はじめに
研究紹介
提案手法

研究紹介

木下大輔

富山県立大学　電子情報工学科

February 10, 2021

研究概要

はじめに
研究紹介
提案手法

研究概要

取引プラットフォーム (MetaTrader5) で取得した tidk データからヒストリカルデータを作成し、各インジケータのパラメータの最適化を行った後、直行表と最適化パラメータを用いてバックテストを行うことで評価指標を導く。それらから各インジケータの主効果を求め、全てのインジケータの組み合わせで評価指標の予測値を求める。それらの予測値を用いてデータ包絡分析を行い各ルールごとの効率値を算出する。それらを比較し最適なルールを選択し実際に自動売買に利用するシステムの構築。

引継ぎ状況

はじめに
研究紹介
提案手法

現在の最新のプログラムは動いていることが確認できる。

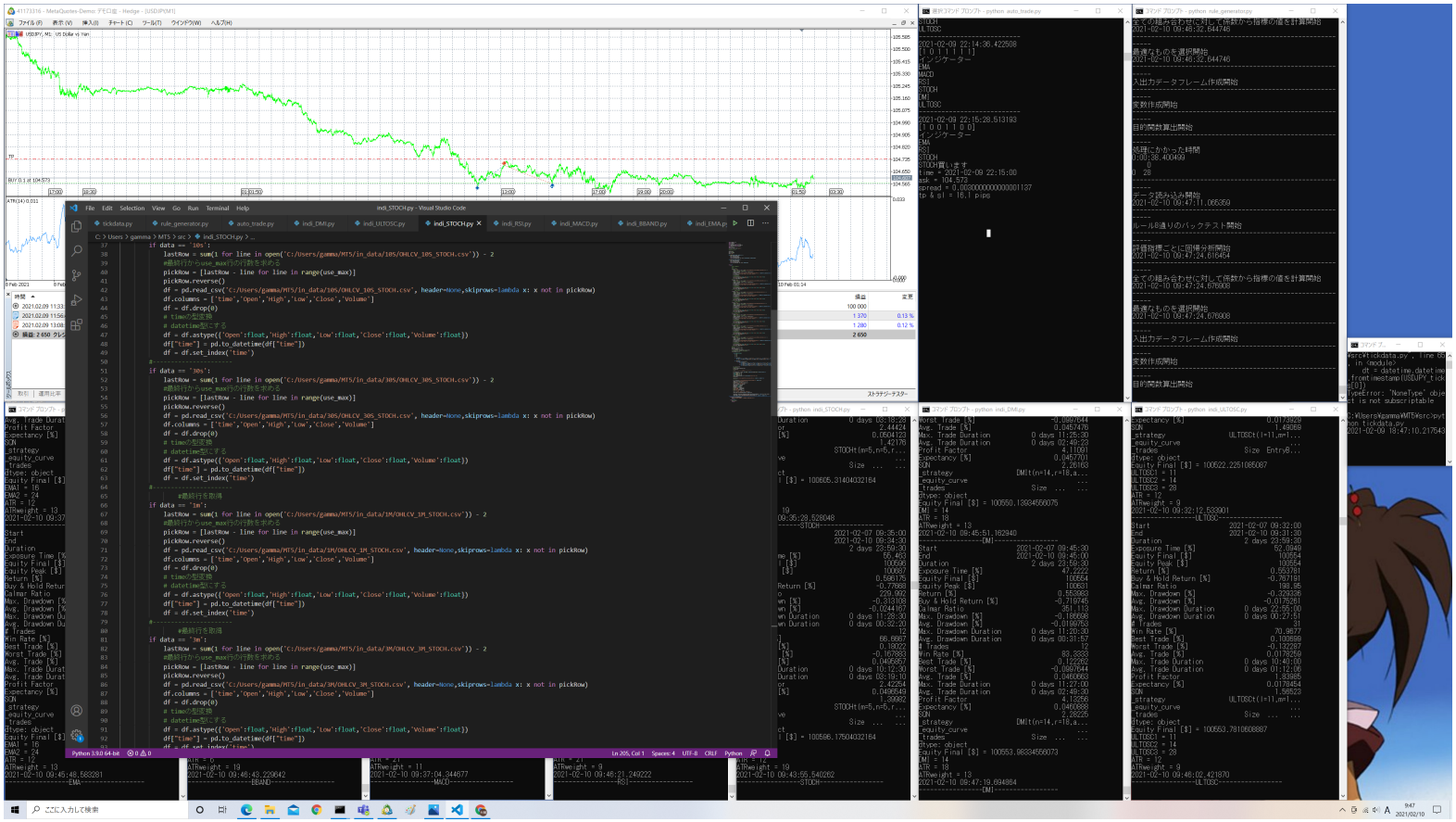

図 1: 現在動かしているプログラム

現在の活動

はじめに
研究紹介
提案手法

行ったこと

- 使っていない 8 個のインジケータのプログラムを現在のプログラムで動かせるように変更。
- 現在のプログラムは指定してある 10 秒足でのリサンプリングしているが、それらを他の秒足でもリサンプリングできるようにプログラムを変更。
- チャート上で利確と損切りとオーダーに入った際の価格の表示。

```
def next(self):
    #buy if
    if self.atr_ema * self.atrweight_ema >= 0.03:
        if (self.ema1[-2] < self.ema2[-2] and self.ema1[-1] > self.ema2[-1]):
            self.buy(limit=self.data['Close'], tp = self.data['Close'] + (self.atr_ema * self.atrweight_ema) + (spread * 0.01), sl = self.data['Close'] -
    if self.atr_bband * self.atrweight_bband >= 0.03:
        if (self.bband_lower[-2] > self.data.Close[-2] and self.bband_lower[-1] < self.data.Close[-1]):
            self.buy(limit=self.data['Close'],tp = self.data['Close'] + (self.atr_bband * self.atrweight_bband) + (spread * 0.01), sl = self.data['Close'
    if self.atr_macd * self.atrweight_macd >= 0.03:
        if (self.macd[-2] < self.macdsignal[-2] and self.macd[-1] > self.macdsignal[-1] and self.macd[-1] < 0 and self.macdsignal[-1] < 0):
            self.buy(limit=self.data['Close'],tp = self.data['Close'] + (self.atr_macd * self.atrweight_macd) + (spread * 0.01), sl = self.data['Close']
```

図 2: 変更した部分

はじめに
研究紹介
提案手法

課題

1. **インジケータの種類**
   インジケータの数が増えるごとに処理にかかる時間が増えるため、現在はインジケータの種類を減らしている。

2. **処理時間**
   インジケータの種類が増えると現在よりも処理にかかる時間が増えてしまう。どのように処理時間を短縮するか。

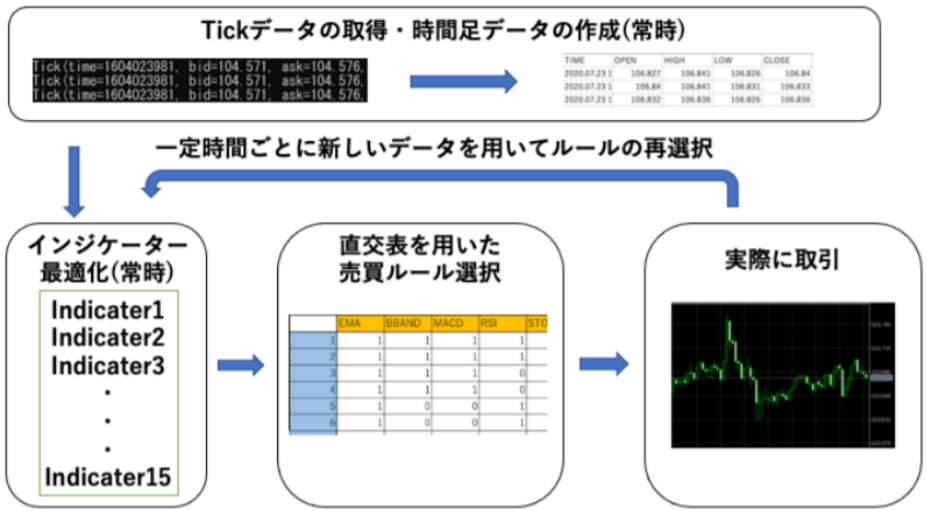

図 3: 取引の様子

提案手法

はじめに
研究紹介
提案手法

提案手法

- インジケータの種類によって増える処理時間を減らすためのプログラムの並列処理。
- SNS 分析の適用

処理に大きく時間をかけてしまう部分に並列処理を行うことで現在のプログラムよりも処理時間を短縮することができ、また使用するインジケータの種類が増えることでよりたくさんのルールの効率値を求めることができ現在のプログラムよりも有意性を持ったシステムを構築することが出来るのではないかと考える。テクニカル分析では予測できない動きをファンダメンタル分析を用いることで予測できるようになりさらに有意性が上がるのではないかと考える。

はじめに
研究紹介
提案手法

SNS 分析

SNS の口コミなどが市場に与える影響を調べることで市場の変動を予測することが出来るのではないかと考える。全てのユーザに対して分析を行うことは不可能なので特に市場に影響を与えるアカウントなどを分析して彼らから市場の予測を予測することが出来るのではないかと考える。

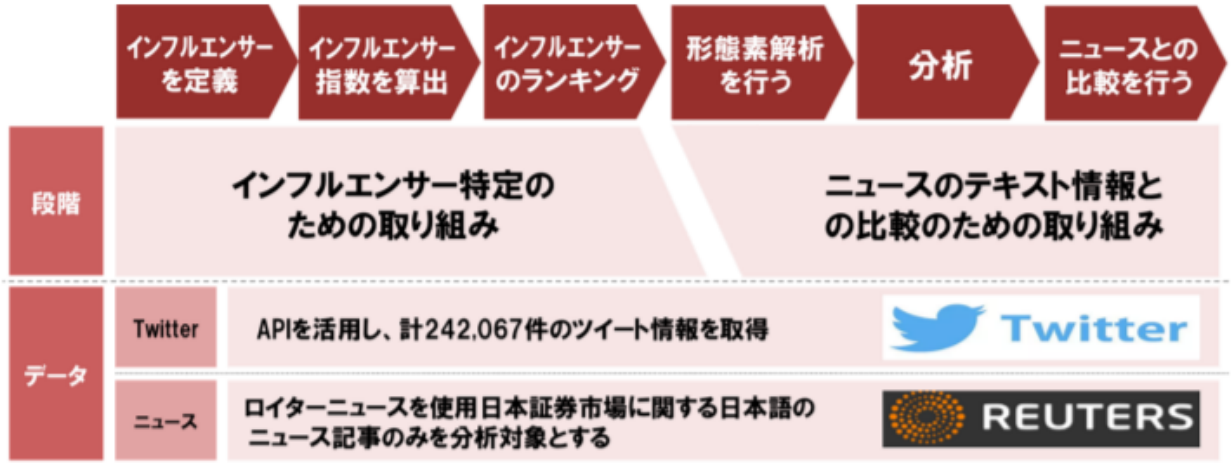

図 4: SNS 分析の流れ

並列処理

concurrent.futures モジュールを用いることでプログラム内の処理を並列化処理をすることが出来る。そのため簡単な計算プログラムを作成し処理時間を測ってみたところ並列化処理による処理時間の短縮をすることが出来た。このことから自動売買プログラムの各インジケータの主効果を求める部分や、組み合わせの評価指標の予測値の算出などでも処理時間を短縮することが出来るかもしれない。

```
C:¥Users¥gamma¥succession>python single_cal.py
処理時間:5.602s

C:¥Users¥gamma¥succession>python multi_thread.py
処理時間:1.422s
```

図 5: 処理時間の違い

はじめに
研究紹介
提案手法

分散処理

クラウドコンピューティングを使用して処理時間を短縮することも可能ではないかと考える。今回のプログラムであれば各評価値や予測値などの算出にクラウドコンピューティングを利用して計算資源を確保すればインジケータの種類が増えた場合でも処理時間を短縮することが出来ると考える。

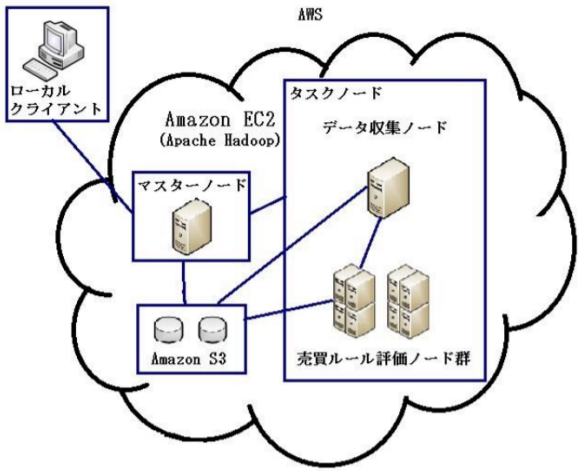

図 6: クラウドコンピューティング