Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра системного программирования

Крайчик Георгий Ильич

Методы предсказания динамики изменения биржевых котировок

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:
к. ф.-м. н., доцент Графеева Н. Г.

Рецензент:
д. ф.-м. н., профессор Нестеров В. М.

Санкт-Петербург
2015

SAINT PETERSBURG STATE UNIVERSITY

Software Engineering Chair

Kraichik Georgii

Predictive methods of stock exchange dynamics

Bachelor's Thesis

Admitted for defence.
Head of the chair:
Ph.D., Professor Terekhov A.N.

Scientific supervisor:
Ph.D., Associate Professor Grafeeva N.G.

Reviewer:
Ph.D., Professor Nesterov V.M.

Saint-Petersburg
2015

Оглавление

Введение

Задача прогнозирования финансовых временных рядов была и остается актуальной, поскольку предсказание является необходимым элементом любой инвестиционной деятельности. Сама идея инвестирования – вложения денег с целью получения дохода в будущем – основывается на идее прогнозирования будущего. В последнее время стали доступны мощные средства сбора и обработки информации, что предоставляет возможность применять различные ресурсоемкие Data Mining методы. Широкое применение Data Mining в данной области обусловлено наличием в большинстве временных рядов сложных зависимостей, которые не удается обнаружить, прибегая к помощи прочих методов.

Торговля ценными бумагами осуществляется на специальных площадках, называемых фондовыми биржами. Среди основных задач фондовой биржи можно выделить следующие: организация торговли ценными бумагами, стандартизация контрактов, обеспечение гарантий по исполнению сделок, установление равновесной рыночной цены по каждому активу и прочее [12].

Одной из самых важных функций, выполняемой фондовой биржей, является перераспределение денежных средств в экономике, а также предоставление доступа к рынку капитала. Инвесторы, у которых имеются временно свободные денежные средства, имеют возможность вложить их в ценные бумаги, тем самым передавая их тем участникам торгов, которые в них нуждаются. Этот механизм стимулирует развитие экономики, поскольку денежные средства, вложенные в ценные бумаги, начинают "работать" у других участников торгов, тем самым способствуя развитию экономики.

Помимо осуществления своих основных функций, биржа предоставляет возможность участниками торгов получать прибыль за счёт изменения котировок ценных бумаг. Например, инвестор может купить актив по одной цене и продать его через некоторое время по более высокой цене, тем самым зафиксировав собственную прибыль. В связи с этим на бирже появляются участники торгов, единственной целью которых является получение прибыли за счёт колебания цен активов. Такие трейдеры обеспечивают т.н. ликвидность ценных бумаг - способность к достаточно быстрому закрытию позиций по установившимся рыночным ценам. Это приводит к повышению инвестиционного качества ценных бумаг, увеличивая привлекательность компаний для потенциальных инвесторов.

Фондовая биржа, являясь организатором торгов, взимает комиссию за каждую заключаемую участниками сделку. Способ удержания комиссионных сборов варьируется в зависимости от типа контракта. Чаще всего, при осуществлении операций купли-продажи акций, бирже уплачивается доля от всего объёма заключаемой сделки, в то время как при заключении фьючерсного контракта, бирже уплачивается некоторая фиксированная величина.

Для получения торговой прибыли, инвесторам необходимо уметь предсказывать динамику изменения биржевых котировок для построения эффективных торговых стратегий. Существует множество различных подходов, прогнозирующих будущие цены активов, однако в настоящее время все большую и большую популярность приобретают методы кластеризации временных рядов. Эта работа будет посвящена исследованию применимости методов кластерного анализа для построения эффективных торговых стратегий на фондовых биржах.

1. Справочная информация

1.1. Винеровский случайный процесс

Определение: Случайным процессом называется семейство случайных величин $X(t) = X(t, \omega)$, заданных на одном вероятностном пространстве $(\Omega, \mathbf{F}, \mathbf{P})$ и зависящих от параметра t, принимающего значения из некоторого непустого множества T. [11]

Обозначать случайный процесс будем $\{X_t\}_{t \in T}$. Обычно под индексом t понимается время. В тех случаях, когда T не более чем счётно, $\{X_t\}_{t \in T}$ называется процессом с дискретным временем. Если же T является связным подмножеством вещественной прямой, то $\{X_t\}_{t \in T}$ называется процессом с непрерывным временем.

Определение: Случайный процесс $\{X_t\}$, $t \in [0, T]$, $T \in \mathbf{R}$ называется винеровским процессом, если он обладает следующими свойствами:

1. $\{X_t\}$ - процесс с независимыми приращениями (т.е. $\forall t_0, ..., t_n \in [0, T]$ случайные величины $X_{t_0}, X_{t_1} - X_{t_0}, ..., X_{t_n} - X_{t_{n-1}}$ независимы).

2. $X_0 = 0$ почти наверное.

3. $X_t - X_s \in N(0, \sigma^2(t-s))$ $\forall s, t : 0 \leq s < t \leq T$, где $N(0, \sigma^2(t-s))$ является множеством нормально распределённых случайных величин со средним 0 и дисперсией $\sigma^2(t-s)$

1.2. Факторизация временных рядов

В задачах прогнозирования финансовых временных рядов часто возникает потребность в усреднении нескольких рядомстоящих элементов временного ряда, "склеивая" их в одно значение. Для предсказания динамики изменения биржевых котировок периодически требуется вычислять средневзвешенные по объёму значения котировок актива. Такое "склеивание" нескольких значений воедино будем называть факторизацией временного ряда. Дадим формальное определение этой процедуры.

Определение: Левосторонней факторизацией временного ряда $\{x_i\}_{i=1}^N$ по временному ряду $\{v_i\}_{i=1}^N$ с шагом $n \in \mathbf{N}$ будем называть временной ряд $\{y_k\}_{k=1}^M = lfact(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n)$, такой что $y_k = \sum\limits_{i=nk-n+1}^{\min\{nk,N\}} x_i v_i \ : \sum\limits_{i=nk-n+1}^{\min\{nk,M\}} v_i$, где $M = \lceil \frac{N}{n} \rceil$.

В этом определении ряд $\{x_i\}_{i=1}^N$ выступает в роли котировок актива, а ряд $\{v_i\}_{i=1}^N$ в роли количества заключённых контрактов (объёма). В процессе факторизации вычисляется средневзвешенное по объёму значение котировка актива.

Рассмотрим пример. Задача: получить левостороннюю факторизацию временного ряда
$\{x_i\}_{i=1}^5 = \{10, 20, 30, 40, 50\}$ по $\{v_i\}_{i=1}^5 = \{6, 1, 7, 3, 8\}$ с шагом $n = 3$.

$M = \lceil \frac{5}{3} \rceil = 2$, $y_1 = \frac{10 \cdot 6 + 20 \cdot 1 + 30 \cdot 7}{6+1+7} \approx 20.71$, $y_2 = \frac{40 \cdot 3 + 50 \cdot 8}{3+8} \approx 47.27$. Таким образом, результатом будет являться временной ряд $\{20.71, 47.27\}$.

Определение: Перевёрнутым рядом $\{x_i\}_{i=1}^N$ будем называть $\{y_i\}_{i=1}^N = reverse(\{x_i\}_{i=1}^N)$, где $y_i = x_{N-i+1}$.

Пример: $reverse(\{1, 2, 3\}) = \{3, 2, 1\}$.

Определение: Правосторонней факторизацией временного ряда $\{x_i\}_{i=1}^N$ по $\{v_i\}_{i=1}^N$ с шагом $n \in \mathbf{N}$ будем называть $reverse(lfact(reverse(\{x_i\}_{i=1}^N), reverse(\{v_i\}_{i=1}^N), n))$ (для краткости обозначать будем $rfact(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n)$).

По своей сути, правосторонняя факторизация отличается от левосторонней только тем, что "склеивание" осуществляется справа налево, а не слева направо. Для примера, приведённого выше, правосторонней факторизацией будет являться ряд $\{\frac{20 \cdot 1 + 10 \cdot 6}{1+6}, \frac{50 \cdot 8 + 40 \cdot 3 + 30 \cdot 7}{8+3+7}\} \approx \{11.43, 40.56\}$.

Замечание: В тех случаях, когда N кратен n, левосторонняя и правосторонняя факторизации всегда совпадают.

1.3. Задача линейной регрессии

Пусть имеется два набора данных $\{x_i\}_{i=1}^Z, \{y_i\}_{i=1}^Z$. Метод линейной регрессии ищет прямую вида $y = ax + b$, такую что сумма квадратов расстояний точек (x_i, y_i) до неё минимально. То есть коэффициенты a и b подбираются таким образом, что $\sum_{i=1}^{Z} (\hat{y_i} - y_i)^2 \to \min$. Коэффициент a будем называть наклоном регрессионной прямой. В этой работе решением задачи линейной регрессии будем называть коэффициент a при условии, что коэффициент $b = 0$.

Будем обозначать решение задачи линейной регрессии через $regr(\{x_i\}_{i=1}^Z, \{y_i\}_{i=1}^Z)$. В тех случаях, когда нам потребуется искать решение для k правых / левых элементов последовательностей $\{x_i\}_{i=1}^Z$, $\{y_i\}_{i=1}^Z$, будет осуществляться поиск решения задачи линейной регрессии для факторизованных временных рядов.

$$rregr(\{x_i\}_{i=1}^Z, \{y_i\}_{i=1}^Z, k) \stackrel{\text{def}}{=} regr(\{x_i\}_{i=Z-k+1}^Z, \{y_i\}_{i=Z-k+1}^Z)$$

$$lregr(\{x_i\}_{i=1}^Z, \{y_i\}_{i=1}^Z, k) \stackrel{\text{def}}{=} regr(\{x_i\}_{i=1}^k, \{y_i\}_{i=1}^k)$$

В рамках этой работы реализована библиотека на языке C#, решающая задачу поиска линейной регрессии.

1.4. Решение задачи линейной регрессии для факторизованных временных рядов

Пусть $\{x_i\}_{i=1}^N$ - временной ряд котировок актива, $\{v_i\}_{i=1}^N$ - ряд объёмов заключённых сделок. Требуется найти решение задачи линейной регрессии (используя Q точек)

результата факторизации ряда $\{x_i\}_{i=1}^N$ по ряду $\{v_i\}_{i=1}^N$ с шагом $n \in \mathbf{N}$ (для краткости, далее будет рассматриваться только правосторонняя факторизация).

Сначала получим правостороннюю факторизацию $\{y_i\}_{i=1}^M = rfact(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n)$ (см. п.1.2). Далее вычислим $\{d_i\}_{i=1}^M = rfact(\{1, 2, ..., N\}, \{1, ..., 1\}, n)$ и потом вычислим $rregr(\{d_i\}_{i=1}^M, \{y_i\}_{i=1}^M, Q)$. Описанную выше процедуру будем обозначать *rrf*.

$$rrf(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n, Q) \stackrel{\text{def}}{=} rregr(rfact(\{1, ..., N\}, \{1, ..., 1\}, n), rfact(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n), Q)$$

Аналогичным образом можно определить процедуру *lrf*:

$$lrf(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n, Q) \stackrel{\text{def}}{=} lregr(lfact(\{1, ..., N\}, \{1, ..., 1\}, n), lfact(\{x_i\}_{i=1}^N, \{v_i\}_{i=1}^N, n), Q)$$

1.5. Технические индикаторы

Техническим индикатором называется функция, построенная на значениях статистических показателей торгов (цены, объём торгов и т. д.), анализ поведения которой призван ответить на вопрос изменится или сохранится текущая тенденция на рынке. На основе анализа технических индикаторов принимаются решения об открытии или закрытии торговых позиций [9].

Технические индикаторы будут использованы в работе при формировании векторов признаков в алгоритме кластеризации финансовых временных рядов.

1.5.1. Экспоненциальное скользящее среднее

Экспоненциальное скользящее среднее (EMA - Exponential Moving Average) - один из наиболее полезных трендовых индикаторов. EMA помогает выявлять тренды и находить наилучшие моменты времени для открытия позиций. Индикатор зависит от единственного параметра $\alpha \in (0, 1]$.

Значение индикатора EMA с параметром α в момент времени $i \in [1..M]$ временного ряда $\{y_i\}_{i=1}^M$ будем обозначать через EMA(α, i). Через EMA(α) будем обозначать последовательность $\{\text{EMA}(\alpha, i)\}_{i=1}^M$. Ниже приведена методика расчёта величины EMA(α, t):

$$\text{EMA}(\alpha, t) \stackrel{\text{def}}{=} \begin{cases} y_1 & t = 1 \\ \alpha y_t + (1 - \alpha)\text{EMA}(\alpha, t - 1) & t \geq 2 \end{cases}$$

Параметр α отвечает за чувствительность индикатора к новым данным. При $\alpha \approx 0$ данные временного ряда $\{y_i\}$ практически не учитываются, поскольку $\alpha y_t \approx 0$. Таким образом, EMA(α) $\approx \{y_1\}_{i=1}^M$. В другом крайнем случае при $\alpha = 1$, получаем последовательность $\{y_i\}_{i=1}^M$, что свидетельствует о том, что EMA(1) полностью совпадает с исходным временным рядом. Из всего вышесказанного можно сделать вывод, что чем больше α, тем чувствительнее индикатор к новым данным исходного временного

ряда.

Индикатор EMA может применяться в качестве сглаживания исходного временного ряда $\{y_i\}_{i=1}^{M}$. EMA(α) нивелирует скачкообразность данных, делая их более гладкими. На Рисунке 1 изображен график временного ряда, а также графики последовательностей EMA(0.05) и EMA(0.005). Как видно из изображения, индикатор EMA(0.05) достаточно точно повторяет график временного ряда (при этом сглаживая его), а EMA(0.005) похож на прямую линию, задающую направление тренда на всем представленном промежутке времени.

Следует обратить внимание, что параметр α является чувствительностью к распознаванию трендов. При уменьшении α индикатор выделяет только крупные тенденции, в то время как при увеличении α индикатор начинает распознавать более мелкие колебания значений временного ряда.

Также следует отметить, что индикатор EMA работает с некоторым запозданием. Ему требуется время для того, чтобы "настроиться" на новую рыночную тенденцию. По графику EMA(0.05) можно видеть, что его экстремумы всегда находятся несколько правее экстремумов исходного временного ряда.

Рис. 1: График временного ряда и индикатор EMA

1.5.2. Схождение/расхождение экспоненциальных скользящих средних

Схождение/расхождение экспоненциальных скользящих средних (Exponential Moving Average Convergence/Divergence – EMACD) представляет собой разность двух экспоненциальных скользящих средних с различными параметрами. При этом при расчёте EMACD всегда из более чувствительного экспоненциального среднего вычитается ме-

нее чувствительное.

$$\text{EMACD}(\alpha_1, \alpha_2, t) \stackrel{\text{def}}{=} \text{EMA}(\alpha_2, t) - \text{EMA}(\alpha_1, t)$$

$$\text{EMACD}(\alpha_1, \alpha_2) \stackrel{\text{def}}{=} \{\text{EMACD}(\alpha_1, \alpha_2, t)\}_{t=1}^{M}$$

где $0 < \alpha_1 < \alpha_2 \leq 1$

При анализе двух скользящих экспоненциальных средних с разными параметрами α, более быстрое EMA будет отражать более свежие ожидания рынка, в отличие от более медленного EMA. А значит, если EMACD выше 0, то это сигнал к началу восходящего тренда, а если EMACD ниже 0, то следует ожидать снижения цены.

Для предвосхищения схождения двух скользящих средних (т.е. приближения EMACD к 0), применяют α_3-сглаживание значений EMACD (при этом параметр α_3 выбирается таким образом, чтобы $\alpha_2 < \alpha_3 \leq 1$). Получаемая таким образом последовательность называется сигнальной. Экспоненциально сглаженный с параметром α_3 временной ряд $\text{EMACD}(\alpha_1, \alpha_2)$ будем обозначать $\text{EMACD}_{\alpha_3}(\alpha 1, \alpha 2)$ (сигнальный временной ряд).

При помощи сигнальной линии EMACD можно получить более ранние сигналы изменения настроений участников торгов. Если EMACD пересекает свою сигнальную линию снизу вверх, то это является сигналом к началу восходящего тренда, если EMACD пересекает свою сигнальную линию сверху вниз, то это сигнал к появлению нисходящего тренда (см. Рисунок 2).

Рис. 2: График временного ряда и индикатор EMACD

1.5.3. Parabolic SAR

Индикатор Parabolic SAR (Stop and Reversal) имеет широкое применение для закрытия торговых позиций. SAR имеет несколько параметров, определяющих процесс его вычисления. Вычисляется SAR следующим образом [9]:

$$AL(al, step, mx, i) \stackrel{\text{def}}{=} \min\{al + (i-1) \cdot step, mx\} \quad \forall i \in [1..M]$$

$$SAR(sar, al, step, mx, i) \stackrel{\text{def}}{=} \begin{cases} sar & i = 1 \\ AL(i)(H(i-1) - SAR(i-1)) + SAR(i-1) & \text{long postion} \\ AL(i)(L(i-1) - SAR(i-1)) - SAR(i-1) & \text{short position} \end{cases}$$

где AL(i) (сокр. от AL($al, step, mx, i$)) - фактор ускорения в момент времени i.

H($i-1$), L($i-1$) - максимальная и минимальная цены актива за промежуток времени $i-1$; в тех случаях, когда исходный временной ряд отражает тиковые данные о торгах, будем полагать H $\equiv$ L.

В тех случаях, когда из контекста ясно, чему равны параметры $sar, al, step$ и mx, значение SAR в момент времени $i \in [1..M]$ будем сокращённо обозначать SAR(i). Аналогично предыдущим индикаторам последовательность $\{\text{SAR}(i)\}_{i=1}^{M}$ будем обозначать просто SAR.

Для успешного применения индикатора SAR для закрытия позиций необходимо закрывать длинные позиции в тех случаях, когда цена актива опускается ниже SAR, а короткие - в тех случаях, когда цена поднимается выше значений SAR.

Использование индикатора SAR не рекомендуется для открытия позиций, поскольку в случае возникновения бокового тренда, появляется множество ложных сигналов (см. Рисунок 3). Parabolic SAR лучше всего использовать совместно с индикатором ADX [9] для минимизации количества ложных сигналов.

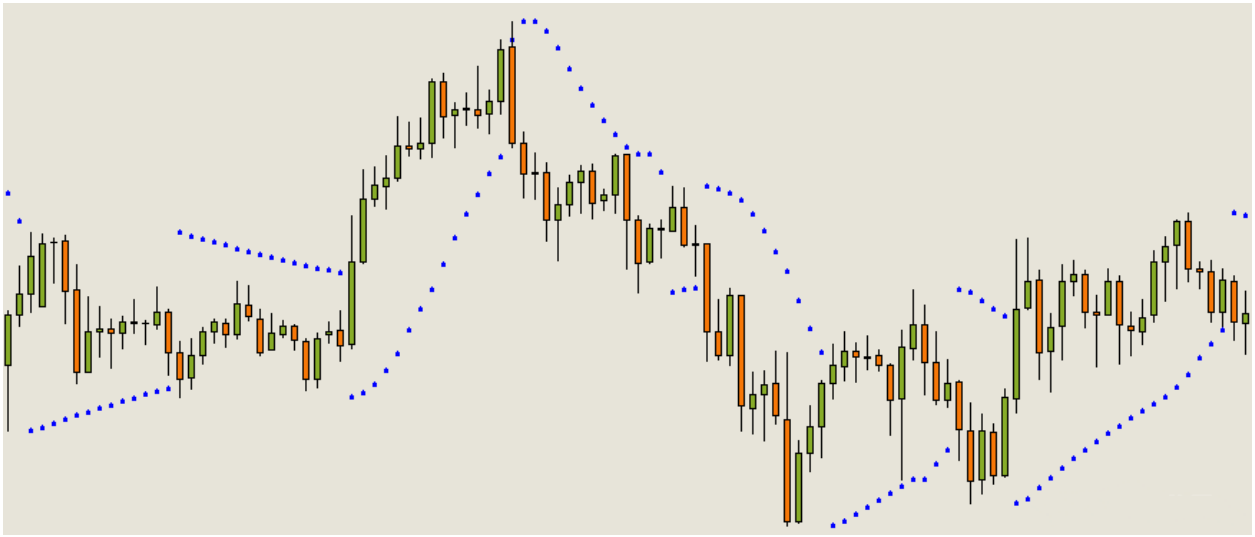

Рис. 3: График временного ряда и индикатор Parabolic SAR

1.5.4. Стохастический осциллятор

Стохастический осциллятор показывает положение текущей цены относительно диапазона цен за определённый период в прошлом. Основная идея его применения заключается в том, что при наличии возрастающего тренда цена закрытия за очередной промежуток времени имеет тенденцию останавливаться вблизи предыдущих максимумов. Аналогично при наличии на рынке нисходящего тренда, цена закрытия имеет склонность останавливаться вблизи предыдущих минимумов. Фактически, индикатор демонстрирует расхождение цены закрытия текущего периода относительно цен предыдущих периодов в рамках заданного временного промежутка.

Для построения индикатора сначала необходимо построить быстрый стохастический осциллятор.

$$\%K(n,i) \stackrel{\text{def}}{=} \begin{cases} \frac{C(i)-L(n,i)}{H(n,i)-L(n,i)} & i \geq n \\ \frac{C(i)-L(i,i)}{H(i,i)-L(i,i)} & i < n \end{cases}, \text{ где}$$

$C(i)$ - цена закрытия периода номер i

$L(n,i)$ - самая низкая цена за последние n периодов

$H(n,i)$ - самая высокая цена за последние n периодов

Введём обозначение: $\%K(n) \rightleftharpoons \{\%K(n,i)\}_{i=1}^{M}$. Вычислим медленный стохастический осциллятор $\%D(n)$, применив к временному ряду $\%K(n)$ экспоненциальное сглаживание EMA с некоторым параметром α (например, $\alpha = 0.2$). Последовательность $\%D(n)$ будем называть сигнальной.

Применять стохастичекий осциллятор на практике возможно следующим образом: если $\%K(n)$ пересекает $\%D(n)$ снизу, то это сигнал к покупке актива, если же $\%K(n)$ пересекает $\%D(n)$ сверху - это сигнал к продаже.

Рис. 4: График временного ряда и стохастический осциллятор

2. Существующие решения

Существует множество подходов к решению задачи прогнозирования финансовых временных рядов, использующих методы из самых разнообразных разделов математики. Все эти методы можно разделить на два класса: к первой категории относятся те методы, которые используют некоторую априорную информацию о характеристиках временного ряда, а ко второй категории относятся те методы, которые не учитывают никакой дополнительной информации, кроме входных данных.

К первому типу, например, относятся методы, основанные на решении стохастических дифференциальных уравнений (СДУ, см. раздел 2.1). В этих методах утверждается, что динамика изменения цены актива подчиняется некоторым закономерностям, описываемым СДУ. Это предположение довольно точно моделирует действительность, поэтому данные методы имеют широкое применение в области предсказания финансовых временных рядов.

Ко второму типу относятся методы вероятностного и статистического анализа, нейронные сети и методы кластеризации (см. разделы 2.2, 2.3). Преимущество данных методов заключается в высокой чувствительности к смене характеристик временного ряда (переобучаемости) и потенциально более высокой точности предсказания, поскольку данные модели позволяют выявлять скрытые зависимости данных, которые установить априорно не представляется возможным.

2.1. Стохастичесие дифференциальные уравнения

В литературе первое упоминание о СДУ появилось в начала двадцатого века в работах, связанных с описанием броуновского движения. Возникновение СДУ потребовало создания собственного исчисления, впоследствии получившего название теории стохастических интегралов [6]. Теория получила широкое применение в физике, биологии и химии, а также в теории вероятностей и финансовой математике. Наиболее известный и часто используемый пример СДУ - уравнение с т.н. белым шумом, рассматриваемым в качестве производной винеровского процесса (см. п.1.1).

2.1.1. Логнормальное распределение цены актива

Пусть S_τ - цена актива в момент времени $\tau \in [t, +\infty)$. Запишем в общем виде стохастическое дифференциальное уравнение (подробнее об СДУ см. [14]), описывающее динамику изменения цены S_τ с течением времени:

$$dS_\tau = a(S_\tau, \tau)d\tau + b(S_\tau, \tau)d\omega_\tau, \text{ где} \tag{1}$$

$a(S_\tau, \tau)$ - трендовая составляющая динамики изменения цены актива.

$b(S_\tau, \tau)$ - вероятностная составляющая динамики изменения цены актива (т.н. белый шум).

ω_τ - винеровский случайный процесс (см п.1.1)

Для моделирования динамики изменения цены бездивидендного актива принято рассматривать частный случай формулы (1) при $a(S_\tau, \tau) = aS_\tau$, $b(S_\tau, \tau) = \sigma S_\tau$
$(a \in \mathbf{R}, \sigma \in \mathbf{R}_+)$ [15]:

$$dS_\tau = (aS_\tau)d\tau + (\sigma S_\tau)d\omega_\tau \tag{2}$$

Определение: Стохастическое дифференциальное уравнение (2) называется геометрическим броуновским движением при начальных данных $S_t = S$.

Следует обратить внимание, что если из уравнения убрать вероятностную составляющую $(\sigma S_\tau)d\omega_\tau$, то СДУ (2) превратится в обыкновенное дифференциальное уравнение первого порядка $S'_\tau = a \cdot S_\tau$, у которого имеется единственное решение $S_\tau = Se^{a(\tau - t)}$ при начальных данных $S_t = S$. Это хорошо характеризует поведение актива в долгосрочной перспективе (в долгосрочной перспективе цена актива определяется в основном трендовой составляющей), поскольку благосостояние экономики при отсутствии внешних воздействий растёт экспоненциально быстро [1]. Параметр a в данном случае играет роль относительного приращения цены актива за малый промежуток времени. Чем больше a, тем быстрее растёт величина S_τ.

С другой стороны, наличие вероятностной составляющей $\sigma(S_\tau)d\omega_\tau$ добавляет элемент случайности в процесс ценообразования актива. Коэффициент σ отвечает за степень разброса величины S_τ. Чем больше σ, тем больше волатильность актива [5] и, как следствие этого, величина S_τ начинает вести себя более непредсказуемо.

Решение СДУ (2), можно найти в явном виде:

$$S_\tau = Se^{(a - \frac{\sigma^2}{2})(\tau - t)} \cdot e^{\sigma(\omega_\tau - \omega_t)} \tag{3}$$

Заметим, что из определения винеровского процесса (см. п 1.1) следует, что
$\xi \equiv \omega_\tau - \omega_t$ является нормально распределённой случайной величиной ($\xi \in N(0, \tau - t)$). Прологарифмировав левую и правую части равенства (3), получаем выражение $ln(\frac{S_\tau}{S}) = (a - \frac{\sigma^2}{2})(\tau - t) + \sigma\xi$. Отсюда можно получить свойства случайной величины $\frac{S_\tau}{S}$:

1. $\mathbf{E}\ ln(\frac{S_\tau}{S}) = (a - \frac{\sigma^2}{2})(\tau - t)$. Отсюда видно, что чем больше значение параметра a, тем больше математическое ожидание $ln(\frac{S_\tau}{S})$ и, следовательно, тем больше величина S_τ.

2. $\mathbf{D}\ ln(\frac{S_\tau}{S}) = \sigma^2(\tau - t)$. Это означает, что σ влияет на дисперсию случайной величины S_τ. При этом стандартное отклонение $ln(\frac{S_\tau}{S})$ прямо пропорционально величине σ.

3. $ln(\frac{S_\tau}{S}) \in N((a - \frac{\sigma^2}{2})(\tau - t), \sigma^2(\tau - t))$. Это в точности означает, что случайная

величина S_τ имеет логнормальное распределение [10].

Таким образом, сделав предположение, что цена актива измеяется согласно уравнению (2), можно сделать вывод, что котировка S_τ является логнормально распределённой случайной величиной. Основываясь на статистических данных о поведении цены актива, вычисляются параметры a и σ. Далее, обладая всей необходимой информацией о распределении S_τ, появляется возможность предсказывать значения будущих котировок и строить прибыльные торговые стратегии.

Недостатки модели:

1. В реальности параметры a и σ не могут являться константами, вследствие постоянно изменяющейся рыночной конъюнктуры. Модель может достаточно хорошо функционировать в условиях развитой и устойчивой экономики.

2. Дисперсия случайной величины $ln(\frac{S_\tau}{S})$ прямо пропорциональна времени, прошедшему от начала отсчёта t. Это означает, что данный подход не дает точных долгосрочных прогнозов, в связи с увеличением разброса случайной величины S_τ.

3. На рисунке 5 представлены графики теоретического распределения (логнормального) и реального распределения (эмпирического, построенного на основе статистической информации о поведении S_τ). Из рисунка видно, что в тех случаях, когда S_τ оказывается в окрестности своего математического ожидания, модель достаточно хорошо предсказывает поведение актива. Однако логнормальное распределение S_τ сильно занижает вероятности больших отклонений цены от своего ожидаемого значения, что в реальности зачастую оказывается неверным. Реальное распределение имеет "толстые хвосты" (fat-tailed distribution), поэтому при применении модели следует учитывать, что вероятность крупных отклонений цены от ожидаемых значений значительно выше моделируемой.

Описанная модель широко применяется для оценки рисков производных финансовых инструментов и расчёте показателя Value at Risk [15].

2.1.2. Модель стохастической волатильности Хестона

Модель Хестона является уточнением модели, описанной в (2.1.1). Она допускает, что динамика изменения цены бездивидендного актива подчиняется закону геометрического броуновского движения, однако предполагает, что волатильность цены актива является не постоянной величиной, а случайным процессом [4].

В соответствии с моделью Хестона, динамика изменения цены и волатильности

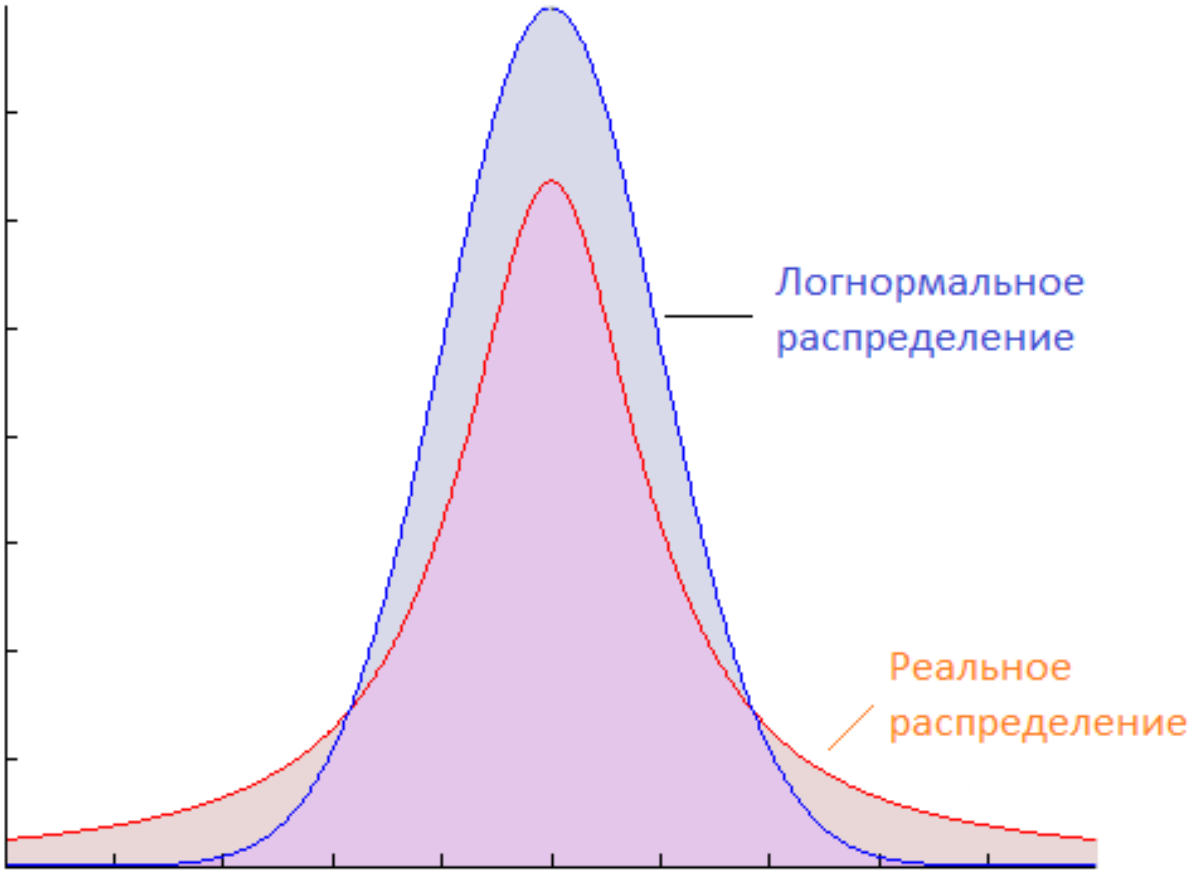

Рис. 5: Логнормальное и эмпирическое распределение S_τ

актива описываются системой двух СДУ: [7]

$$\begin{cases} dS_\tau = \mu S_\tau d\tau + \sqrt{\nu_\tau} S_\tau dz_\tau \\ d\nu_\tau = k(\theta - \nu_\tau)d\tau + \sigma\sqrt{\nu_\tau} d\omega_\tau \end{cases} \tag{4}$$

где ω_τ, z_τ - стандартные винеровские процессы, $d\omega_\tau dz_\tau = \rho d_\tau$, ρ - коэффициент корреляции между процессами ω_τ, z_τ.

k - скорость возвращения к равновесной дисперсии доходности

θ - равновесная дисперсия доходности

σ - волатильность дисперсии доходности

Модель стохастической волатильности Хестона лучше приближает теоретическое распределение цены актива к реальному (эмпирическому) распределению по сравнению с моделью геометрического броуновского движения с постоянной волатильностью (2.1.1). Модель Хестона учитывает корреляцию между ценой и волатильностью, что позволяет лучше отразить потенциальную изменчивость рыночной конъюнктуры, давая более точные оценки будущих значений котировок актива. Несмотря на

перечисленные преимущества, модель Хестона имеет множество параметров, не изменяющихся с течением времени, что свидетельствует об ограниченной точности производимых вычислений.

2.2. Нейронные сети

Математическое понятие нейронной сети появилось с середины 1940-ых годов после публикации статьи У. Маккалока и У. Питтса о логическом исчислении идей и нервной активности. С 1950-ых теория нейронных сетей активно развивалась в области решения задач искусственного интеллекта. Нейросети получили широкое распространение в решении задач прогнозирования, распознавания образов, дискриминантного анализа и др.

Нейронные сети представляют собой систему соединённых и взаимодействующих между собой т.н. искусственных нейронов. Каждый нейрон имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим нейронам. Будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, группы нейронов способны решать довольно сложные задачи кибернетики, адаптивного управления, прогнозирования и др. [8]

У нейросетей имеется ряд преимуществ: во-первых, нейросетевой анализ, не предполагает никаких ограничений на характер входной информации, нейронная сеть самообучаема. Во-вторых, нейросети способны находить оптимальные для данного временного ряда индикаторы и строить по ним оптимальную стратегию предсказания. Кроме того, одним из значимых преимуществ нейросетей является то, что они не используют никаких априорных утверждений о характере данных (в отличие от моделей, описанных в п.2.1.1, п.2.1.2).

Несмотря на вышеперечисленные достоинства, практика показывает, что обучение нейросетей на "сырых" данных не дает ожидаемого качества прогнозирования. Перед подачей данных на вход нейросети, практически всегда требуется их преобразование к такому виду, в котором нейросеть смогла бы лучше уловить скрытые закономености. Такая подготовка данных называется предобработкой. Обычно к "сырым" данным применяется некоторый алгоритм, преобразовывающий их в более "понятную" для нейросети форму. Как показывает практика, именно предобработка данных является ключом к эффективному функционированию нейронных сетей.

Несмотря на огромный потенциал нейронных сетей в решении задач прогнозирования временных рядов, на практике качество их работы не всегда соответствует ожиданиям. Связано это, в первую очередь, с не очевидным процессом выбора топологии сети и алгоритма предобработки данных. В настоящее время, в решении задач прогнозирования финансовых временных рядов, все больше и больше приобретают популярность методы кластерного анализа данных.

2.3. Кластеризация

Кластеризация - это процедура разбиения всех элементов некоторого множества на группы по принципу их схожести. В одну группу (по-другому класс или кластер) должны попадать те элементы множества, которые в некотором смысле "схожи" друг с другом. В то же время всякая пара элементов из разных классов, должна иметь существенные отличия друг от друга (в рамках той же самой меры "похожести"). Кластеризация - одно из самых обширных направлений Data Mining. Методами кластерного анализа можно решать задачи порождения и проверки гипотез, разработки классификации и проч.

Методы кластеризации могут применяться для достижения самых различных целей. Во-первых, кластерный анализ применим для более глубокого понимания данных путём выявления кластерной структуры. Разбиение множества на группы схожих объектов позволяет упростить и повысить качество дальнейшей обработки данных, применяя к каждому кластеру свой метод анализа. Во-вторых, кластеризация применима для сжатия анализируемых данных. После разбиения на кластеры, дальнейшие исследования проводятся на наиболее типичном его представителе, при этом оставшиеся элементы кластера в дальнейшем не используются. В-третьих, при помощи кластеризации возможно выявление нетипичных объектов. С её помощью находятся те элементы множества, которые не похожи ни на один существующий кластер.

Независимо от предмета изучения, применение кластерного анализа предполагает следующие этапы:

1. Отбор выборки для кластеризации. Входные данные должны удовлетворять критериям однородности и полноты. Однородность требует, чтобы все объекты выборки были одной природы и описывались сходным набором характеристик. Критерий полноты данных накладывает ограничение на репрезентативность выборки.

2. Предподготовка данных. Во многих случаях необходимо использовать некоторые преобразования исходных данных, которые бы способствовали повышению качества кластеризации. Более того, практически во всех случаях требуется проведение нормализации данных, что будет описано далее.

3. Введение пространства признаков. Всякому объекту исходной выборки ставится в соответствие некоторый вектор признаков, который является набором характеристик конкретного объекта.

4. Введение меры сходства (расстояния между векторами) в пространстве признаков. Чем меньше расстояние между парой объектов, тем более они "похожи". В то же время, чем расстояние больше, тем меньше у них сходств.

5. Применение одного из алгоритмов кластеризации - непосредственное разбиение исходного множества на группы (в большинстве задач эти группы не пересекаются, однако существуют алгоритмы, разбивающие множество объектов и на пересекающиеся кластеры).

Для того, чтобы приблизиться к пониманию, каким образом кластеризация применима к прогнозированию финансовых временных рядов, необходимо погрузиться в биржевую терминологию. На рынке ценных бумаг широко используется понятие тренда (или по-другому тенденции). Понятие тренда применяется для описания текущих рыночных настроений (рыночных тенденций) к изменению биржевых котировок. Обычно на рынке выделяют три основных типа тренда: восходящий (когда цена актива склонна к росту), нисходящий (при стрлемлении цены актива к снижению), а также боковой (когда цена актива колеблется в окрестности своего текущего значения). Основным свойством тренда является то, что он имеет некоторую протяжённость во времени. Если каким-то образом получится достоверно определить, что в определённый момент времени элемент временного ряда попадает в один из трёх трендов, то с некоторой степенью уверенности можно судить, что несколько следующих за выбранным элементом точек будут принадлежать этому же тренду. Если же в исходный момент времени занять соответствующую тренду позицию (для восходящего тренда - длинную, а для нисходящего - короткую), то через некоторый промежуток времени появится возможность закрыть позицию с получением прибыли.

Таким образом, каждую точку финансового временного ряда будем классифицировать к одному из трех кластеров - к восходящему, нисходящему или боковому тренду. Основной трудностью решения этой задачи является выбор вектора признаков и задания функции расстояния между ними. В настоящее время кластеризация финансовых временных рядов является очень актуальной темой, поэтому имеется множество статей, предлагающих к рассмотрению различные способы введения вектора признаков и функции расстояния.

Один из способов динамической кластеризации временных рядов описан в статье И.Д. Полосухина [17]. В этой статье кластеризация осуществляется не в рамках одного временного ряда, а проводится кластеризация множества временных рядов. Иными словами, на группы разбиваются не элементы временного ряда, а сами временные ряды. Первым элементом вектора признаков, представленном в этой статье, было выбрано его математическое ожидание (матожидание множества его элементов). В качестве второго элемента вычислялось его стандартное отклонение. Третий элемент - показатель "тренд" (см. статью [17]). Четвёртым элементом рассчитывался показатель, основанный на использовании коэффициентов разложения ряда по методу главных компонент (использовался метод "Гусеницы" [2]). Пятый - показатель Хёрста [13]. В качетсве функции похожести было выбрано Евклидово расстояние, при кластеризации использовался алгоритм k-среднее. Результаты приведены в вышеупо-

мянутой статье [17].

Помимо разбиения временного ряда на непересекающиеся кластеры, имеются алгоритмы т.н. нечёткой кластеризации, когда каждому элементу временного ряда сопоставляется не класс, которому он принадлежит, а вектор вероятностей принадлежности элемента к каждому из существующих кластеров. (Один из наиболее актуальных методов описан в статье А.К. Тищенко и И.П. Плисса [18]) Нечёткая кластеризация оказывается весьма эффективной при прогнозировании финансовых временных рядов, поскольку предоставляет наиболее "честный" ответ на вопрос, к какому классу следует отнести очередную точку. Более того, подобный вероятностный подход, предоставляет информацию, позволяющую проводить глубокий анализ рисков, возникающий при построении торговых стратегий.

Поскольку кластеризация является одной из самых актуальных и перспективных направлений прогнозирования финансовых временных рядов, данная работа будет посвящена разработке, апробации и сравнению торговых стратегий, основанных на кластерном анализе данных.

3. Постановка задачи

Для построения торговых стратегий, необходимо разработать некую формальную модель, в рамках которой будут проводиться исследования, построение и апробация торговых стратегий. Поскольку работа посвящена прогнозированию финансовых временных рядов, в качестве подобной модели будет выступать специальная машина, моделирующая поведение роботов, осуществляющих торговую деятельность на реальных фондовых рынках. Сущность строящийся машины будет двоякой: с одной стороны, машина будет моделировать операции купли-продажи активов, тем самым принимая на себя задачу осуществления торговых поручений и проведения корректных расчётов, а с другой - проводить анализ данных и принимать торговые решения.

Построение подобного робота позволит формализовать задачу разработки торговых стратегий, а также предоставит инструментарий для сравнения их друг с другом. Модель будет максимально приближена к настоящим торговым роботам, осуществляющим деятельность на реальных фондовых биржах. Практически все свойства, присущие торговле на реальных рынках, будут перенесены на строящуюся модель.

3.1. Определение машины Robot

Введём понятие машины *Robot*, задачей которой будет осуществление торговой деятельности на фондовой бирже. Будем считать, что зафиксированы дискретные моменты времени $\{t_n\}_{n=1}^{M}$, причём $\forall i, j \in [1..M]\ t_i < t_j \Leftrightarrow i < j$. В каждый момент времени t_i машина находится в состоянии $Robot(pos_i, money_i, dcn_i)$, где

1. pos_i - объем открытой позиции (в контрактах) в момент времени t_i; причем если $pos_i > 0$, то занята длинная позиция по активу, а если $pos_i < 0$, то занята короткая позиция по активу.

2. $money_i$ - объем доступных машине денежных средств в момент времени t_i.

3. dcn_i - функция (алгоритм) принятия торговых решений. Формально, $dcn_i : \mathbf{R}^{k_i} \to \mathbf{Z}$, где k_i - количество исторических данных, используемых функцией dcn_i для принятия торгового решения. Функция возвращает размер открываемой позиции (в контрактах). Если функция вернула положительное значение, необходимо занять длинную позицию по активу, если же отрицательное - короткую. В тех случаях, когда будет возвращён 0, никаких торговых операций проводить не следует.

Входной информацией для машины *Robot* является последовательность котировок актива $\{d_n\}_{n=0}^{M}$, где d_0 является символическим обозначением всей имеющейся (и интересующей нас) информации о биржевых котировках до момента времени t_1. Данные d_0, например, будут использоваться в качестве обучающей выборки

для алгоритма кластеризации. Остальные данные являются вещественными числами $\forall i \in [1..M]\ d_i \in \mathbf{R}$.

Машина *Robot* способна исполнять следующие команды:

1. $Init(money)$ - процедура инициализации машины в момент времени t_0. Помимо объёма денежных средств, на вход машине подаются некоторые другие параметры (о них речь пойдёт позже).

2. $Trade(amount, price)$ - процедура исполнения торговых поручений. $amount$ - количество контрактов, $price$ - цена, по которой будет заключаться торговая сделка. Перед исполнением поручения, *Robot* проверяет корректность проводимой операции, проверяя возможность заключения сделки заданного объёма по заданной цене. Чтобы открыть длинную позицию по активу, $amount$ должен быть положительным, чтобы открыть короткую - отрицательным. Результатом работы $Trade$ является объем (в контрактах) фактически открытых позиций.

3. $Move(\{d_0, ..., d_i\})$ - функция перехода из i-ого состояния машины в $i+1$.

До момента времени t_1, машина находится в начальном состоянии $Robot(pos_0 = 0, money_0, dcn_0)$. Опишем i-ый шаг машины, $i \in [1..M]$. В момент времени t_i на вход машине поступает информация об очередном значении котировки d_i, в результате чего запускается функция $Move$, которая в зависимости от значения функции принятия торгового решения dcn_i, переводит машину из $i-1$ состояния в i-ое. Таким образом, в очередной момент времени t_i осуществляется переход

$$Robot(pos_{i-1}, money_{i-1}, dcn_{i-1}) \xrightarrow{Move(\{d_0,...,d_i\})} Robot(pos_i, money_i, dcn_i) \tag{5}$$

В момент времени M машина обязана завершить свою работу, т.е. оказаться в состоянии $Robot(0, money_M, dcn_M)$. В силу потенциальной неограниченности временного ряда, величину M можно выбирать динамически: M можно выбрать в тот момент, когда машина, в ходе торговли, сама по себе окажется в финальном состоянии. С другой стороны, если считать, что M - заранее фиксированная величина, то попасть в заключительное состояние можно, просто закрыв позицию в момент времени $M-1$.

Поведение описанной выше машины по сути задаётся только набором функций $\{dcn_n\}_{n=1}^{M}$, а также правилами перехода между состояниями машины (которые скрыты внутри ее реализации). В этой работе все функции dcn_i будут полагаться равными друг другу, в связи с этим называть их будем просто dcn.

Стоит отметить, что из приведённого определения машины *Robot* следует, что одна машина не может работать более чем с одним активом одновременно. Чтобы построить модель, работающую с произвольным количеством активов, необходимо создать несколько машин *Robot*, каждая из которых будет работать лишь с одним активом.

Таким образом, приведённое определение *Robot* не умаляет общности определения машины, поддерживающей работу с несколькими активами одновременно.

Помимо параметров описанных выше, которые являются частью состояния машины *Robot*, имеется несколько других величин, характеризующих полную конфигурацию машины.

1. *asset_type* - тип актива. В рамках данной работы может принимать одно из двух значений: либо ASSET, либо FUTURES. ASSET означает, что в качестве базисного актива рассматривается какой-либо долевой бездивидендный финансовый инструмент. FUTURES означает, что в качестве актива рассматривается фьючерсный контракт на произвольный финансовый инструмент. От поля *asset_type* зависит способ исполнения торговых поручений, а также схема расчёта комиссии, взимаемой биржей.

2. *history* - последовательность состояний машины, принимаемой ей в процессе торговли. После каждой совершенной торговой сделки, в это поле дописывается новое состояние машины, а также цена последней заключённой сделки. Поддержка поля *history* должна быть реализована в машине, поскольку по информации, сохранённой в этой последовательности, будут вычисляться характеристики торговой сессии.

3. *comission* - размер комиссии биржи. С каждой заключённой торговой сделки бирже уплачиваются комиссионные сборы. При этом, как уже упоминалось ранее, метод взимания комиссии зависит от типа актива, описываемого полем *asset_type*.

4. *delay* - временная задержка при исполнении торговых поручений (в миллисекундах). При торговле на реальных фондовых площадках между отправкой торгового поручения и его исполнением проходит некоторый промежуток времени (издержки передачи и обработки данных). Машина *Robot* реализует подобную задержку.

3.2. Ограничения модели

Машина *Robot* во многом моделирует роботов, осуществляющих торговую деятельность на фондовых биржах, однако имеется ряд особенностей, которые в данной модели не учитываются:

1. Не учитываются объёмы заключаемых сделок. Это означает, что в реальности в определённые моменты времени может возникать ситуация, при которой робот не сможет закрыть позицию по конкретной цене в полном объёме.

2. В машине *Robot* анализируются цены последней заключённой сделки, в то время как необходимо рассматривать цены спроса и предложения. Данное упрощение приводит к возникновению рисков дополнительных издержек, поскольку в реальности сделка заключается не по цене $last$, а по цене bid или ask в зависимости от типа открытой позиции.

3. В этой работе фьючерсы являются немаржируемыми финансовыми инструментами (как форварды), однако гарантийное обеспечение при открытии фьючерсных позиций все равно взимается [15].

Из перечисленных ограничений видно, что машина *Robot* в некотором приближении хорошо моделирует действительность, что даёт возможность строить торговые стратегии, оставаясь в рамках описанной модели.

3.3. Показатели эффективности торговой сессии

После построения нескольких торговых стратегий, необходимо ввести характеристики, которые бы позволили сравнивать стратегии друг с другом. Эффективность стратегии будет определяться средним значением эффективности нескольких торговых сессий.

Введем несколько характеристик, определяющих эффективность торговой сессии машины *Robot*:

- $AY \stackrel{\text{def}}{=} \frac{money_M - money_0}{money_0 \cdot k}$ - средняя доходность одной сделки за период $[t_0, ..., t_M]$, где k- количество заключённых сделок за торговую сессию. Также в качестве альтернативного показателя можно использовать величину $\frac{money_M - money_0}{(t_M - t_0)k}$.

- $R^2 \in [0, 1]$ - коэффициент детерминации кумулятивной доходности [3].

Введем функцию эффективности торговой сессии. $eff(AY, R^2) \stackrel{\text{def}}{=} log(AY + 1) \cdot R^2$, где log - натуральный логарифм. Величина $eff > 0 \Leftrightarrow AY > 0$. При этом чаще всего, когда $eff > 0$, обязательно $log(AY + 1) \in [0, 1]$, что делает эту величину сопоставимой с R^2. Заметим, что чем больше значение eff, тем эффективнее работа машины.

Приведённая формула расчёта эффективности торговой сессии отнюдь не является единственной. На самом деле, в качестве эффективности может выступать любая функция переменных AY, R^2, отражающая основные зависимости эффективности от её характеристик. Функция должна быть неубывающей по AY и по R^2.

Апробация торговой стратегии будет проводиться следующим образом: после проведения N торговых сессий, будут вычислены эффективности каждой из них, формируя при этом последовательность $\{eff_n\}_{n=1}^N$. Качество (эффективность) торговой стратегии будет определяться её математическим ожиданием и стандартным отклонением последовательности $\{eff_n\}_{n=1}^N$.

3.4. Формулировка задачи

Определив понятие качества (эффективности) торговой стратегии, появляется возможность сформулировать поставленную задачу. Требуется подобрать такую функцию принятия решения dcn, чтобы максимизировать математическое ожидание функции
$eff(AY, R^2)$. Формально, необходимо найти $\underset{dcn \in Strat}{\operatorname{argmax}} \mathbf{E}\ eff(AY, R^2)$, где $Strat$ - множество торговых стратегий (функций принятия решения).

Однако совершенно очевидно, что нахождение решения сформулированной выше задачи в реальности не представляется возможным (сам факт существования такого решения является сомнительным), поэтому в этой работе будет представлено несколько торговых стратегий (основанных на кластеризации), среди которых будет выбрана та, у которой математическое ожидание $eff(AY, R^2)$ будет наибольшим.

4. Кластеризация временных рядов

Кластеризация - один из наиболее популярных методов прогнозирования финансовых временных рядов. Существует множество различных методов кластеризации данных, некоторые из которых описаны в разделе 2.3. В этой главе будет описана одна из возможных методик кластеризации (применительно к задаче предсказания динамики изменения биржевых котировок), будет произведена разметка обучающего множества, будут сформированы различные вектора признаков, а также будет приведены наилучшие результаты работы метода.

4.1. Постановка задачи кластеризации финансовых временных рядов

Пусть $\{x_n\}_{n=1}^N$ - временной ряд котировок актива, $CL = \{-1, 0, 1\}$ - конечное множество номеров (идентификаторов) кластеров. Здесь -1 обозначен нисходящий тренд, 1 обозначен восходящий тренд, а 0 - боковой тренд. Пусть также имеется некоторая обучающая выборка $\{y_n\}_{n=1}^M$, предшествующая временному ряду $\{x_n\}_{n=1}^N$, для каждой точки которой известно, к какому тренду (кластеру) она принадлежит. Всякой точке временного ряда x_n (равно как и y_n) ставится в соответствие l-мерный вектор признаков $p(x_n) \in \mathbf{R}^l$. Далее в пространстве признаков вводится функция расстояния ρ, определяющая меру схожести пары точек временного ряда. Через $cl(x_n)$ (или $cl(p(x_n))$) будем обозначать номер кластера, к которому принадлежит точка x_n.

В данной работе для кластеризации использовался алгоритм kNN (k nearest neighbours). Схема его работы следующая: сначала производится предподготовка данных (нормализация, сглаживание и пр.), затем из обучающего множества формируется пространство признаков. Для всякой точки y_n вычисляется ее вектор признаков $p(y_n)$, помечаемый меткой $cl(y_n)$. Проделав данную процедуру для каждой точки обучающей выборки, формируется пространство векторов признаков, про каждый элемент которого известно, какому кластеру он принадлежит. Далее на вход алгоритму поступает временной ряд $\{x_n\}_{n=1}^N$, каждую точку которого необходимо отклассифицировать к одному из трёх трендов. Для классификации очередной точки x_n, находится k ближайших соседей вектора $p(x_n)$ в пространстве признаков (относительно функции ρ). Предположим, были выбраны векторы $\{q_i\}_{i=1}^k$, причём для каждого q_i известно $cl(q_i)$. Для определения тренда $cl(p(x_n))$ целесообразно применять метод взвешенного голосования:

$$cl(p(x_n)) \stackrel{\text{def}}{=} \underset{j \in \{-1,0,1\}}{\operatorname{argmax}} \sum_{\{i \in [1..k] \ | \ cl(q_i)=j\}} \frac{1}{\rho^2(p(x_n), q_i)}$$

Определив $j = cl(p(x_n))$, вектор $p(x_n)$ добавляется к пространству признаков с меткой принадлежности к кластеру j. После этого алгоритм применяется для клас-

сификации очередного элемента временного ряда x_{n+1}.

Следует отметить, что для реализации алгоритма kNN необходимо, во-первых, каким-то образом произвести разметку обучающего множества; во-вторых, нормализовать входные данные; в-третьих, требуется разработать функцию p, отвечающую за формирование вектора признаков и, наконец, выбрать оптимальную функцию расстояния между векторами признаков. Каждый из перечисленных этапов будет описан в последующих разделах.

4.2. Разметка трендов временных рядов

Для того, чтобы научиться решать задачу кластеризации финансовых временных рядов, необходимо уметь производить разметку трендов обучающего множества. Не существует единого подхода к разметке трендов временного ряда, поскольку само понятие тренда не является чем-то однозначно определяемым. В этом разделе будет приведено математически точное понятие тренда, а также будет разработан алгоритм, осуществляющий разметку трендов финансовых временных рядов.

4.2.1. Определение тренда

Пусть $\{x_i\}_{i=1}^N$ - временной ряд котировок, каждую точку которого необходимо отклассифицировать к одному из трёх трендов: к восходящему, нисходящему или боковому (-1 соответствует нисходящему тренду, 1 - восходящему, а 0 - боковому).

Определение: $\Delta(i,j) \stackrel{\text{def}}{=} \frac{x_j - x_i}{x_i}$

Определение: Пусть $\alpha \in \mathbf{R}_+, i, j \in [1..N]$. Тогда

$$trend_\alpha(i,j) \stackrel{\text{def}}{=} \begin{cases} -1, & \text{если } \Delta(i,j) < -\alpha \\ 0, & \text{если } -\alpha \leq \Delta(i,j) \leq \alpha \\ 1, & \text{если } \Delta(i,j) > \alpha \end{cases} \tag{6}$$

В частности, $trend_0(i,j) = 0 \Leftrightarrow x_i = x_j$ и $|trend_0(i,j)| = 1 \Leftrightarrow x_i \neq x_j$.

Утверждение: $\forall i, j \in [1..N]\ |trend_0(i,j) \cdot (x_j - x_i)| = |x_j - x_i|$

▷ Пусть $x_i \neq x_j$. Тогда $|trend_0(i,j) \cdot (x_j - x_i)| = |trend_0(i,j)| \cdot |x_j - x_i| = |x_j - x_i|$. Если же $x_i = x_j$, получаем тождество $0 = 0$. ◁

Определение: Будем говорить, что на отрезке времени $[t_i, t_j]$ (или для краткости на отрезке времени $[i,j]$) имеет место нисходящий / боковой / восходящий α-тренд, если $trend_\alpha(i,j) = -1/0/1$.

Функция $trend$ принимает значение 0 только в тех случаях, когда относительное изменение значения временного ряда за промежуток времени $[i,j]$ по модулю оказалось не более, чем α. Исходя из приведённого выше определения функции $trend$,

становится ясно, что α имеет смысл "пороговой" константы или "чувствительности" распознавания тренда.

Действительно, справедливо утверждение: $\forall i, j \in [1..N]\ \exists \alpha^* \in \mathbf{R}_+ : |trend_{\alpha^*}(i,j)| = 1$. Это, в частности означает, что для всякого отрезка $[i,j]$, на котором имел место боковой α-тренд, можно так уменьшить α, что на $[i,j]$ будет иметь место тренд нисходящий или восходящий.

Определение: Пусть $\{i_k\}_{k=1}^M$ - строго монотонно возрастающая подпоследовтельность $[1..N]$, $\alpha \in \mathbf{R}+$. Тогда

$$\omega_\alpha(\{i_k\}_{k=1}^M) \stackrel{\text{def}}{=} \sum_{k=1}^{M-1} |trend_\alpha(i_k, i_{k+1}) \cdot (x_{i_{k+1}} - x_{i_k})| \tag{7}$$

Функцию ω_α следует воспринимать в качестве функции "тяжести" или "полезности" подпоследовательности временного ряда. $trend$ в этом определении играет роль характеристической функции. Она переводит в 0 все незначительные колебания $x_{i_{k+1}} - x_{i_k}$, в то время как крупные колебания цены оставляет без изменения.

Определение: Пусть $\alpha \in \mathbf{R}+$. α-базисом временного ряда $\{x_i\}_{i=1}^N$ будем называть такую последовательность $\{i_k\}_{k=1}^M \subset [1..N]$, что $i_1 = 1 \wedge i_M = N \wedge \forall \{j_k\}_{k=1}^{M'} \subset [1..N] \Longrightarrow \omega_\alpha(\{i_k\}_{k=1}^M) \geq \omega_\alpha(\{j_k\}_{k=1}^{M'})$. При этом весом (или величиной) α-базиса будем называть $\omega_\alpha(\{i_k\}_{k=1}^M$.

Определение: Точка x_i (или для краткости i) называется α-базисной, если i принадлежит α-базису.

Иными словами α-базисом $\{x_i\}_{i=1}^N$ называется $\underset{\{i_k\} \in D}{\operatorname{argmax}} \omega_\alpha(\{i_k\})$, где D - множество всевозможных строго монотонно возрастающих подпоследовательностей $[1..N]$, содержащих точки 1 и N. α-базис состоит их тех элементов временного ряда, в которых происходит "переключение" между трендами. Для того, чтобы α-базис был всегда не пуст, в него добавляются крайние индексы временного ряда 1 и N. То есть любой временной ряд начинается и заканчивается базисной точкой.

Следует заметить, что, основываясь на определении (7), α-базис может быть не единственным. Пример: $\{x_i\} = \{10, 20, 30\}$, $\alpha = 1\%$. Рассмотрим всевозможных претендентов на роль α-базиса: $\{1,3\}, \{1,2,3\}$. Величина первого равна $|trend_{0.01}(1,3) \cdot (30-10)| = 20$, величина второго равна $|trend_{0.01}(1,2) \cdot (20-10)| + |trend_{0.01}(2,3) \cdot (30-20)| = 10 + 10 = 20$. Таким образом получается, что обе подпоследовательности являются 0.01-базисами.

Рассмотрим крайний случай, когда $\alpha = 0$. Докажем, что в этом случае $\{1, ..., N\}$ является 0-базисом $\{x_i\}_{i=1}^N$ (то есть каждая точка временного ряда $\{x_i\}_{i=1}^N$ является 0-базисной). Для доказательства этого факта потребуется доказать промежуточное утверждение:

Утверждение: Пусть $\{i_k\}$ - подпоследовательность $\{1, ..., N\}$, $1 \leq j \leq N$, $j \notin \{i_k\}$.

Тогда $\omega_0(\{i_k\}) \leq \omega_0(\{i_k\} \cup \{j\})$. (Здесь подразумевается, что j вставляется в $\{i_k\}$ таким образом, что получающаяся последовательность остается строго монотонно возрастающей).

▷ Пусть последовательность $\{i_k\} \cup \{j\}$ приняла вид $\{i_1, ..., i_p, j, i_{p+1}, ..., M\}$. Тогда $\omega_0(\{i_k\} \cup \{j\}) = \omega_0(\{i_k\}_{k=1}^{p}) + |trend_0(p, j) \cdot (x_j - x_p)| + |trend_0(j, p+1) \cdot (x_{p+1} - x_j)| + \omega_0(\{i_k\}_{k=p+1}^{M}) = \omega_0(\{i_k\}_{k=1}^{M}) + |trend_0(p, j) \cdot (x_j - x_p)| + |trend_0(j, p+1) \cdot (x_{p+1} - x_j)| - |trend_0(p, p+1) \cdot (x_{p+1} - x_p)| = \omega_0(\{i_k\}_{k=1}^{M}) + |x_p - x_j| + |x_{p+1} - x_j| - |x_{p+1} - x_p| \geq \omega_0(\{i_k\}_{k=1}^{M})$ (поскольку $|x_p - x_j| + |x_{p+1} - x_j| \geq |x_{p+1} - x_p|$ по неравенству треугольника). ◁

▷ Из доказанного выше утверждения следует, что если $\{i_k\} \subset \{j_k\}$, то $\omega_0(\{i_k\}) \geq \omega_0(\{j_k\})$. Но из этого следует, что $\{1, ..., N\}$ обязана быть 0-базисом, поскольку $\forall \{i_k\}_{k=1}^{M} \subset \{1, ..., N\} \Longrightarrow \omega_0(\{i_k\}_{k=1}^{M}) \leq \omega_0(\{1, ..., N\})$. ◁

Определение: α-разметкой временного ряда $\{x_i\}_{i=1}^{N}$ называется последовательность $\{c_i\}_{i=1}^{N}$, такая что $\forall j \in [i_k, i_{k+1})\ c_j = trend_\alpha(i_k, i_{k+1})$, где $\{i_k\} - \alpha$-базис временного ряда $\{x_i\}_{i=1}^{N}$.

Решением задачи классификации временного ряда (в контексте данной задачи) является нахождение его α-разметки (всякому x_i сопоставляется $c_i \in \{-1, 0, 1\}$).

4.2.2. Алгоритм α-разметки временного ряда

После того, как было получено определение понятие тренда и разметки временного ряда, необходимо разработать алгоритм, который всякому временному ряду $\{x_i\}_{i=1}^{N}$ и всякому $\alpha \in \mathbf{R}_+$ будет сопоставлять его α-разметку.

Определим функцию $basis_\alpha$ одного целочисленного аргумента. $basis_\alpha(k)$ - это α-базис временного ряда $\{x_i\}_{i=k}^{N}$. Заметим, что $basis_\alpha(N) = \{N\}\ \forall \alpha \in \mathbf{R}_+$, а также, что ответом на поставленную задачу является значение $basis_\alpha(1)$.

Пусть для некоторого $k \in [1..N]$ справедливо $\forall i > k$ известно значение $basis_\alpha(i)$. Научимся, обладая этой информацией, вычислять значение $basis_\alpha(k)$.

$$b1 = \{k\} \cup basis_\alpha(\operatorname*{argmax}_{k<i\leq N} \omega_\alpha(\{k\} \cup basis_\alpha(i)))$$

$$b2 = \{k\} \cup basis_\alpha(\operatorname*{argmax}_{k<i\leq N} \omega_\alpha(\{k\} \cup \overline{basis_\alpha(i)})) \tag{8}$$

$$basis_\alpha(k) = \operatorname*{argmax}_{b\in\{b1,b2\}} \omega_\alpha(b)$$

Под обозначением $\overline{basis_\alpha(i)}$ здесь понимается $basis_\alpha(i) \backslash \{i\}$.

Получив соотношение (8), появляется возможность описать алгоритм построения α-разметки $\{x_i\}_{i=1}^{N}$. Последовательно будем перебирать всевозможные k от N до 1 и вычислять соответствующее им значение $basis_\alpha(k)$ по формулам (8). Как уже упоминалось ранее, $basis_\alpha(1)$ является решением задачи поиска α-разметки.

Оценим сложность алгоритма. Алгоритм $N - 1$ раз вычисляет значение $basis_\alpha$,

руководствуясь формулами (8). Для вычисления $basis_\alpha(k)$ требуется $N-k$ раз вычислить значение функции ω_α, которое в общем случае вычисляется за $N-k$ операций. Таким образом, алгоритм должен будет совершить порядка $O(N^3)$ операций.

Существенным ускорением может служить вычисление ω_α за $O(1)$. Заметим, что в очередной раз ω_α вызывается от $\{k\} \cup basis_\alpha(i)$, а $\omega_\alpha(basis_\alpha(i))$ был уже вычислен на предыдущем шаге алгоритма. Это также можно понять, записав рекуррентное соотношение $\omega_\alpha(\{k\} \cup basis_\alpha(i)) = \omega_\alpha(\{k, i\}) + \omega_\alpha(basis_\alpha(i))$. Значение первого слагаемого вычисляется за $O(1)$, а значение второго было вычислено на предыдущем шаге. Таким образом, пересчёт ω_α возможен за $O(1)$ операций. Но это в свою очередь означает, что сложность описанного выше алгоритма α-разметки равна $O(N^2)$.

4.2.3. Применение алгоритма α-разметки к реальным данным

В качестве временного ряда $\{x_i\}_{i=1}^N$ были выбраны средневзвешенные по объёму котировки контракта RTS-6.15 с 11:00 до 12:00 22 апреля 2015 года. К этим данным применялся описанный выше алгоритм α-разметки для $\alpha \in \{0.1\%, 0.3\%, 0.5\%\}$. Далее строились графики для наглядного отображения полученной информации (см. рисунки 5, 6, 7).

Как уже упоминалось ранее, α влияет на чувствительность к распознаванию трендов. На представленных графиках это очень хорошо прослеживается: при $\alpha = 0.1\%$ смена трендов происходит очень часто, в то время, как при $\alpha = 0.5\%$ тренды сменяют друг друга относительно редко.

Опишем динамику выявления трендов в зависимости от параметра α. При $\alpha = 0$ 0-базисом является весь временной ряд (см. п.4.2.1), т.е. смена трендов происходит постоянно. При возрастании α количество выявленных трендов начинает уменьшаться и, начиная с некоторого значения α^*, выявляется ровно 1 боковой тренд. Вычислить значение α^* довольно просто: в качестве α^* можно выбрать любое число превосходящее величины $\frac{max-min}{min}$ и $\frac{min-max}{max}$, где $max = \max\limits_{1 \le i \le N} x_i$, $min = \min\limits_{1 \le i \le N} x_i$.

Определение: Разметкой временного ряда $\{x_i\}_{i=1}^N$ будем называть некоторую последовательность $\{c_i\}_{i=1}^N$, где $\forall i \; c_i \in \{-1, 0, 1\}$.

Очевидно, что всякая α-разметка временного ряда является его разметкой.

Определение: Пусть $\{c_i\}_{i=1}^N$, $\{d_i\}_{i=1}^N$ являются разметками временного ряда $\{x_i\}_{i=1}^N$ соответственно. Коэффициентом схожести разметок $\{c_i\}_{i=1}^N$, $\{d_i\}_{i=1}^N$ будем называть величину

$$\rho(\{c_i\}_{i=1}^N, \{d_i\}_{i=1}^N) \stackrel{\text{def}}{=} \frac{|\{i : c_i = d_i\}|}{N} \tag{9}$$

Если $\{c_i\}_{i=1}^N$, $\{d_i\}_{i=1}^N$ являются α_1, α_2-разметками, то их коэффициент схожести будем сокращённо обозначать $\rho(\alpha_1, \alpha_2)$. В таблице 1 приведены значения $\rho(\alpha_1, \alpha_2)$ для всевозможных α_1, $\alpha_2 \in \{0.1\%, 0.2\%, 0.3\%, 0.4\%, 0.5\%\}$ для временного ряда, рассматриваемого в этом разделе.

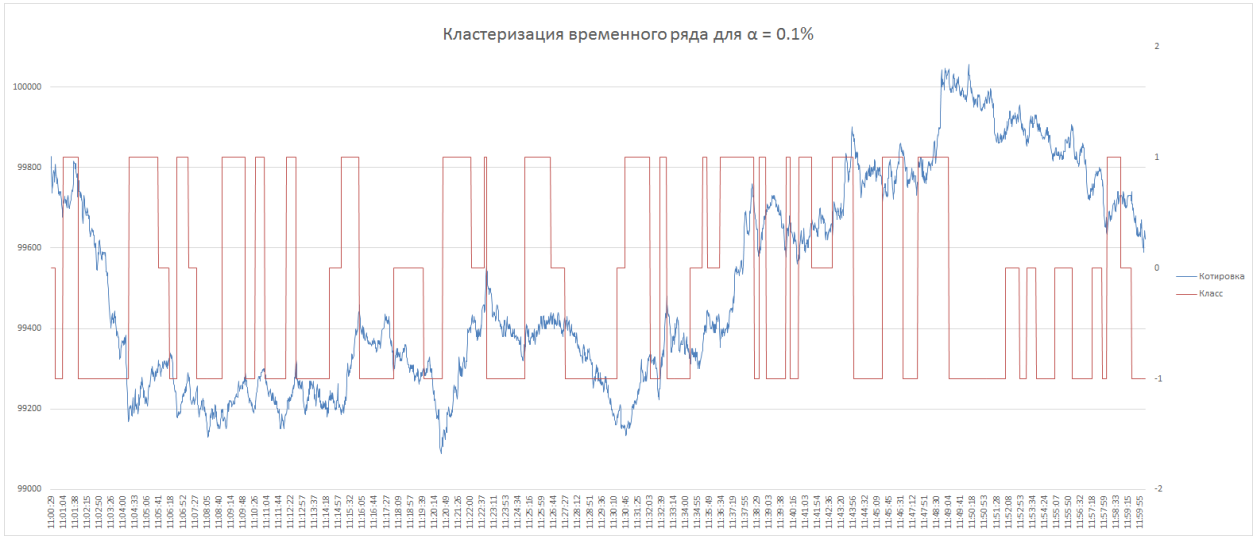

Рис. 6: 0.1%-разметка временного ряда RTS-6.15

Рис. 7: 0.3%-разметка временного ряда RTS-6.15

Рис. 8: 0.5%-разметка временного ряда RTS-6.15

Таблица 1: Коэффициент схожести α-разметок

α	0.1%	0.2%	0.3%	0.4%	0.5%
0.1%	1.00	0.51	0.504	0.473	0.355
0.2%	0.51	1.00	0.766	0.613	0.348
0.3%	0.504	0.766	1.00	0.639	0.406
0.4%	0.473	0.613	0.639	1.00	0.453
0.5%	0.355	0.348	0.406	0.453	1.00

Из Таблицы 1 видно, что разметки могут сильно отличаться друг от друга в зависимости от параметра α. Так, например, $\rho(0.2\%, 0.5\%) = 0.348$, что в свою очередь означает, что доля одинаково отклассифицированных точек равна всего 34.8%. Отсюда становится ясно, что результат α-разметки временного ряда и, как следствие, результат последующей кластеризации сильно зависит от выбранного α.

4.3. Предобработка данных

Предобработка данных является одним из важнейших этапов кластеризации. Практика показывает, что перед формированием пространства признаков, данные необходимо преобразовать к виду, удобному для обучения. Преобразование исходных данных необходимо для более качественной работы алгоритма кластеризации и во многом определяет его эффективность.

4.3.1. Нормализация данных

Рассмотрим пример, демонстрирующий необходимость нормализации данных. Пусть вектор признаков состоит всего из двух элементов: первым элементом будет являться усреднённое значение котировки актива (за некоторый промежуток времени), а вторым - вероятность наступления некоторого события в будущем (например, вероятность того, что в следующий момент времени котировка вырастет на несколько пунктов). Предположим также, что в пространстве признаков введено евклидово расстояние между векторами.

Ключевой проблемой приведённого примера является тот факт, что величины двух характеристик не сопоставимы друг с другом. Первая компонента вектора может принимать сколь угодно большие значения (допустим, в приведённом примере, котировка колеблется в окрестности 1000), а вторая компонента всегда принадлежит вещественному отрезку $[0, 1]$ (поскольку она отражает вероятность наступления определённого события). При классификации очередного элемента, "вклад в расстояние" вносимый второй координатой будет несущественен, поскольку её величина пренебрежимо мала по сравнению с первой.

Пусть в пространстве признаков имеется два вектора: $v_1 = (1000, 1)$ и $v_2 = (1003, 0)$, принадлежащие разным кластерам. На вход алгоритму kNN подаётся вектор $x = (1001, 0)$ при $k = 1$. $\rho(x, v_1) = 1^2 + 1^2 = 2$, $\rho(x, v_2) = 2^2 + 0^2 = 4$. Таким образом, вектор x находится ближе к вектору v_1 и попадёт к нему в кластер. В то же время, очевидно, что x должен быть отклассифицирован также как и вектор v_2, поскольку отклонение значение котировки не столь существенно по сравнению с тем, что второй элемент вектора принял диаметрально противоположное значение.

Для того, чтобы избежать подобных проблем, применяются различные методы нормализации данных. Одним из них является линейное преобразование всех компонент вектора к одному и тому же диапазону значений. В качестве такого диапазона может выступать, например, отрезок $[0, 1]$. Пусть $\{p_i\}_{i=1}^N$ - вектора признаков обучающего множества, $p_i \in \mathbf{R}^l$. Вычисляются вектора $mx, mn \in \mathbf{R}^l$, такие что $\forall j \in [1..l]$

$$mx(j) = \max_{i \in [1..N]} p_i(j)$$

$$mn(j) = \min_{i \in [1..N]} p_i(j)$$

Затем вычисляется последовательность $\{p'_i\}_{i=1}^N$, где $p'_i(j) = \frac{p_i(j) - mn(j)}{mx(j) - mn(j)}$ $\forall j \in [1..l]$. Каждый элемент вычисленной последовательности попадает в куб $[0, 1]^l$, что обеспечивает корректность последующей кластеризации.

Перед обработкой очередного элемента временного ряда, сначала проводится его нормализация и только затем классификация. Следует отметить, что после нормализации элемента не из обучающей выборки, результирующий вектор может не попасть в куб $[0, 1]^l$. Так может произойти, если одна из координат элемента временного ряда, оказалась больше максимума (или меньше минимума), зафиксированного среди элементов обучающей выборки. Теоретически, эта проблема может снижать качество кластеризации, однако на практике ее влияние незначительно. В этой работе применялся именно этот метод нормализации.

Другой способ нормализации данных производится следующим образом:

$$p(x_i)(j) = \frac{x_i(j) - E(j)}{\sigma(j)}$$

где $p(x_i)$ - вектор признаков элемента временного ряда x_i

$E(j) = \mathbf{E}\, p_i(j)$ - математическое ожидание j-ой координаты обучающей выборки

$\sigma(j)$ - стандартное отклонение j-ой координаты обучающей выборки

Данное преобразование известно тем, что оно преобразовывает нормально распределённую случайную величину в стандартно распределённую. Пусть $\xi \in N(E, \sigma^2)$, $\eta = \frac{\xi - E}{\sigma}$. Докажем, что $\eta \in N(0, 1)$: $\mathbf{E}\eta = \frac{1}{\sigma}(E - E) = 0$, $\sigma(\eta) = \frac{1}{\sigma}\sigma(\xi - E) = \frac{\sigma}{\sigma} = 1$.

Помимо описанных выше способов нормализации, существует множество других. Не существует универсального совета относительно выбора методики нормализации

данных - её выбор сильно зависит от решаемой задачи.

4.3.2. Сглаживание данных

Зачастую перед нормализацией финансовых временных рядов производится их сглаживание. Сглаживание нивелирует сильную локальную скачкообразность ряда, делая мгновенное изменение цены актива более плавным и "непрерывным". Эта процедура позволяет выделять тренды на принципиально более качественном уровне, поскольку многие шумы, неминуемо возникающие при работе с финансовыми временными рядами, в значительной мере ослабевают.

Наиболее популярным методом сглаживания является вычисление экспоненциального скользящего среднего EMA(α), которое было описано в п.1.5.1. Подбор соответствующего параметра α осуществляется эмпирическим путём, основываясь на данных обучающей выборки.

Однако более эффективным способом сглаживания данных признано т.н. двойное экспоненциальное сглаживание. Его отличие от одинарного состоит в введении дополнительного параметра, характеризующего трендовую составляющую временного ряда.

Пусть $\{y_i\}_{i=1}^N$ - исходный временной ряд, по которому необходимо получить его двойное экспоненциальное сглаживание. Также положим, что заданы некоторые константы α, γ, являющиеся параметрами сглаживания. $\alpha \in [0, 1), \gamma \in [0, 1]$. Сглаженный ряд $\{S_t\}_{t=1}^N$ вычисляется по следующей формуле:

$$\begin{cases} S_t = \alpha y_t + (1-\alpha)(S_{t-1} + b_{t-1}) & t \geq 2 \\ b_t = \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1} & t \geq 2 \\ S_1 = y_1 \\ b_1 = y2 - y1 \end{cases} \tag{10}$$

Существуют другие варианты определения значения b_1. Например, в качестве b_1 возможно рассматривать $\frac{(y_2-y_1)+(y_3-y_2)+(y_4-y_3)}{3}$ или $\frac{y_n-y_1}{n-1}$. Для прогнозирования будущих значений ряда через $n \geq 1$ шагов используется соотношение $S_{t+n} = S_t + nb_t$.

Вычисляемый таким образом сглаженный временной ряд, будем обозначать $DEMA(\alpha, \gamma)$. Параметры α, γ подбираются по данным обучающей выборки при помощи алгоритма Левенберга — Марквардта [16].

4.4. Формирование вектора признаков

После того, как в разделах 4.2, 4.3 были получены алгоритмы разметки обучающего множества и предобработки данных, появляется необходимость сформировать различные вектора признаков. В этом разделе будет представлено несколько способов

их формирования.

Здесь и далее будем обозначать через $\{x_i\}_{i=1}^N$ обучающую выборку, а через $\{y_i\}_{i=1}^T$ - временной ряд, каждую точку которого необходимо отклассифицировать в режиме реального времени (online). При прогнозировании биржевых котировок, помимо временного ряда цены актива, часто используется временной ряд объемов заключенных сделок (в количестве контрактов). Этот ряд будем обозначать через $\{v_i\}_{i=1}^T$. В тех случаях, когда по каким-либо причинам информацией о последовательности $\{v_i\}_{i=1}^T$ мы не располагаем, будем считать, что каждый ее элемент равен 1.

После выбора правила, по которому будут вычисляться вектора признаков, и функции расстояния, задающей меру похожести пары векторов, применяется алгоритм кластеризации kNN. Экспериментальным путем было установлено, что наилучшее качество распознавания трендов достигается при $k = 21$. В результате работы алгоритма kNN, применённого к временному ряду $\{y_i\}_{i=1}^T$, получаем последовательность $\{a_i\}_{i=1}^T$, $a_i \in \{-1, 0, 1\}$, являющуюся разметкой ряда $\{y_i\}_{i=1}^T$ (см. раздел 4.2.3). Далее применяется алгоритм α-разметки к $\{y_i\}_{i=1}^T$, получая при этом разметку $\{b_i\}_{i=1}^T$. Качеством распознавания будем называть величину $\rho(\{a_i\}_{i=1}^T, \{b_i\}_{i=1}^T) \in [0, 1]$ (см. соотношение 9 в разделе 4.2.3).

Качество алгоритма кластеризации (при заданном α, алгоритме сглаживания, нормализации данных, векторе признаков и функции расстояния) будет определяться двумя величинами:

- $\mathbf{E}(\rho(\{a_i\}_{i=1}^T, \{b_i\}_{i=1}^T))$ - математическое ожидание качества распознавания. Оно будет вычисляться как среднее арифметическое. Эту величину далее будем называть средним качеством алгоритма кластеризации.

- $\sigma(\rho(\{a_i\}_{i=1}^T, \{b_i\}_{i=1}^T))$ - стандартное отклонение качества распознавания, вычисляемое как квадратный корень выборочной дисперсии. Эту величину будем называть отклонением качества алгоритма кластеризации.

Следует обратить внимание, что в Таблице 1 раздела 4.2.3 приводилась в пример схожесть разметок обучающего множества для различных α. Из таблицы видно, что, например, для $\alpha_1 = 0.2\%$, $\alpha_2 = 0.5\%$ схожесть разметок составляет всего 34.8%. Это означает, что формируемые вектора признаков обязательно должны каким-то образом зависеть от величины α. Ведь если для некоторого α_1 построенный вектор признаков обеспечивает достаточно высокую эффективность распознавания, то, допустим, для $\alpha_2 = 3 \cdot \alpha_1$ алгоритм кластеризации, в основе которого лежит тот же вектор признаков, будет работать гораздо менее качественно.

Очевидно, что чем выше среднее качество алгоритма кластеризации и чем ниже его отклонение, тем более эффективным и надёжным можно считать построенный алгоритм.

4.4.1. Метод линейной регрессии

Основываясь на решении задачи линейной регрессии, было построено три различных вектора признаков:

1. Фиксируется некоторый момент времени k временного ряда $\{y_i\}_{i=1}^T$. Для него формируется одномерный вектор признаков $p_k \in \mathbf{R}$, единственной координатой которого является наклон регрессионной прямой, рассчитанный по процедуре $rrf(\{y_i\}_{i=1}^T, \{v_i\}_{i=1}^T, 1, 10)$ (подробно о процедуре rrf см. п.1.4). В качестве функции расстояния рассматривался модуль $|\cdot|$.

2. Для $k \in [1..T]$ формируется трёхмерный вектор признаков $p_k \in \mathbf{R}^3$. Расстояние между векторами признаков задавалось Евклидовым расстоянием.

$$p_k = \begin{pmatrix} rrf(\{y_i\}_{i=1}^T, \{v_i\}_{i=1}^T, 1, 10) \\ rrf(\{y_i\}_{i=1}^T, \{v_i\}_{i=1}^T, 3, 10) \\ rrf(\{y_i\}_{i=1}^T, \{v_i\}_{i=1}^T, 5, 10) \end{pmatrix}$$

3. Формируется пятимерный вектор признаков аналогично п2. Евклидово расстояние.
   $j \in [1..5]$, $p_k[j] = rrf(\{y_i\}_{i=1}^T, \{v_i\}_{i=1}^T, 2j-1, 10)$.

В п.2,3 каждая координата вектора представляет из себя коэффициент наклона регрессионной прямой. Однако чем больше номер координаты, тем более "глобально" вычисляется наклон. Например, для случая $p_k(5)$ вычислялись средневзвешенные по объёму значения котировок блоками по $2 \cdot 5 - 1 = 9$ секунд и только затем применялась регрессия к усреднённым данным. Такой подход позволяет улавливать тренды различных срочностей. Чем больше период усреднения, тем более "крупные" тренды выявляются алгоритмом.

Качество алогоритма кластеризации при выборе соответствующего пространства признаков представлено в Таблице 2 (отсутствие сглаживания данных, расчёт качества производился на 12 временных рядах в зависимости от параметра α).

Таблица 2: Качество алгоритма кластеризации

	Вектор №1			Вектор №2			Вектор №3		
α	0.02%	0.05%	0.10%	0.02%	0.05%	0.10%	0.02%	0.05%	0.10%
Среднее	49.9%	47.5%	39.6%	49.4%	49.3%	49.6%	49.2%	50.1%	48.7%
Отклонение	1.0%	2.8%	4.2%	4.2%	9.0%	8.6%	3.7%	6.1%	11.6%

Для вектора признаков №1 наилучший результат достигается при $\alpha = 0.02\%$. При росте α среднее качество начинает убывать в то время, как отклонение начинает

возрастать. Для вектора №2 средняя распознаваемость практически не изменяется (в зависимости от α) в то время как отклонение увеличивается в два раза при переходе от $\alpha = 0.02\%$ к $\alpha = 0.05\%$. У вектора №3 достигается наибольшее среднее качество при $\alpha = 0.05\%$, однако отклонение высоко по сравнению с вектором №1.

4.4.2. Историческая зависимость данных

В этом разделе будем считать, что ряд $\{y_i\}_{i=1}^{T}$ является непосредственным продолжением обучающей выборки $\{x_i\}_{i=1}^{N}$, т.е. в момент времени $N + i$ котировка актива равна y_i. Объединение этих рядов будем обозначать $\{x_i\}_{i=1}^{N+T}$.

Опишем процесс формирования вектора признаков $p_k, k \in [1..N + T]$. Осуществляется предобработка исходных данных при помощи двойного экспоненциального сглаживания (см. п.4.3.2) и их последующей нормализации (см. п.4.3.1). Параметры сглаживания выбираются экспериментальным путём в зависимости от обучающей выборки. Выбирается некоторый $\alpha \in \mathbf{R}_+$, и к обучающей выборке применяется алгоритм α-разметки обучающего множества (обозначим получаемую разметку через $\{c_i\}_{i=1}^{N}$), фиксируется некоторая величина $w \in \mathbf{N}$ (опытным путем установлено, что наилучшее качество достигается при $w = 24$) и строится вектор признаков p_k:

$$p_k = \begin{pmatrix} c_{k-w} \\ c_{k-w+1} \\ \cdots \\ c_{k-2} \\ c_{k-1} \end{pmatrix} \quad k \in [1..T]$$

Вычислив вектор p_k, запускается алгоритм kNN, возвращающий результат классификации $c_k = kNN(p_k)$. Затем величина c_k добавляется к уже полученной разметке и алгоритм продолжает свою работу для $k + 1$. Качество алгоритма кластеризации представлено в Таблице 3.

Таблица 3: Качество алгоритма кластеризации

α	0.05%	0.10%	0.20%
Среднее	45.3%	53.4%	62.1%
Отклонение	6.5%	10.4%	13.6%

Заметим, что в отличие от метода линейной регрессии (см. п. 4.4.1), данный подход обладает высокой степенью разбора качества, однако его среднее значение демонстрирует существенно более высокие показатели. Максимум достигается при $\alpha = 0.20\%$, что свидетельствует о том, что применение данного подхода целесообразно при выявлении более крупных рыночных тенденций.

4.4.3. Технические индикаторы

Многие торговые стратегии основываются на сигналах, поступающих от различных технических индикаторов (см. раздел 1.5). Возникает интерес сформировать вектор признаков, состоящий из технических индикаторов и подставить его в алгоритм кластеризации. В рамках этой работы был сформирован следующий вектор признаков:

$$p_k = \begin{pmatrix} DEMA(\alpha_1, \gamma, k) \\ EMACD_{\alpha_4}(\alpha_2, \alpha_3, k) \\ \%K(n, k) \\ SAR(sar, al, step, mx, k) \end{pmatrix}, k \in [1..T]$$

Величины $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \gamma, n, sar, al, step, mx, k$ выбирались динамически, основываясь на данных обучающей выборки. Характеристики качества полученного алгоритма кластеризации приведены в Таблице 4.

Таблица 4: Качество алгоритма кластеризации

α	0.05%	0.10%	0.20%
Среднее	44.2%	39.6%	34.9%
Отклонение	3.1%	4.2%	9.1%

Из приведённых выше данных можно сделать вывод, что алгоритм не применим для распознавания крупных трендов при $\alpha \geq 0.20\%$, а распознавание мелких трендов демонстрирует недостаточно высокое качество по сравнению с алгоритмом, основанном на методе линейной регрессии (см. п. 4.4.1).

4.4.4. Преобразованные технические индикаторы

Поскольку применение технических индикаторов в "сыром" виде не дало значимых результатов, появляется мысль преобразовать их значения к виду, который бы был более "удобен" для восприятия алгоритмом кластеризации.

Многие торговые стратегии используют технические индикаторы в качестве т.н. сигналов к покупке или продаже активов. Часто сигналами являются превышение или понижение индикатора относительно некоторой сигнальной линии. При этом в большинстве подобных моделей (особенно у трендовых индикаторов), у вырабатываемого сигнала имеется два состояния: сигнал к покупке и сигнал к продаже актива. При этом очевидно, что наличие сигнала не гарантирует, что цена пойдет в нужном направлении. Оно лишь сообщает о том, что с достаточно большой степенью вероятности цена изменится в соответствующую сторону.

Предлагается осуществить преобразование дискретной бинарной сигнальной модели к непрерывной. В качестве сигнала индикатора будет выступать вероятность того,

что индикатор предсказывает направление движения цены верно. Очевидно, что в тех случаях, когда индикатор превышает свою сигнальную линию более ярковыражено, вероятность корректности поступающего сигнала должна приближаться к единице, в то время как при колебании индикатора около сигнальной линии, вычисляемая вероятность должна быть относительно низкой.

Для каждого используемого технического индикатора будет вычислена вероятность корректности выработки им сигнала и именно из этих величин будут сформированы векторы признаков.

$$p_k = \begin{pmatrix} \frac{1}{2\pi} \cdot arctg(EMACD) \cdot \chi_{[0;+\infty)}(EMACD) \\ \frac{1}{2\pi} \cdot arctg(-EMACD) \cdot \chi_{(-\infty;0]}(EMACD) \\ f(\%K) \\ DEMA_{trend} \end{pmatrix}, \text{ где}$$

$$f(x) = \begin{cases} 0.0 & x \leq 0.3 \\ 2.0 \cdot (x - 0.8) + 1 & 0.3 < x < 0.8 \\ 1.0 & x \geq 0.8 \end{cases}$$

Рассмотрим преобразование индикатора EMACD. Первая компонента вектора признаков характеризует меру того, что в данной точке имеет место восходящий тренд. В тех случаях, когда $EMACD < 0$, первая компонента обращается в 0. Когда $EMACD > 0$ применяется преобразование арктангенсом. Арктангенс является монотонно возрастающей функцией и $arctg(\mathbf{R}_+) = [0, 1)$, что позволяет преобразовать величину EMACD к некоторому подобию вероятности. Вероятно, для обеспечения более качественной работы алгоритма распознавания, необходимо совершить линейное растяжение величины EMACD, однако это оставлено за рамками данной работы. Аналогично вторая компонента вектора характеризует "вероятность" нисходящего тренда.

Для преобразования стохастического осциллятора была выбрана функция f, подбиравшаяся экспериментальным путем. По-другому функцию f можно представить в виде $f(x) = \chi_{[0.3,+\infty)}(x) \cdot \min(2 \cdot (x - 0.8) + 1), 1.0)$. f является монотонно неубывающей функцией, вероятность "срабатывания" индикатора $\%K$ растёт прямопропорционально на отрезке $[0.3, 0.8]$.

В качестве пятой координаты берётся трендовая составляющая двойного экспоненциального сглаживания (см. п. 4.3.2). Характеристики качества работы алгоритма представлены в Таблице 5.

Данные, представленные в таблице, демонстрируют относительно низкое среднее качество и высокую степень его разброса. Возможно, это связано с не совсем уместным преобразованием индикаторов или с неудачным выбором индикаторов EMACD, %K для распознавания трендов временного ряда контракта RTS-6.15.

Таблица 5: Качество алгоритма кластеризации

α	0.02%	0.05%	0.10%
Среднее	34.1%	41.3%	33.3%
Отклонение	14.2%	14.5%	9.9%

4.4.5. Совмещение векторов

В этом разделе был сформирован вектор признаков, состоящий из наклона регрессионной прямой (параметры регрессии подбирались в зависимости от α на основе предыдущих исследований), преобразованных технических индикаторов (см. раздел 4.4.4), трендовой составляющей двойного экспоненциального сглаживания, а также исторической зависимости данных. Полученные характеристики качества кластеризации приведены в Таблице 6.

Таблица 6: Качество алгоритма кластеризации

α	0.02%	0.05%	0.10%
Среднее	47.2%	45.3%	41.4%
Отклонение	2.1%	3.5%	4.2%

Из полученных результатов можно сделать вывод, что совмещение описанных выше подходов не даёт увеличения качества распознавания трендов. Наилучший результат вновь демонстрируется при распознавании трендов для самых малых значений α.

4.5. Сравнение качества распознавания

Наилучшим подходом для распознавания краткосрочных трендов можно считать метод, основанный на решении задачи линейной регрессии (см. п.4.4.1). Наилучший результат достигается при $\alpha = 0.02\%$ для одномерного вектора состоящего из наклона регрессионной прямой, построенной по 10 точкам временного ряда. Среднее качество достигает значения 49.9%, а отклонение - 1.0%.

Для распознавания более крупных трендов следует использовать метод, основанный на исторической зависимости данных (см. п.4.4.2). Наилучший результат наблюдается при $\alpha = 0.20\%$: среднее качество принимает значение 62.1%, в то время как отклонение достигает 13.6%. Следует отметить, что, несмотря на большую степень разброса качества, этот подход демонстрирует наилучшую среднюю распознаваемость.

Методы, основанные на применении технических индикаторов, продемонстрировали недостаточно высокое качество распознавания трендов. Возможно, это связано

с неудачным выбором используемых индикаторов или неподходящей функцией расстояния.

Для повышения качества распознаваемости и уменьшения отклонения от среднего, необходимо прибегать к методам кластеризации множества временных рядов. Все временные ряды разбиваются на кластеры в соответствии с некоторыми их характеристиками, после чего к рядам из разных кластеров применяются различные алгоритмы распознавания трендов. Это позволит повысить качество прогнозирования финансовых временных рядов, а, следовательно, увеличит эффективность торговых стратегий.

5. Торговые стратегии

В главе 4 были получены алгоритмы кластеризации, позволяющие с некоторой степенью точности классифицировать каждую точку временного ряда к одному из трёх классов: к восходящему, нисходящему и боковому трендам. Основываясь на построенных алгоритмах, необходимо разработать несколько торговых стратегий, произвести их апробацию и сравнение друг с другом.

В главе 3 была построена модель, в рамках которой должно осуществляться тестирование торговых стратегий. В момент инициализации машины *Robot*, в качестве d_0 ей на вход подаётся обучающая выборка $\{y_i\}_{i=1}^{N}$, к которой впоследствии применяется алгоритм α-разметки для некоторого выбранного α. Затем, в соответствии с моделью, на вход машине последовательно подаются элементы временного ряда $\{d_i\}_{i=1}^{T}$. В момент времени i на вход машине поступает величина d_i, которая классифицируется алгоритмом кластеризации (заложенным в логику функционирующей торговой стратегии) к одному из трёх трендов. Результат классификации является основным сигналом для принятия решения о проведении операции купли-продажи актива.

Обобщённый алгоритм принятия торгового решения, рассматриваемый в данной работе, приведён на Рисунке 9.

На представленной блок-схеме d является очередным элементом временного ряда, c - идентификатор тренда, is_signal - функция, накладывающая дополнительные условия на принятие торгового решения (об этом подробнее речь пойдет далее), операции типа "открыть/закрыть короткую/длинную позицию" скрыты внутри реализации машины Robot (см. раздел 3.1). На выход подается одно целое число - количество контрактов, которое было куплено/продано в зависимости от знака возвращаемого значения.

В этой работе будет построено несколько алгоритмов, которые будут отличаться друг от друга только функцией is_signal. Функция is_signal принимает в качестве аргумента величину c_i - результат классификации точки d_i алгоритмом kNN. Функция зависит не только от c_i, но и от всего временного ряда $\{c_k\}_{k=1}^{i}$. Основной задачей этой функции является снижение количества ложных посылов на проведение торговых операций. Эта функция должна, например, не допускать открытие/закрытие позиций в тех случаях, когда восходящие и нисходящие тренды чередуют друг друга слишком часто, поскольку частая смена сигналов свидетельствует о высокой неопределённости в принимаемом решении.

В работе рассматривалось три функции is_signal:

- $is\_signal(c_i) \stackrel{\text{def}}{=} true$ - в этой реализации используется "сырая" кластеризация. Каждое сообщение от алгоритма kNN воспринимается как сигнал к покупке/продаже актива.

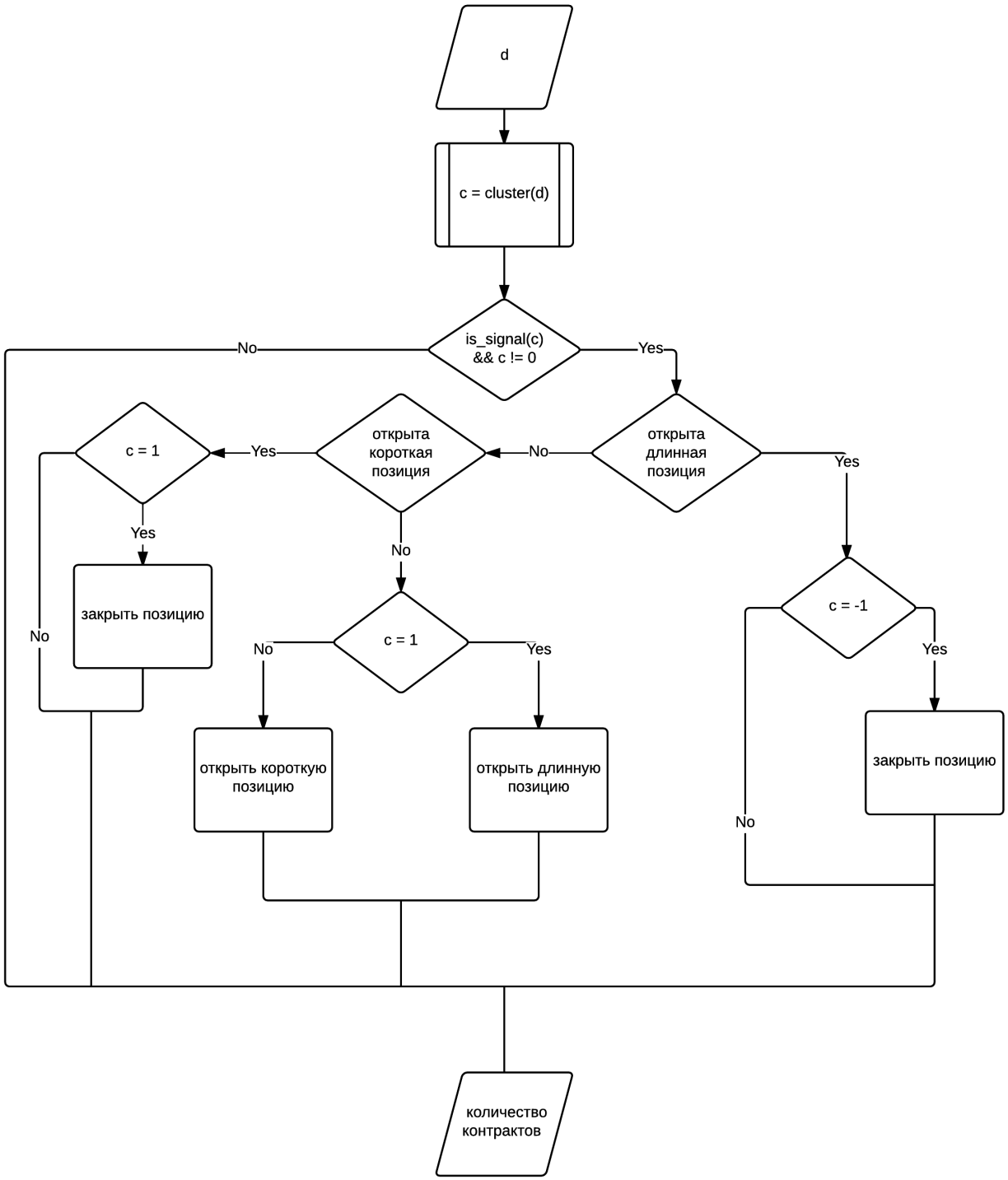

Рис. 9: Обобщённый алгоритм принятия торгового решения

- $is\_signal(c_i) \stackrel{\text{def}}{=} (c_i = 1 \wedge c_{i-1} = 1) \vee (c_i = -1 \wedge c_{i-1} = -1)$ - очередное сообщение от алгоритма кластеризации воспринимается как сигнал только в тех случаях, когда он продублирован дважды. При такой реализации функции is_signal, торговой стратегией игнорируются все сообщения алгоритма кластеризации, которые не подтвердились два раза подряд.

- $is\_signal(c_i) \stackrel{\text{def}}{=} (c_i = 1 \wedge c_{i-1} = 1 \wedge c_{i-2} = 1 \wedge c_{i-3} = 1) \vee (c_i = -1 \wedge c_{i-1} = -1 \wedge$

$c_{i-2} = -1 \wedge c_{i-3} = -1$) - аналогично пункту 2. Очередное сообщение от алгоритма кластеризации воспринимается в качестве сигнала только в том случае, когда три предыдущих результата классификации совпали с текущим.

В Таблице 7 приведены усреднённые характеристики каждой из трёх торговых стратегий при 0.05%-разметке (в качестве входных данных рассматривались секундные средневзвешенные по объёму котировки контракта RTS-6.15 за 1 час торгов).

Таблица 7: Характеристики торговых стратегий

	Стратегия №1	Стратегия №2	Стратегия №3
Средняя доходность одной сделки	0.0095%	0.0110%	0.0124%
Средний размер одной сделки	1.9	2.2	2.48
R^2	0.74	0.80	0.66
Количество заключённых сделок	427	266	166
eff (нормированный)	3.03	3.80	3.53

Стратегией с наибольшим коэффициентом R^2 является Стратегия №2. Величина *eff*, вычисленная в соответствии с определением, приведённом в разделе 3.3, соответствует динамике изменения величины R^2 - максимальное значение эффективности также достигается у Стратегии №2.

6. Заключение

В настоящей работе был разработан программный комплекс на языке C#, в рамках которого была реализована описанная в разделе 3.1 машина *Robot*, моделирующая торговых роботов, функционирующих на реальных фондовых биржах. Был создан класс *Robot*, соответствующий вышеописанной машине, основной причиной создания которого является возможность сокрытия внутри своей реализации всех технических деталей, связанных с проведением торговых операций на виртуальной фондовой бирже. При инстанцировании класса *Robot*, на вход конструктору подаётся набор параметров, задающих характеристики виртуальной биржи (временная задержка, размер комиссионных сборов и т.п.), а также алгоритм принятия торговых решений, определяющий дальнейшее поведение робота. Подобный подход позволяет разделить логику принятия торговых решений и механику проведения биржевых расчётов, позволяя перенести фокус исследований на разработку и апробацию торговых стратегий, основанных на методах кластерного анализа данных.

При построении алгоритмов кластеризации финансовых временных рядов был разработан алгоритм разметки трендов обучающего множества, были сформированы различные векторы признаков, а также проведено сравнение качества различных алгоритмов кластеризации. Все торговые стратегии строились на основе полученных алгоритмов кластеризации, в соответствии с схемой, представленной на рисунке 9 главы 5.

Результаты оценки качества (см. таблицу 7 главы 5) построенных торговых стратегий свидетельствует о применимости полученных алгоритмов для осуществления торговой деятельности на реальных фондовых биржах, при условии достаточно низких комиссионных сборов и умеренной временной задержки.

Список литературы

[1] EReport. Экономический рост, его показатели. — 2015. — URL: http://www.ereport.ru/articles/macro/macro09.htm (дата обращения: 09.05.2015).

[2] Golyandina N., Nekrutkin V., Zhigljavsky A. Analysis of Time Series Structure. SSA and Related Techniques. — Chapman & Hall/CRC, 2001.

[3] Gujarati D.N., Porter D.C. Basic Econometrics. Fifth Edition. — McGraw-Hill/Irwin, 2009.

[4] Heston S.L. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. — Yale University, 1993.

[5] Jorion P. Predicting Volatility in Foreign Exchange Market. — Journal of Finance, 1995.

[6] Morters P., Peres Y. Brownian Motion. — Cambridge University Press, 2008.

[7] Nogel U., Mikhailov S. Heston's stochastic volatility modelimplementation, calibration and some extensions. — Wilmott magazine, 2003.

[8] Saad E.W., Prokhorov D.V., Wunsch D.C. Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks. — IEEE, 1998.

[9] Wilder J. New Concepts in Technical Trading Systems. — Greensboro, NC, 1978.

[10] Боровиков В. STATISTICA - искуство анализа данных на компьютере. — Питер, 2003.

[11] Боровков А.А. Теория вероятностей. Второе издание. — Москва «Наука», 1986.

[12] Дефоссе Г. Фондовая биржа и биржевые операции. — Издательство "Феникс", 1992.

[13] Калуш Ю.А., Логинов В.М. Показатель Хёрста и его скрытые свойства. — Сибирский журнал индустриальной математики, 2002.

[14] Леваков А.А. Стохастические дифференциальные уравнение. — Белорусский Государственный Университет, 2009.

[15] Лобанов А.А, Чугунов А.В. Энциклопедия Финансового Риск-Менеджмента. — Альпина Паблишер, 2003.

[16] Осовский С. Нейронные сети для обработки информации. — Финансы и статистика, 2002.

[17] Полосухин И.Д. Динамическая кластеризация временных рядов с использованием агрегированных показателей. — "Вестник НТУ "ХПИ", Харьковский Политехнический Институт, 2011.

[18] Тищенко А.К., Плисс И.П. Сегментация многомерных нестационарных временных рядов с помощью метода нечеткой кластеризации. — Харьковский национальный университет радиоэлектроники, 2012.