

ГЕНЕРАЦИЯ ПРАВИЛ ТОРГОВОЙ СИСТЕМЫ С ПОМОЩЬЮ АНАЛИЗА ИСТОРИЧЕСКИХ ДАННЫХ ГЕНЕТИЧЕСКИМ АЛГОРИТМОМ

Искрич Д.П. студент кафедры информатики СПбГУ, iskrich@inbox.ru,
Григорьев Д.А. к.ф.-м.н, доцент кафедры информатики СПбГУ,
gridmer@yandex.com

Аннотация

В данной работе описывается торговый метод, основанный на применении бинарных диаграмм решения. Как известно, задача классификации данных является одной из наиболее известных в сфере data mining. В финансовой сфере, эта задача может интерпретироваться как извлечение торговых правил, исходя из исторических данных. Основная идея нашего подхода: создание оптимальных деревьев, по которым можно составить правила для входа в позицию (long) и выхода из неё (short).

Также будут представлены результаты работы алгоритма на примере акций IT компаний.

Введение

Любой инвестор стремится знать в какое время и при каких параметрах ему нужно покупать/продавать определённые инструменты. В сфере трейдинга существуют несколько подходов изучения рынка, в том числе - технический анализ, основанный на расчёте множества параметров. Сочетания параметров и их значений могут формировать правила при которых трейдер либо покупает инструмент, либо продаёт его.

Рассмотрим, например, следующий простой алгоритм торговли:

```
if A is HIGH then  
  | goBuy();  
else  
  | if B is LOW then  
    | goSell();  
  | else  
    | goBuy();  
  | end  
end
```

Нетрудно заметить, что правила для сигналов покупки/продажи могут быть представлены в виде списка булевских формул в конъюнктивной нормальной форме. В данном случае:

BUY(LONG) Signals

1. *A is HIGH*
2. $\neg(A \text{ is HIGH}) \wedge \neg(B \text{ is LOW})$

SELL(SHORT) Signals

1. $\neg(A \text{ is HIGH}) \wedge (B \text{ is LOW})$

Отметим ещё одно представление, в виде направленного ациклического графа [1], в данном случае мы будем иметь следующее дерево:

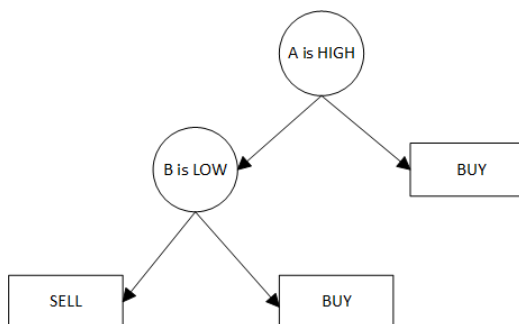


Рис. 1: Представление в виде дерева

Начиная с корня, мы проверяем предикат в узле, в случае истинности переходим к правому потомку, в противном случае к левому, в конечном итоге мы приходим к определённом сигналу. Представление правил в виде дерева решений удобно не только в визуальном контексте, но и в техническом, так как мы можем применять известные алгоритмы из теории графов для обработки правил.

Основная задача данной работы заключается в построении оптимального дерева решений с помощью генетического алгоритма. При этом, в качестве предикатов будут использованы различные технические индикаторы: MA, STO, FORCE.

Описание системы

Структура дерева

Узлы дерева могут представлять из себя *нетерминальные узлы*, включающие в себя предикат, и *терминальные*, листья которые могут принимать два значения *Buy/Sell*.

Предикат состоит из простого логического выражения:

if *INDICATOR is LEVEL*

В данной системе значение индикатора отображается в одно из 7 значений: [*ExtremelyLow*, *VeryLow*, *Low*, *Medium*, *High*, *VeryHigh*, *ExtremelyHigh*].

Индикатор принимает один из этих уровней относительно текущих максимальных и минимальных значений. При этом на каждом новом шаге минимальные и максимальные значения могут меняться.

Такая дискретизация обусловлена тем, что такие правила могут быть легко адаптированы на новые данные, опираясь лишь на максимальные/минимальные значения.

Индикаторы

Деревья в данной системе состоят из предикатов, которые содержат следующие индикаторы.

1. *MA* - скользящая средняя, один из самых популярных индикаторов, вычисляется по формуле:

$MA_t = \frac{1}{n} \sum_{i=0}^{n-1} C_{t-i}$ где n - скользящий интервал, C_{t-i} цена в период $t - i$

2. *STO* - Стохастический осцилятор, показывает текущее положение цены, относительно некоторого периода:

$STO_t = \frac{C_t - L_n}{H_n - L_n} \cdot 100$, где C_t - цена текущего периода, L_n - самая низкая цена за n периодов, H_n - самая высокая цена за n периодов.

3. *FORCE* - индекс силы, учитывает изменения цены, размер ценового движения и объем торгов:

$FORCE_t = Volume_i \cdot (C_t - C_{t-1})$, где $Volume_i$ текущий объем торгов, C_t - цена текущего периода.

Параметры генетического алгоритма

Общий принцип работы алгоритма представлен на рис. 2

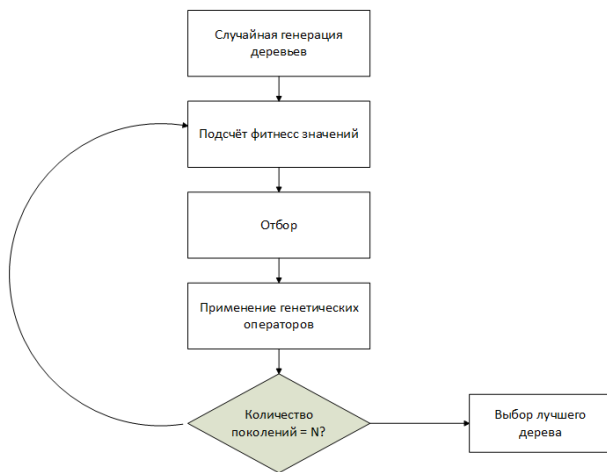


Рис. 2: Генетический алгоритм

1. *Начальная популяция* формируется случайным образом, предикат узла каждого дерева состоит из случайно выбранного наименования индикатора и случайно выбранного значения.
2. *Фитнесс функция* вычисляется посредством симуляции торговли на исторических данных, и представляет из себя коэффициент прибыльности в процентах.
3. *Отбор* происходит методом рулетки, у каждого дерева вероятность быть выбранным равна:
$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}, \text{ где } f_i \text{ фитнесс функция дерева } T_i$$
4. *Генетические операторы* построены по аналогии с [2] и состоят из оператора кроссовера и оператора мутации.

Оператор кроссовера принимает на вход двух родителей, которые тоже выбираются методом рулетки, и выдаёт двух потомков. Создание потомков происходит путём выбора случайного узла в обоих деревьях, далее происходит обмен поддеревьев начиная с этого узла.

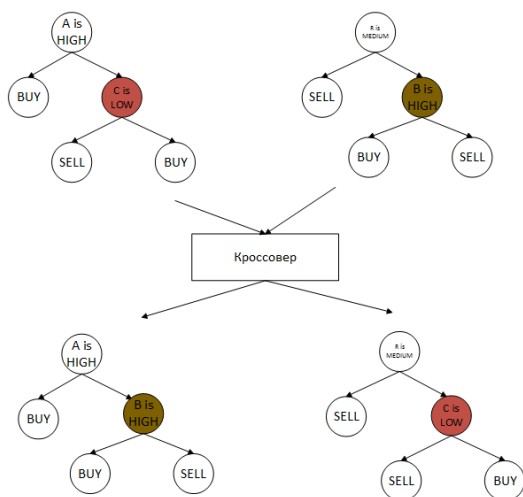


Рис. 3: Оператор кроссовера

Оператор мутации основан на том же принципе, в дереве выбираются случайно два узла и меняются местами.

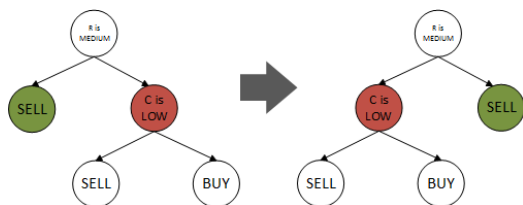


Рис. 4: Оператор мутации

Тестирование

Таблица 1: Параметры генетического алгоритма

Размер популяции	100
Количество поколений	20
Коэффициент кроссовера	0.25
Коэффициент мутации	0.1

Для тестирования системы были выбраны акции 4 компаний IT индустрии:

Таблица 2: Наиболее популярные IT компании

Название компании	Листинг на бирже
Apple, Inc.	AAPL
International Business Machines	IBM
Oracle Corporation	ORCL
Microsoft Corporation	MSFT

Период исторических данных 1 год: *30 января 2014 - 30 января 2015* с интервалом в 1 день. После нахождения наилучшего дерева на этом участке, тестируем это дерево за период: *30 января 2015 - 30 января 2016*.

Таблица 3: Тестирование системы

Символ	Прибыль на тренируемых данных	Прибыль на новых
AAPL	9.27%	18.86%
IBM	31.33%	18.73%
ORCL	16.92%	12.10%
MSFT	17.45%	13.21%

Заключение

В данной работе было дано описание системы, которая на основе генетического подхода, способна генерировать торговые правила. В основе правил лежат 3 индикатора, при этом система способна расширяться и использовать в качестве отправной точки гораздо большее число индикаторов.

Также система не исключает наличие в итоговом дереве решений и правил, определённых пользователем, основанных на пересечении близких по своей природе индикаторов.

Литература

- [1] Sheldon B. Akers. Binary Decision Diagrams // IEEE Transactions on Computers, June 1978.
- [2] Jean-Yves Potvin, Patrick Soriano, Maxime Vallée Generating trading rules on the stock markets with genetic programming //Computers & Operations Research 06/2004