

MetaTrader4or5 と Python による自動売買

大谷 和樹

富山県立大学 情報基盤工学講座

1. はじめに
2. インジケータ
3. 進捗状況
4. おわりに

October 1, 2020

背景

MetaTrader5 (MT5) と Python で、様々なインジケータを利用して自動売買を行うシステムの開発を行っている。研究の背景や目的などについては、しっかりとした文にまとめることができていないため、今後まとめていく必要がある。

目的

自動売買で勝てるシステムを作る。

使用するライブラリ TA-Lib

動作確認

TA-Lib を使ったインジケータの計算が実際に行うことができるのかを確認めるために、まずは過去データを用いてインジケータの計算を行った。

今回は任天堂の過去データを使用した。

```
# 単純移動平均(SMA: Simple Moving Average)
output = np.c_[output, talib.SMA(close)]
cols += ['SMA']

# 加重移動平均(WMA: Weighted Moving Average)
output = np.c_[output, talib.WMA(close)]
cols += ['WMA']

# 指数移動平均(EMA: Exponential Moving Average)
output = np.c_[output, talib.EMA(close)]
cols += ['EMA']
```

Figure: TA-Lib のコード

はじめに
インジケータ
進捗状況
おわりに

```

1 単純移動平均(SMA: Simple Moving Average)
2 加重移動平均(WMA: Weighted Moving Average)
3 指数移動平均(EMA: Exponential Moving Average)
4 2重指数移動平均(DEMA: Double Exponential Moving Average)
5 3重指数移動平均(TEMA: Triple Exponential Moving Average)
6 三角移動平均(TMA: Triangular Moving Average)
7 Kaufmanの適応型移動平均(KAMA: Kaufman Adaptive Moving Average)
8 MESAの適応型移動平均(MAMA: MESA Adaptive Moving Average)
9 トレンドライン(Hilbert Transform - Instantaneous Trendline)
10 ボリンジャー・バンド(Bollinger Bands)
11 MidPoint over period
12
13 変化率(ROC: Rate of change Percentage)
14 モメンタム(Momentum)
15 RSI: Relative Strength Index
16 MACD: Moving Average Convergence/Divergence
17 APO: Absolute Price Oscillator
18 PPO: Percentage Price Oscillator
19 CMO: Chande Momentum Oscillator
20
21 ヒルベルト変換 - Dominant Cycle Period
22 ヒルベルト変換 - Dominant Cycle Phase
23 ヒルベルト変換 - Phasor Components
24 ヒルベルト変換 - SineWave
25 ヒルベルト変換 - Trend vs Cycle Mode
26
27 60日単純移動平均
28 15日ボリンジャー・バンド
29 21日RSI

```

Figure: 計算したインジケータ

はじめに

インジケータ

進捗状況

おわりに

Date	Original	EMA	WMA	EMA	DEMA	TEMA	TMA	KAMA	MAMA	FAMA	HT_TREND	BBANDS_upper	BBANDS_mid	BBANDS_lower	MIDPOINT	ROC	MOV	RSI	MACD	MACD_signal	MACD_hist	APO	PPO
2007/3/4	25960																						
2007/3/5	28180																						
2007/3/6	28610																						
2007/3/7	29600																						
2007/3/11	30710																						
2007/3/12	31350											31024.5275	29390	27755.4725									
2007/3/15	31550											31543.0613	29668	27582.5387									
2007/3/16	32360											32102.1097	30222	27141.8980									
2007/3/17	32080											32041.5172	30690	28738.4228									
2007/3/18	32110											32745.4257	31680	30454.5743									
2007/3/19	32150											32664.7185	31880	31135.2815									
2007/3/20	32150											32594.0602	31920	31245.908	0.0514087	1540							
2007/3/22	32350											32704.82	32080	31455.18	0.1240448	3570							
2007/3/23	32350											32654.6427	32040	31445.3573	0.13118204	3740							
2007/3/24	32600											32007.0768	32160	31281.9211	0.17919193	4090							
2007/3/25	33000											33086.1625	32510	31153.8375	0.1905120521	2890	70.7259953						
2007/3/26	34000											34306.1825	33010	31683.8375	0.21880124	2790	72.8246631						
2007/3/29	34250											34862.4462	33410	31957.5538	0.18657946	2790	73.30275						
2007/3/30	34850											35159.0154	33810	32468.9848	0.277516772795	2590	75.2699221						
2007/3/31	35450											35764.0298	34420	33075.9162	0.16084024	3490	77.1874652						
2007/4/1	34950											35718.7616	34710	33703.2384	0.08709176	2890	72.3119253						
2007/4/2	34350											35640.8636	34770	33889.1384	0.06047619	2890	66.932231						
2007/4/3	32950											35219.2888	34510	32800.7132	0.01854714	600	56.3934305						
2007/4/4	32950											35182.2954	34120	32547.1884	0.02215504	450	56.4588914						
2007/4/7	320450											35435.8298	33520	31604.1712	0.0166687	-550	52.9891773						
2007/4/8	33960											34742.1131	33320	31887.8888	0.0134284	450	60.728431						
2007/4/9	33250											34691.9677	33100	32108.0233	0.0024449	-800	56.0741861						
2007/4/12	32950											34656.2725	32840	31884.1275	0.0447153	-1690	52.3899502						
2007/4/14	320400											34741.1151	32940	31822.8849	0.07130313	-2490	50.8811395						
2007/4/15	31450											34416.1883	32740	31683.8375	0.03450	-4090	45.4978751						
2007/4/16	32200	32370	32462.1505	32370								33865.4148	32380	31214.5852	0.0786838	-2750	49.8821314						
2007/4/19	32350	32449.6467	32462.8602	32168.7587								33203.388	32210	31394.612	0.0482342	-2590	50.8530711						
2007/4/20	31710	32547	32571.4444	32325.5457								32778.2403	32010	31264.7117	0.0450	-1250	47.8213863						
2007/4/21	31400	32643.3233	32638.4544	32385.8531								32506.1443	31820	31083.8557	0.04595927	-1590	45.3783334						
2007/4/22	32400	327156.6887	32822.7957	32274.5077								327195.8753	32010	31224.1247	0.0015408	-90	51.5947971	281.154143	790.012017	-508.85787	-526.92308	-1.5990662	
2007/4/23	33860	32860	32896.1075	32372.9244								33868.0159	32330	30671.9807	0.0044183	-150	58.7399935	362.499995	704.541593	-341.8817	-519.20077	-1.5725102	
2007/4/26	33850	32916.6447	32942.3454	32414.6488								34224.537	32470	31077.443	0.0045	-290	54.1233339	367.957103	636.144023	-975.58986	-603.84615	-1.5243193	
2007/4/27	32040	32944	32888.0323	32415.5372								34200.4716	32610	31019.5284	0.007657	-290	60.4388932	306.4906	570.213058	-263.72306	-467.62827	-2.017707	

Figure: インジケータの計算結果

自動売買

平松くんが作成していたプログラムの中で、売買ルールに関係している部分を自身が作ったもの書き換え、実際に動かした。
今回は単純移動平均 (SMA) を用いたゴールデンクロス・デッドクロスで売買を決定している。

- Close のかわりに Tick を使用
- 計算に使用する期間は、短期 100・長期 200 に設定

ゴールデンクロスとデッドクロス

ゴールデンクロスとは、短期の単純移動平均線が、長期の単純移動平均線を下から上に突き抜けるケースのこと。買いシグナルの1つ。
デッドクロスは、逆に短期の単純移動平均線が、長期の単純移動平均線を上から下に突き抜けるケースのこと。売りシグナルの一つ。

ゴールデンクロス（買いシグナル）

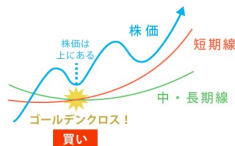


Figure: ゴールデンクロス

デッドクロス（売りシグナル）



Figure: デッドクロス

実行結果

8/14

はじめに

インジケータ

進捗状況

おわりに

2020.07.29 11:39...	usdjpy	667678093	sell	0.1	104.861			2020.07.29 11:48:21	104.870	-90	-0.01 %
2020.07.29 11:59...	usdjpy	667694211	sell	0.1	104.912			2020.07.29 12:19:37	104.916	-40	-0.00 %
2020.07.29 12:30...	usdjpy	667721093	sell	0.1	104.910			2020.07.29 12:39:48	104.944	-340	-0.03 %
2020.07.29 12:43...	usdjpy	667733078	sell	0.1	104.921			2020.07.29 12:52:27	104.913	80	0.01 %
2020.07.29 13:00...	usdjpy	667747724	sell	0.1	104.930			2020.07.29 13:03:22	104.925	50	0.00 %
2020.07.29 13:07...	usdjpy	667754121	sell	0.1	104.934			2020.07.29 13:16:29	104.966	-320	-0.03 %
2020.07.29 13:22...	usdjpy	667768285	sell	0.1	104.933			2020.07.29 13:25:53	104.940	-70	-0.01 %
2020.07.29 13:27...	usdjpy	667772102	sell	0.1	104.945			2020.07.29 13:47:47	104.998	-530	-0.05 %
2020.07.29 13:48...	usdjpy	667792139	sell	0.1	104.989			2020.07.29 13:51:12	105.004	-150	-0.01 %
2020.07.29 13:53...	usdjpy	667795674	sell	0.1	104.996			2020.07.29 14:04:42	104.992	40	0.00 %
2020.07.29 14:14...	usdjpy	667819137	sell	0.1	105.019			2020.07.29 14:22:23	104.995	240	0.02 %
2020.07.29 14:29...	usdjpy	667836351	sell	0.1	104.974			2020.07.29 14:33:55	104.979	-50	-0.00 %
2020.07.29 14:39...	usdjpy	667845620	sell	0.1	104.935			2020.07.29 14:50:51	105.027	-920	-0.09 %
2020.07.29 15:13...	usdjpy	667885674	sell	0.1	105.155			2020.07.29 15:24:41	105.107	480	0.05 %
2020.07.29 15:26...	usdjpy	667901119	sell	0.1	105.087			2020.07.29 15:30:53	105.078	90	0.01 %
2020.07.29 15:41...	usdjpy	667916174	sell	0.1	105.077			2020.07.29 16:08:02	105.037	400	0.04 %
2020.07.29 16:23...	usdjpy	667967928	sell	0.1	104.976			2020.07.29 16:30:31	105.061	-850	-0.08 %
2020.07.29 16:37...	usdjpy	667989406	#667967928, エキスパートによる注文		105.001			2020.07.29 16:43:35	105.062	-610	-0.06 %
2020.07.29 16:49...	usdjpy	668006108	sell	0.1	105.022			2020.07.29 16:55:24	105.045	-230	-0.02 %
2020.07.29 16:56...	usdjpy	668014184	sell	0.1	105.038			2020.07.29 17:00:08	105.006	320	0.03 %
2020.07.29 17:08...	usdjpy	668031000	buy	0.1	104.964			2020.07.29 17:20:25	104.987	230	0.02 %
2020.07.29 17:45...	usdjpy	668081224	buy	0.1	104.942			2020.07.29 18:10:25	105.071	1 290	0.12 %
2020.07.29 18:15...	usdjpy	668129712	buy	0.1	105.085			2020.07.29 18:46:09	105.029	-560	-0.05 %
2020.07.29 19:00...	usdjpy	668173792	buy	0.1	105.070			2020.07.29 19:03:16	105.055	-150	-0.01 %
2020.07.29 19:05...	usdjpy	668178404	buy	0.1	105.038			2020.07.29 19:09:07	105.000	-380	-0.04 %

Figure: SMA を用いて自動売買した結果

残高：97,788JPY → 95,148JPY
結果：-2640

指摘事項 1

インジケータの計算をする際に使用するデータの値は一定の間隔ごとに取得しないといけない

修正 1

MT5 からとってきた tick データをどんどん取得していく中で、1 秒ごとにその時点で最新の tick データ resample をして ohlc の形で別のデータフレームに保存していくようにした。これにより、一定時間間隔の ohlc のデータフレームを作成することができた。秒足の部分に関しては指定できるので、任意の時間感覚で resample することができる。

はじめに
インジケータ
進捗状況
おわりに

time					
2020-10-01	11:57:39	105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01	11:57:39	105.519	105.519	105.519	105.519
2020-10-01	11:57:40	105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01	11:57:39	105.519	105.519	105.519	105.519
2020-10-01	11:57:40	105.519	105.519	105.519	105.519
2020-10-01	11:57:41	105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01	11:57:39	105.519	105.519	105.519	105.519
2020-10-01	11:57:40	105.519	105.519	105.519	105.519
2020-10-01	11:57:41	105.519	105.519	105.519	105.519
2020-10-01	11:57:42	105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01	11:57:39	105.519	105.519	105.519	105.519
2020-10-01	11:57:40	105.519	105.519	105.519	105.519
2020-10-01	11:57:41	105.519	105.519	105.519	105.519
2020-10-01	11:57:42	105.519	105.519	105.519	105.519
2020-10-01	11:57:43	105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01	11:57:39	105.519	105.519	105.519	105.519
2020-10-01	11:57:40	105.519	105.519	105.519	105.519
2020-10-01	11:57:41	105.519	105.519	105.519	105.519
2020-10-01	11:57:42	105.519	105.519	105.519	105.519
2020-10-01	11:57:43	105.519	105.519	105.519	105.519
2020-10-01	11:57:44	105.519	105.519	105.519	105.519

Figure: 1 秒足の ohlc データの取得結果

指摘事項 2

データフレームに入れ続けることでデータが溢れてしまう

修正 2

データフレームの中身がある程度の大きさになったら古いものから削除していくようにする。実際には、どんどん取得されてくる tick データをデータフレーム 1 に格納していき、その中から 1 秒ごとに resample して別のデータフレーム 2 に格納している。データフレーム 1 の中身をどんどん削除していけばデータフレーム 2 も対応して減っていく。

		open	high	low	close
time					
2020-10-01 10:42:24		105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01 10:42:24		105.519	105.519	105.519	105.519
2020-10-01 10:42:25		105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01 10:42:24		105.519	105.519	105.519	105.519
2020-10-01 10:42:25		105.519	105.519	105.519	105.519
2020-10-01 10:42:26		105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01 10:42:25		105.519	105.519	105.519	105.519
2020-10-01 10:42:26		105.519	105.519	105.519	105.519
2020-10-01 10:42:27		105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01 10:42:25		105.519	105.519	105.519	105.519
2020-10-01 10:42:26		105.519	105.519	105.519	105.519
2020-10-01 10:42:27		105.519	105.519	105.519	105.519
2020-10-01 10:42:28		105.519	105.519	105.519	105.519
		open	high	low	close
time					
2020-10-01 10:42:26		105.519	105.519	105.519	105.519
2020-10-01 10:42:27		105.519	105.519	105.519	105.519
2020-10-01 10:42:28		105.519	105.519	105.519	105.519
2020-10-01 10:42:29		105.519	105.519	105.519	105.519

Figure: データフレームの中身を削除していく例

はじめに

インジケータ

進捗状況

おわりに

オリジナリティ

先生と相談した結果，マルチンゲールやココモ式，ポートフォリオ選択等のオペレーションズ・リサーチのシステムを組み込んでいく方針にした．この中でどれを組み込むのかはそれぞれ調査した結果決定する．

今回の進捗

- ① 一定時間間隔の ohlc のデータフレームを作成する
- ② データフレームの中身がある程度の大きさになったら古いものから削除していくように変更
- ③ 上記2つを組み込んで動かしてみたら売買が行われなかった& 日付変更時にエラーで止まってしまった

今後の課題

- 1 動かなかった原因の調査
- 2 秒足の ohlc のデータをローソク足として描画
- 3 インジケータの最適化
- 4 オペレーションズ・リサーチに関して調査