

knowledge

This technical document will guide you through researching and developing a quantitative trading strategy in . A trading algorithm is nothing but a computer program with pre-defined rule sets to execute trade orders. It h ALGOGENE supports the use of third party libraries utilizing different academic theories and state of the art al ALGOGENE provides strategy backtest and development over various Commodities, Cryptos, Equities, Equity I Here is to

explain

how the

'CONTRA

CT SIZE'

and

'SETTLE

CURRENC

Y' is

related

to the

calculatio

n of

margin

amount

and

profit-

and-loss

under

backtest

environm

ent as

well as in

real

trading.

Suppose

we have

ALGOGENE's Python IDE is where you develop your trading idea. The platform is architected in an event stream

In the design, ALGOGENE strived to make everything as simple as possible while preserving all essential components. The reason of doing this is to lower the barrier for people from different backgrounds to get into the field of

Once you have come up with a trading strategy, the first step to backtest it on ALGOGENE is to initialize the ba

Strategy
Name: A
strategy
name
simply
provides
a high
level
descriptio
n for a
strategy
backtest.
It doesn't
have to
be
unique,
or it can
even be
set to an
empty
string.
Neverthel
ess, it is
strongly
encourag
ed to
give a
meaning

Start Period: You are required to specify the starting month for backtest. In case you have multiple instrument

End

Period:

Similarly,

you are

also

required

to specify

the end

month

for

backtesti

ng, which

must be

at least

equal to

or

greater

than the

start

period.

Any

invalid

setting

will by

default

set to the

same as

start

Data
Interval:
Data
interval
refers to
the
frequency
of data
feed
used for
backtesti
ng. The
higher
frequency
you
choose,
the more
accurate
trading
results it
will
generate
d, but in
the
compens
ation of
higher
computat

Initial
Capital:
Initial
capital is
the initial
investme
nt
amount
in dollar
value
you use
to
execute a
trading
strategy.
For
strategy
backtest,
to
simplify
the
complexit
y of
capital
managem
ent,
ALGOGEN
F

Base
Currency:
Base
currency
refers to
the
account
currency
unit for
your
specified
initial
capital
amount.
For each
trade
order,
the profit
and loss
will be
converted
from the
settle
currency
to the
base
currency.

Leverage:

Different
markets/
brokers/
countries
have
different
restrictio
ns on the
maximum
leverage
one can
take.

Therefore

,

ALGOGEN

E

provides

a

flexibility

to set the

leverage

ratio

subject

to your

own risk

appetite.

In

Allow
Short
Sell: In
real
world,
there
might be
difficultie
s or
restrictio
ns on
short sell
over
certain
markets.
This
variable
is
introduce
d to
better
simulate
the real
situation.

If it is set
to False,
any

Position
Netting:
It is for
system
handling
for
partial
close.

Suppose
you open
3
offsetting
trades:

#1: buy
2 share
of ABC at
\$123

#2: sell
1 share
of ABC at
\$124

#3: sell
1 share
of ABC at
\$125

Transaction Cost:
To better simulate the trading performance in real market, you can specify the per order transaction cost based on your knowledge and experience. It will be expressed in an all-in amount to cover

Risk free
rate: Risk-
free rate
is the
minimum

guaranteed interest
rate that
you
could
receive if
your
capital is
alternativ
ely
invested
on a risk-
free
asset (eg.
US
treasury
bond,
bank
saving
rate,
etc).
System

instruments: It is necessary to select at least 1 instrument to include in backtest. The available instrument symbols can be referred to 'Data Source' section.

For Options/Futures, subscribing an instrument

Benchmark: It compares how your strategy performs relative to a benchmark 'buy-and-hold' strategy of a selected index or underlying.

Enable
News
Stream:
It
provides
news
feed
from
various
global
publisher
s in
multiple
languages
.

When
news
data feed
is on, the
backtest
instance
will
additional
ly listen
to
historical
News

Enable
Economic
s Stream:
This data
stream
provides
fundame
ntal
statistics
in
country-
or city-
wide
including
GDP,
Inflation
Rate,
Unemplo
yment
Rate, etc.
It
supports
analysis
from
economic

perspecti
ves

Enable
Weather
Stream:
Some
financial
markets
such as
commodi
ty usually
found to
have
cyclical
or
seasonal
effect.
This data
stream
provides
city-wide
weather
status
such as
temperat
ure,
humudity
, wind
speed,
etc

Under
[My
History]
section,
all your
previous
backtest
results
and
outstandi
ng
running
tasks can
be found.

For any
non-
finished
oustandin
g task(s),
you can
either
'Load' to
view its
latest
simulatio
n image,
or

After
initializing
the setup for
backtest
environment, you
are now
prepared
to get
into the
core part
to code
up your
strategy.
ALGOGEN
Event
handlers
are
basically
server-
client
callback
functions,
where
your
client
script will

Stream
Market
Data
Feed:
The API
function
'on_marketdatafeed'
allows
you to
get price,
volume,
sensitivity
, and
order
book
data for
the
subscribed
instruments. This
API is a
First-In-
First-Out
(FIFO)
data
stream

·ma`
object
include
following
attribute:
instrume
nt,
expiry,
right,
strike,
minTick,
symbol,
timestam
p,
bidPrice,
askPrice,
midPrice,
highPrice,

lowPrice,
lastPrice,
volume,
bidSize,
askSize,
bidOrder
Book,
and
askOrder

The keys
in the
'ab'
object
are
realizedP
L,
unrealize
dPL,
cumDepo
sit, NAV,
marginUs
ed, and
available
Balance.
The
realizedP
L key
represent
s the
total
realized
Profit
and Loss
of all
closed
trades,
represent

Bulk Data

Feed:

There is another way to assess market data via 'on_bulkdatafeed'.

From this function, all your subscribed instrument(s) will be presented in a JSON/Python dictionary object 'bd', and you can extract all of

on_bulkdatafeed'
has very
similar
attributes
as
'on_mark
etdatafeed', except
that it
has to be
assessed
with a
directory
key
method.
The first
key for
'bd' is
your
subscribed
symbol,
while the
second
key for
'bd[your_
symbol']

More
important
ly, for
multiple
instrument
subscription
under
non tick
data
environment
(say
1 hour
bar), it is
common
that
different
instruments
may
have
different
trading
hours. To
check
whether
the
market
datasets

News
Data
Feed:
ALGOGEN
E News
database
is
sourced
from
200+
different
channels
globally,
including
BBC,
CNN,
Reuters,
etc. The
news
context
also
covers
various
languages
in
English,
Chinese,
Japanese

The 'nd'
object in
the
'on_news
datafeed'
function
include
following
keys:
published

(datetime
) , source
(string),
category
(string),
title
(string),
authors
(list), text
(string),
top_imag
e (string),
movies
(list), link
(string),
and lang
(string)

Weather
Data
Feed:
ALGOGEN
E

weather
database
collected
real-time
weather
details
from
over
200+
cities and
regions
globally.

Weather
event
data can
be
accessed
via
callback
function
'on_weat
herdatafe

The 'wd' object in the 'on_weat' herdatafe ed' function includes following keys: timestamp, city, country, coord_lat,

coord_lon, sunrise, sunset, visibility, pressure, temperature_min, temperature_max, temperature, humidity, wind speed

Economic
s
Statistics
Data
Feed:
ALGOGEN
E
Economic
s
database
collected
over
10,000
economic
time
series
across
different
countries
and
regions
globally.
There are
100,000+
economic
s time
series in
the

The 'ed' object in the 'on_econs datafeed' function includes following keys: timestamp, series_id, title, tag, seasonal_adj, popularity, y, obs_start, src, geo, freq, date, units, and value. Here are the descriptions of the

Corporate

Announcement:

For stock market, a corporate action is an event carried out by a company that materially impacts its stakeholders. These actions include the payment of dividends, stock split, mergers

The 'ca'
object in
'on_corp
Announce
ment'
function
include
following
attributes
: symbol
(string),
event
(string),
announce
_date
(datetime
),
ex_date
(datetime
),
payable_
date
(datetime
),
dividend_
amt
(float),
is_special

Order
Feed:
Immediately after
submitting an
order event
'self.event.s
endOrder
(orderObj
)', you
will
receive a
system's
message
from 'def
on_orderf
eed(self,
of)'
telling
you
whether
your
order has
been
executed
successfully

The 'or'
object in
the
'on_order
feed'
functions
include
following
attributes
:
insertTim
e,
tradeID,
openclos
e,
orderRef,
market,
broker,
productty
pe,
instrume
nt,
expiry,
right,
strike,
buysell,
fill_price,
and

Daily PL
Feed:
Under
ALGOGEN
E's
backtest
environm
ent,
there is a
tab 'PnL'
showing
the daily
profit
and loss
of your
trading
algorithm
. On the
other
hand,
whenever
market
data
stream
comes to
a new
date, you
can also

PnL
details in
'on_daily
PLfeed'
can be
extracted
from a
dictionary
key
method.
'pl'
object
include
following
keys:
Acdate
and
TotalPL.
Here are
the
descriptio
n of the
keys:

Acdate:
datetime.
The
accountin

Moreover
, you can
access
the PnL
breakdown
and the
day
range
PnL by
symbol
from
'on_daily
PLfeed'.
It can be
extracted
using a
dictionary
key
based on
'your_symbol'.
'pl[your_symbol]'
includes
following
keys:
FixedPL,
FloatPL

Position
Feed:
The trade
position
details
can be
found in
tab
'Position',
which is
presente
d in a
daily
basis. On
the other
hand,
whenever
there is
position
updated
due to
successful
orders
opening/
closing/
cancel,
you can
get

op`
object
can be
accessed
through
a
dictionary
key
method.
The first
key for
'op' will
be your
subscript
ed
symbol,
while the
second
key for
'op[your_
symbol]'
has the
following
attributes
:
LastTrade
Time and
netVolum

Secondly,
the full
outstandi
ng 'oo'
object
can be
accessed
through
a
dictionary
key
method.
The first
key for
'oo' must
come
from one
of the
system
generate
d
tradeID,
while the
second
key for
'oo[trade
ID]'
include

Lastly
and
similarly,
the full
list of
unfilled
limit/stop
order
'uo'
object
can be
accessed
through
a
dictionary
key
method.
The first
key for
'uo' is a
system
generate
d
tradeID,
while the
second
key for
'uo|trade|

Performance
Report:
Under
the
'Report'
tab, you
can find
a high
level
summary
on how
your
trading
strategy
perform
over the
backtesti
ng
period.
Summary
statistics
in the
report
include:

Tradable
Days:

In addition to the summary statistics, the relative strength of a strategy in terms of activeness, prediction, robustness, consistency, profitability, and recovery will be presented in a radar chart

Opening
or closing
trade
orders in
ALGOGEN
E starts
from
creating
an object
'AlgoAPIU
til.OrderO
bject()'.
Then, the
order will
be
submitte
d
through
function
'self.evt.s
endOrder
(orderObj
)'. The
impleme
ntation
can be
referred
to

In order
to define
a valid
order
object,
there are
several
elements
required
to be set.
'AlgoAPIU
til.OrderO
bject()'
includes
following
attributes
:
instrume
nt,
tradeID,
orderRef,
openclos
e,
buysell,
ordertype
, volume,
price,
expiry

Debugging: During your algorithm development, you might probably encounter any kind of coding errors. In case of this, you will be popped with a warning or error message after submitting the script, and you can go to the 'Console'

The
function
'self.evt.update_order(...)' is
used to
update
the
attached
parameters of an
opened
order.
This
function
includes
several
parameters:
tradeID,
tp, sl,
holdtime
and
orderRef.
Here are
the
descriptions

The
function
'self.evt.update_order(...)' returns
output
'res'. The
returned
output
'res' is
presented in a
JSON
object,
where
the value
can be
accessed
via a
dictionary
key
method.
'res'
includes
following
keys:

the
function
'self.evt.u
pdate_pe
nding_or
der(...)'
is used to
update
the
attached
paramete
rs of a
limit/stop
order
that is
pending
to fill.
The
function
'self.evt.u
pdate_pe
nding_or
der(...)'
has
following
paramete
rs:
tradeID

self.evt.update_pending_order(...)' returns output 'res'. The returned output 'res' is presented in a JSON object, where the value can be accessed via a dictionary key method. 'res' includes following keys: status and msg

The
function
'self.evt.u
pdate_po
rtfolio_sl(
...)' is
used to
set the
portfolio
level
stop loss
(in
percentag
e). When
the
account
NAV
drops for
such
percentag
e from its
previous
high
water
level, the
system
will auto
close all

Suppose
we want
to set a
global
stoploss
level for
the
whole
portfolio
to 10%,
and also
pause
the algo
submittin
g new
trades
for 1
week if
such stop
loss
event
happen.
We can
then
apply
this
setup at
initializati

ALGUGEN

E

provides

a utility

function

'self.evt.g

etHistoric

alBar(con

tract,

numOfBa

r,

interval,

timestam

p)' to

query

historical

market

data in

OHLC

bar. The

function

'self.evt.g

etHistoric

alBar()'

include

the

following

paramete

The
function
'self.evt.getHistoricalBar()'
returns
'res'. The
returned
output
'res' is
presented
in
JSON/
Python
dictionary
object.
The first
key will
be
timestamp
sorted
ascending
ly, and
'res[timestamp]'
includes
following
keys: t h

ALGUGEN

E

provides

a utility

function

'self.evt.g

etHistoric

alNews(la

ng,

count,

starttime,

endtime,

category,

source)'

to query

historical

news.

The

function

'self.evt.g

etHistoric

alNews()'

includes

following

paramete

rs: lang,

count,

starttime

self.evt.getHistoricalNews()
returns
output
'res'. The
returned
output
'res' is
presented
in a list
of news
objects in
JSON
format
with
ascending

timestamp.
p. 'res[i]'
includes
the
following
keys:
published
, source,
category,
title

ALGUGEN

E

provides
a utility
function
'self.evt.g
etHistoric
alEconsta
t(series_i
d,
starttime,

endtime)'
to query
historical
time
series of
Economic

Statistics.
The
function
'self.evt.g
etHistoric
alEconsta
t()'
includes
paramete

The
function
'self.evt.getHistoricalEconomicData()' returns
output
'res'. The
returned
output
'res' is
presented
in a list
of news
objects in
JSON
format
with
ascending

timestamp.
p. 'res[i]'
includes
the
following
keys:
date

The
function
'self.evt.getExchangeRate(cur1, cur2)'
is used to
get the
exchange
rate
between
currencies 'cur1'
and
'cur2'.

The
returned
result
represents the
number
of units in
'cur2' in
exchange
for 1 unit
of 'cur1'.

The
function

For live-
testing or
real-
trading,
the result
of
'self.evt.g
etExchan
geRate()'
will be
the latest
real-time
exchange
rate. For
backtesti
ng, the
result
will be
the
previous
dayend
(time
zone in
UTC+0)
exchange
rate of
the
query

Apart from the 'ab' object in 'on_marketdatafeed' and 'on_bulkdatafeed', the function 'self.event.getAccountBalance()' can also be used to query the current account balance. The result of 'self.event.getAccountBalance()' can be accessed

The
function
'self.evt.getContractSpecification(instrument)' is used to get the product feature or contract specification of a financial instrument. It is particularly useful if we want to get the current tradable Options/Futures.
The
function

The
function
'self.evt.g
etContrac
tSpec()'
outputs
'res'. 'res'
is
presente
d in a
JSON
object,
where
the value
can be
accessed
via a
dictionary
key
method.
'res'
includes
following
keys:
contractSi
ze,
descriptio
n

The
function
'self.evt.getSystem
Orders()' is used to
get
current
positions
(pos),
outstanding
opened
orders
(osOrder)
,
limit/stop
orders
that are
pending
to fill
(pendingOrder). The
function
outputs 3
output:
'pos',
'osOrder'

For the first output 'pos' of the function 'self.evt.getSystemOrders()', the first key will be the financial symbol, and 'pos[symbol]' has the following keys: LastTradeTime, netVolume. Here are the descriptions of the keys.

For the
second
output
'osOrder'
of the
function
'self.evt.g
etSystem
Orders()'
, the first
key will
be a
system
generate
d
'tradeID'
, and
'osOrder[
tradeID]'
has the
following
keys:
LastTrade
Time,
netVolum
e. Here
are the
descriptio

For the last output 'pendOrder' of the function 'self.eventGenerator.generateTradeID', and 'pendOrder[tradeID]' has the following keys: orderRef, market, broker, producttype

For stock market, a corporate action is an event carried out by a company that materially impacts its stakeholders.

These actions include the payment of dividends, stock split, mergers and acquisitions. The function

line
output
'res' of
the
function
'self.event.g
etHistoric
alCorpAct
ion(...)'
will be a
list of
JSON
object,
where
the value
can be
accessed
via a
dictionary
key
method.
'res'
includes
keys:
event,
announce
_date,
ex_date

The
function
'self.evt.getGoogle
TopSearch
h(...)' is
used to
query
Google's
daily top
20 search
results in
country/city level.
'self.evt.getGoogle
TopSearch
h(...)'
includes
parameters:
geo
and date.
Here are
the
descriptions
of the
parameters.

The
output
'res' of
the
function
'self.evt.g
etGoogle
TopSearch
h(...)' will
be a list
of JSON
object
with
search
interests
sorted in
an
descendi
ng order,
where
the value
can be
accessed
via a
dictionary
key
method.
'res'

The
function
'self.evt.getCCASSH
istory(...)'
is used to
query
HKEx
CCASS's
sharehold
ing
informati
on. The
function
'self.evt.getCCASSH
istory(...)'
includes
following
paramete
rs:
symbol,
participan
t_id,
starttime,
endtime.
Here are
the

the
output
'res' of
the
function
'self.evt.g
etCCASSH
istory(...)'
is
presente
d in a
JSON
object,
where
the value
can be
accessed
via a
dictionary
key
method.
'res'
includes
following
keys: t,
symbol,
id, type,
chara

The
function
'self.evt.getChainHistory(...)'
can be
used to
query
the
number
of unique
addresses
that
appeared
for the
first time
in a
transaction
of the
native
coin in
the
network,
query
the
number
of unique
addresses

The
function
'self.evt.g
etChainHi
story(...)'
can be
used to
query
the
number
of unique
addresses
that
appeared
for the
first time
in a
transactio
n of the
native
coin in
the
network.
The
function
'self.evt.g
etChainHi
story(...)'

the
output
'res' of
the
function
'self.evt.g
etChainHi
story(...)'
will be a
list of
JSON
object
sorted in
ascending
time,
where its
value can
be
accessed
via a
dictionary
key
method.
'res'
includes
following
keys: t, v.
Here are

SOPR
stands
for Spent
Output
Profit
Ratio.

SOPR is
compute
d by
dividing
the
realized
value (in
USD)
divided
by the
value at
creation
(USD) of
a spent
output.

In other
word,
$$\text{SOPR} = \frac{\text{price sold}}{\text{price paid}}$$

ALGOGENE. It covers many of the basics of ALGOGENE's API, and is designed for those who are new to the platform. It highlights the benefits over human traders in the timeliness and accuracy of execution, the computational speed for big calculations, and the flexibility of the platform for strategy development. It covers packages for mathematical optimization, statistical modelling, time series analysis, and simulation. It also covers data for major US Index, FX pairs, and Interest Rate Future CFDs, from 2000 up to the most recently completed trading month. All data is provided in a format that is easy to use and understand.

It defines a simple but powerful object-oriented model to process data and to simulate historical environment. It defines a simple but powerful object-oriented

Backtest environment. To do so, you should go to 'Backtest' -> 'script' section. There will be several components:

ts, it is recommended to set it to be the month in which all datasets of subscribed instruments are available.

tform. All you need to get started is to have some basic Python programming skills.

data analysis, and the elimination of emotional effects. Most trading algorithms make decisions based on mathematical analysis, machine learning, data mining, etc. For example, NumPy, SciPy, Pandas, Math, Matplotlib, SciPy. All the datasets are standardized into the same GMT+0 time zone.

defined API callbacks as building blocks. The building blocks include `init_MarketData`, `on_marketdatafeed`, `on_t`

s essential to be defined. That are Strategy Name, Backtest Start Month, Backtest End Month, Data Interval, In

ematical or statistical models that are derived from research conducted on historical data. To develop a succe
iKit-Learn, TA-Lib, etc.To use these libraries, you can simply add an 'import' statement at the beginning of your

bulkdatafeed, on_newsdatafeed, on_econsdatafeed, on_weatherdatafeed, on_orderfeed, on_openPositionfee

itial Capital, Leverage, Allow Short Sell, Position Netting, Enable News Stream, Enable Economics Stream, Enak

Successful algorithm for live trade, it is crucial to find an economic or statistical relationship on which we can base our script.

ed and on_dailyPLfeed.

le Weather Stream, Transaction Cost, Risk free rate, Instruments and Benchmark.

our strategy. Then, we will define our trading strategy and test whether it can effectively generate returns base

ed on historical data.