Введение в Финансовые Рынки

Отчёт по проекту - Осень 2022

Смесь гауссиан для прогнозирования цены

Батраков Юрий

под руководством Куликова А.В.

Содержание

1 Введение

В задаче предсказания стоимости финансовых активов на основе исторических данных есть разные подходы, но предсказывая стоимость, мы всегда получаем желаемое число, потому что алгоритм обязан его выдать, но мы не знаем насколько он "уверен" в своём решении, возможно решение лучшее из возможных в данный момент, но глобально не особо хорошее. Как же получить "уверенность"? Можно предсказывать целое распределение вероятностей на следующее значение цены. Этому подходу и посвящена данная работа.

2 Математическая модель

2.1 Описание

Имеется набор данных $X \in \mathbb{R}^{T \times D}$, где T — количество наблюдений, а D — количество признаков. Кроме того имеется набор предсказываемых значений $Y \in \mathbb{R}^T$, которые необходимо научиться предсказывать.

Во-первых, будем предсказывать не y_t, а прирост цены, т.е. $y_t - y_{t-1}$. Таким образом новая предсказываемая величина $\Delta_t = y_t - y_{t-1}$ лежит около нуля, при этом теряется один пример из данных.

Во-вторых, сделаем предположение, что распределение Δ_t от известных данных можно представить как сумму "ядерных" оценок, количество которых фиксированно и является гиперпараметром, а параметры задаются известными данными.

В-третьих, выберем в качестве такого "ядра" гауссовский колокол, тогда финально вводим математическую модель:

$$p(\Delta_t | X_{t-1}, \dots, X_0, y_{t-1}, \dots, y_0) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mu_i, \sigma_i^2),$$

где K — количество ядер, которыми аппроксимируется распределение, $\pi \in \mathbb{R}^K$ — вектор, задающий вероятности принадлежности точки к одной из гауссиан, такой что $\sum\limits_{i=1}^{K} \pi_i = 1$, $\mu \in \mathbb{R}^K$ — вектор центров гауссиан, $\sigma^2 \in \mathbb{R}_+^K$ — вектор дисперсий гауссиан.

Соответственно теперь задача сведена к предсказанию трёх латентных переменных π, μ, σ^2, оцениваемых поделью по $X_{t-1}, \dots, X_0, y_{t-1}, \dots, y_0$. Чтобы использовать эту модель в машинном обучении будем оптимизировать правдоподобие, соотвественно в качестве функции

потерь возьмём отрицательный логарифм правдоподобия:

$$NLL = -\sum_{n=1}^{B} \ln \left(\sum_{i=1}^{K} \pi_i \left[\mathcal{N}(\mu_i, \sigma_i^2) \right] (\Delta_n) \right),$$

где B — размер батча, Δ_n — n-ый пример предсказываемой величины из батча.

2.2 Преимущества

- Модель предоставляет полные данные о "знаниях" в данный момент;
- Известна "уверенность" модели, что позволяет оперативно либо принимать, либо не принимать предсказание, также есть возможность напрямую посчитать VaR;
- Можно выбирать разные статистики из распределения в качестве предсказания: математическое ожидание (можно показать, что оно выражается как $\pi^T \mu$), медиану, моду;
- Можно оценивать доверительные интервалы для статистик;
- Можно оценивать с какой вероятностью цена вырастет, а с какой упадёт (это может быть полезно для оценки риска от покупки опциона), для этого достаточно посчитать функцию распределения в нуле $F(0) = \sum\limits_{i=1}^{K} \pi_i F_i(0)$, где F — функция распределения $p(\Delta_t | \ldots)$, F_i — функция распределения нормального распределения с параметрами μ_i, σ_i^2;
- Так как модель будет учить не сами y, а латентные переменные задающие распределение на y, то модель должна меньше переобучаться.

2.3 Недостатки

- В модели теперь предсказывается не одно число, а $3K$ чисел, модель перепараметризована, соответственно чтобы научиться предсказвать их больше нужно много данных, в приложении к финансовым котировкам, это проблема;
- Нормальное рапределение имеет слабые хвосты, поэтому задать распределение честно не получится (реальное распределение скорее всего не пикообразное), будет описана только наиболее вероятная часть носителя с точки зрения модели;
- Возможно реальное распределение имеет существенно другую структуру.

3 Архитектура

3.1 Backbone

Чтобы обработать последовательность данных $X_{t-1}, \ldots, X_0$ и $y_{t-1}, \ldots, y_0$ и выделения фиксированного количество признаков, необходим некоторый обработчик. Здесь можно попробовать несколько подходов, наиболее мощными из них: свёрточные нейронные сети, рекурентные нейронные сети, трансформеры. При обработке последовательностей рекурентные сети и трансформеры показывают себя лучше,но при этом трансформерам необходимо коллосальное количество данных, поэтому в качестве обработчика выберем рекурентную нейронную сеть. Среди них тоже есть определённый выбор, наиболее используемыми являются vanilla RNN, LSTM, GRU, последние две имеют долгосрочную память о данных последовательности и чаще всего применяются на практике. В [1] показано, что GRU модель демонстрирует себя хорошо, поэтому окончательно остановим свой выбор на ней.

3.2 Mixture Density Network

Итак после обработчика получен вектор признаков $h \in \mathbb{R}^N$, тогда сделаем три линейных преобразования, $W_1, W_2, W_3 \in \mathbb{R}^{N \times K}$ и на выходе имеем три вектора размерности K, вектор для π необходимо привести к виду вероятностей, для этого применим Gumbel Softmax (это слой как Softmax, только он сильнее выделяет отдельные гауссианы) по рекомендации из [2], а вектор для σ^2 необходимо привести к неотрицательному состоянию, это можно сделать поэлементно применив экспоненту, но в [3] предлагают воспользоваться ELU, увеличенному на 1, таким образом модели будет проще предсказывать дисперсию, так как зависимость будет линейной.

$$ELU(x) = \begin{cases} x & , x \geq 0 \\ e^x - 1 & , x < 0 \end{cases}$$

Чтобы модель не стала всё время выделять одну и ту же гауссиану добавим регуляризацию на π к функции потерь.

4 Результаты

Данные возьмём из открытого API Yahoo Finance, будем предсказывать курс USD/RUB, в качестве дополнительных данных будем использовать данные про нефть марки Brent. По-

лучилось загрузить почасовые данные за 2021 и 2022 годы, однако не все данные там идут с разницей в час, на данном этапе это было опущено и предсказывалось всегда следующее значение когда бы оно ни было, но в целом возможно стоит добавить дополнительное преобразование для таких случаев. Данные разделил на обучающую и тестовую выборку в соотношение $4:1$ соответственно (с у чётом того что данные зависят от времени до определённой даты данные были обучающими, а с некоторой тестовыми).

В качестве точечной оценки из распределения возьмём математическое ожидание за лёгкость в вычислении. В качестве baseline возьмём бустинг CatBoostRegressor, который будет получать данные за последние n дней. Для честного сравнения будем предоставлять нашей модели не все данные, а также только за поледние n дней.

n	RMSE baseline	RMSE RMDN	RMSE RMDN при уверенности	Доля предсказаний
1	0.5881	0.5846	0.5268	5%
5	0.5913	0.5750	0.5686	25%
10	0.5743	0.5767	0.5197	12%

Основное преимущество модели заключается в том, что она выбирает в каких случиях делать предсказание (если плотность рапределения $p(y_t|\ldots)$ выше некоторого порога), а в каких нет, таким образом образом понижая ошибку на тестовых данных.

5 Перспективные направления развития

- Использование большего количества данных (как по историческим масштабам, так и по разнообразию, хотелось бы добавить данные по потребительским способностям/инной показатель отражающий инфляцию в стране, данные относительно новостной повестки);
- В условиях перепараметризации и малого количества данных возможно полезными окажутся байесовские методы;
- При большем количестве данных использовать в качестве backbone BERT;
- Попробовать другие “ядра” для оценки распределения;
- Попробовать генерацию более богатого признакового описания: хотя нейросетевые архитектуры обычно в нём не нуждаются, в [4] утверждается, что в данной модели оно оказывает сильное влияние на результат;

- Изучение различных преобразований при отсутствии данных в определённый промежуток времени;
- Оптимизация основных гиперпараметров (количество гауссиан, регуляризация, размер скрытых слоёв сети);
- Сделать интерфейс для удобного анализа предсказанного распределения в реальном времени.

6 Заключение

Полученная модель решает поставленную перед ней задачу оценки уверенности в предсказании, что позволяет повысить точность предсказания: выбираются только те предсказание, где модель увереннее, чем заданный порог, что позволяет снизить ошибку в предсказании. Кроме того, полученную модель можно применять для других задач, прогресс в нашей задачи показывает, что подход многообещающий и можно продолжать исследования.

Список литературы

1. Xiaoming Li & Chun Wang & Xiao Huang & Yimin Nie, A GRU-based Mixture Density Network for Data-Driven Dynamic Stochastic Programming (2020)

2. C. M. Bishop, Mixture density networks (1994)

3. Guillaumes, A.B., Mixture Density Networks for distribution and uncertainty estimation (2017)

4. Narendhar Gugulothu, Sparse Recurrent Mixture Density Networks for Forecasting High Variability Time Series with Confidence Estimates (2019)