

日线 CCI 的计算与展示

CCI 指标，又称顺势指标，是美国股市技术分析家唐纳德·蓝伯特(Donald Lambert)于 20 世纪 80 年代提出的。该指标专门测量股价、外汇或者贵金属交易是否已超出常态分布范围，属于超买超卖类指标中较特殊的一种。该指标波动于正无穷大和负无穷大之间。但是，它不需要以 0 为中轴线，这一点也和波动于正无穷大和负无穷大的指标不同。

它最早是用于期货市场的判断，后运用于股票市场的研判，并被广泛使用。与大多数单一利用股票的收盘价、开盘价、最高价或最低价而发明出的各种技术分析指标不同，CCI 指标是根据统计学原理，引进价格与固定期间的股价平均区间的偏离程度的概念，强调股价平均绝对偏差在股市技术分析中的重要性，是一种比较独特的技术指标。

按照指标分析的常用思路，CCI 指标的运行区间也分为三类：+100 以上为超买区，-100 以下为超卖区，+100 到-100 之间为震荡区。在+100 到-100 之间的震荡区，该指标基本上没有意义，不能够对大盘及个股的操作提供多少明确的建议，因此它在正常情况下是无效的。

有时候 CCI 指标还是比较好使的。以最近比较活跃的乐视网为例，在股价大幅上涨的时候，CCI 指标有剧烈的波动。下面给出获取股票数据以及计算 CCI 并且画图的 python 代码，希望能抛砖引玉。

首先需要下载 [baostock](#) 和 TA-lib 的 python 库。

默认已经安装 python，且版本是 3.5 或者以上。

安装 pandas: `pip install pandas`

安装 numpy: `pip install numpy`

安装 baostock 数据接口包: `pip install baostock`。

如果有问题，请去官网下载再安装: <http://www.baostock.com>

安装 TA-Lib 库 `pip install talib`

如果失败，可以下载 whl 文件在本地安装。

```
import baostock as bs
import matplotlib.pyplot as plt
import matplotlib.font_manager as matfont
import matplotlib
import datetime
import numpy as np
import talib as ta
import pandas as pd

myfont = matfont.FontProperties(fname=r'C:/Windows/Fonts/msyh.ttf')
zhfont1 =
matplotlib.font_manager.FontProperties(fname='C:/Windows/Fonts/simkai
.ttf')

def get_his_k_data(stockcode = 'sh.600000'):
```

```
bs.login()
# 详细指标参数, 参见“历史行情指标参数”章节
rs = bs.query_history_k_data(stockcode,
    "date,code,open,high,low,close,preclose,volume,amount,pctChg",
    start_date='2015-01-01', end_date='2018-09-13',
    frequency="d", adjustflag="2")
print (rs.error_code)
#### 打印结果集 ####
data_list = []
while (rs.error_code == '0') & rs.next():
    # 获取一条记录, 将记录合并在一起
    data_list.append(rs.get_row_data())
result = pd.DataFrame(data_list, columns=rs.fields)
print (result)
bs.logout()
return result

def plot_two_curve_line(tradingDateList,y1,y2,titletext =
'pic',y1name=u'y1',y2name =u'y2'):
    """根据日期, 画出相同时间序列的两个值的曲线"""

    #将不同量纲的两个值转化为相似的数量级
    multi_num = max(y1)/max(y2)
    y2 = [y*multi_num for y in y2]

    x1 = range(len(tradingDateList))

    datelable = []
    for i_days in range(len(tradingDateList)):
        tradingdate = tradingDateList[i_days]
        date = int(tradingdate[8:])
        #     print date
        #     if date %10 == 0:
        if date == 1:
            datelable.append(tradingdate)
        else:
            datelable.append("")

    x1 = np.array(x1)
    y1 = np.array(y1)
    y2 = np.array(y2)

    fig, ax = plt.subplots()
```

```
plt.xticks(x1,datelable,rotation=30)
ax.plot(x1,y1, color = 'r')
ax.plot(x1,y2, color = 'y')
plt.title(titletext,fontproperties=zhfont1)
plt.legend((y1name,y2name),prop=zhfont1)

#显示在右上角
ax.legend(loc=1)

plt.show()

def plot_ASI_close_pic(stockcode = 'sh.600000'):
    """计算cci的值并且与收盘价的波动进行比较"""
    hisdata = get_his_k_data(stockcode)
    highlist = hisdata['high'].astype('float')
    lowlist = hisdata['low'].astype('float')
    closelist = hisdata['close'].astype('float')
    datelist = hisdata['date']
    CCIlist = ta.CCI(highlist,lowlist,closelist)
    closelist = list(closelist[-100:])
    CCIlist = list(CCIlist[-100:])
    datelist = list(datelist[-100:])
    plot_two_curve_line(datelist, closelist, CCIlist, "%s日线收盘价和
CCI历史曲线"%stockcode, "close", "CCI")

if __name__ == '__main__':
    plot_ASI_close_pic('sz.300104')
```

画出图像如下：

