауссиан для прогнозирования цены

Оцениваем распределение

Батраков Юрий

Предпосылки

Допустим мы обучили модель машинного обучения, которая позволяет с определённой точностью предсказывать следующее значение стоимости некоторого актива. Но мы не знаем, насколько модель “уверена” в своём предсказании, каков доверительный интервал для этого предсказания. Решить это можно было бы, если бы модель предсказывала не одно значение — наиболее вероятное с её точки зрения предсказание, а целое распределение вероятностей по всевозможным предсказаниям.

Математическая модель

Имеются данные $X \in \mathbf{R}^{T \times D}$ где T – количество наблюдений, а D – количество признаков, необходимо предсказывать y .

1. Будем предсказывать не y_t , а $y_t - y_{t-1}$, тогда распределение предсказываемой величины будет находится недалеко от 0
2. Введём латентные переменные π, μ, σ^2 , которые будут задавать распределение $p(y_t - y_{t-1} | x_{t-1}, \dots, x_{t-n}, y_{t-1}, \dots, y_{t-n}) = \sum_{i=1}^K \pi_i N(\mu_i, \sigma_i^2)$, причём $\sum_{i=1}^K \pi_i = 1, \sigma_i^2 \geq 0$
3. Теперь мы предсказываем латентные переменные, получаем распределение и уже из него выбираем точку-статистику Δy

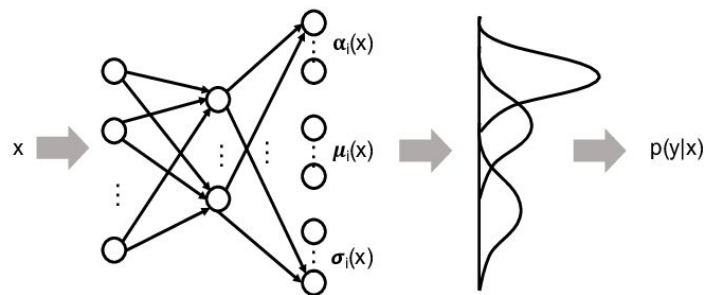
Преимущества модели

1. Получаем полные данные о “знаниях” модели по данной ситуации
2. Знаем насколько модель “уверена” и в зависимости от этого можем либо принимать, либо не принимать предсказание, также можем напрямую посчитать VaR
3. Можем выбирать разные статистики из распределения в качестве предсказания: мат. ож., медиана, мода
4. Можно оценивать доверительные интервалы
5. Можно оценить с какой вероятностью цена вырастет, а с какой упадёт
6. Так как модель учит не сами y , а латентные переменные задающие распределение на y , то модель должна меньше переобучаться

Недостатки модели

1. Перепараметризация модели: для объяснения данных нужно настроить много параметров, а для этого нужно много данных, в приложении к финансовым котировкам, это проблема
2. Гауссианы имеют слабые хвосты, поэтому задать распределение честно не получится, будет описана только наиболее вероятная часть носителя
3. Возможно распределение имеет существенно другую структуру
4. Выбросы обрабатываются скорее всего плохо

Реализация



Возьмём нейронную сеть, передадим ей имеющиеся данные, она выделит полезные признаки и этот вектор признаков отдаём в MDN

MDN — три головы: первая предсказывает π , вторая μ и третья σ . Чтобы выполнить ограничения математической модели к первому предсказанному вектору применим Softmax, а к последнему экспоненту*

В качестве функции потерь можно взять отрицательное правдоподобие, при этом все преобразования дифференцируемые, поэтому сеть может учиться

Технические аспекты

1. Согласно [1] лучше использовать не обычный Softmax, а Gumbel Softmax, поскольку хотелось бы, чтобы модель больше выделяла гауссианы с большими логитами
2. При этом, чтобы сеть не стала выдавать предсказания всегда в одну и ту же гауссиану необходимо добавить регуляризацию на голову, предсказывающую π
3. Согласно [2] вместо экспоненты предлагается использовать ELU, что позволяет модели проще переводить свои предсказания в дисперсию
4. Согласно [3] использование GRU в качестве backbone для MDN даёт наилучшие результаты

Результаты

Логично предположить, что модель хорошо описывает данные, если, например, математическое ожидание хорошо описывает цену в следующий момент.

На почасовых данных от Yahoo Finance по USD/RUB и по нефти марки Brent за последний год:

- В качестве baseline взял CatBoostRegressor, на данных за последние 10 дней дал $RMSE = 0.5743$
- Модель на смеси гауссиан через математическое ожидание с данными за 10 дней дала $RMSE = 0.5197$

Перспективы развития

1. Загрузка данных поминутно, для большего количества информации
2. В условиях перепараметризации и малого количества данных возможно полезными окажутся байесовские методы
3. Использование в качестве backbone BERT'a
4. Попробовать другие ядра для оценки распределения
5. Попробовать feature engineering: хотя нейронные сети обычно в нём не нуждаются, в [4] утверждается, что в данной модели он оказывает сильное влияние на результат
6. Оптимизация гиперпараметров
7. Сделать интерфейс для удобного анализа предсказанного распределения в реальном времени

Литература

1. C. M. Bishop, Mixture density networks (1994)
2. Guillaumes, A.B., Mixture Density Networks for distribution and uncertainty estimation (2017)
3. Xiaoming Li & Chun Wang & Xiao Huang & Yimin Nie, A GRU-based Mixture Density Network for Data-Driven Dynamic Stochastic Programming (2020)
4. Narendhar Gugulothu, Sparse Recurrent Mixture Density Networks for Forecasting High Variability Time Series with Confidence Estimates (2019)