GPU Accelerated Backtesting and ML for Quant Trading Strategies

GTC 2015 | San José | California

Dr. Daniel Egloff
daniel.egloff@quantalea.net
March 18, 2015

Motivation

QuantAlea

- Goals
  - Execute automated algorithmic trading strategies
  - Optimize risk return
- Procedure
  - Extract signals and build price forecasting indicators from market data
  - Transform indicators into buy / sell decisions
  - Apply portfolio risk management
- Challenges
  - Find relevant signals and indicators
  - Engineer and parameterize trading decision
  - Find optimal parameters
- Approach
  - Exploit parallelism in the computations
  - Accelerate calculations by using a GPU cluster

Algo Trading Strategies

QuantAlea

Market data

Rules

Trading decision

Symbol	Pct	Last	Change	Prev	Bid	Ask	High	Low	EVol	Volume
ES #F	+0.11%	136400	+150	136250	136400	136425	136625	134250	1084656	2076198
SP #F	+0.13%	136430	+180	136250			136620	134210	2111	10585
$SPX	-0.05%	1368.39	-0.71	1369.10			1370.08	1363.94		
NQ #F	+0.25%	263200	+650	262550	263175	263200	264150	258650	160303	320105
$NDX D	+0.01%	2638.09	+ 0.17	2637.92			2644.37	2620.96		
$COMP D	-0.04%	2955.09	-1.25	2956.34			2960.67	2939.21		
YM #F	-0.08%	1294600	-1100	1295700	1294500	294600	1296200	1277900	78269	139189
DJ M2	-0.02%	1295500	-200	1295700	1295000		1296000	1291000	0	131
DJ #F	-0.02%	1295500	-200	1295700	1295000		1296000	1291000	0	131
ES U2	+0.17%	135825	+225	135600	135750	135800	135975	133750	908	1679
$INDU D	-0.27%	13003.04	-35.23	13038.27			13036.15	12970.00		43960296
TF #F	+0.23%	78840	+180	78660	78830	78840	79140	77280	66434	153710
$RUT D	-0.06%	791.34	-0.50	791.84			792.67	788.07		
MC #F	+0.29%	96330	+280	96050	96320	96340	96630	94660	15015	26228
NK #F	-1.13%	9155	-105	9260			9155	9155	0	3315
$TRAN D	+0.57%	5257.48	+29.84	5227.64			5257.62	5198.00		4747289
ZB #F	+0.04%	14324	+2	14322	14324	14325	14419	14321	142139	323844
US M2	+0.07%	14325	+3	14322			14325	14325	0	323844
$TYX	-0.20%	3.065	-0.006	3.071			3.070	3.052		
ZN M2	+0.01%	132210	+5	132205	132210	132215	133020	132195	484043	1299349
TY M2	+0.34%	132205	+145	132060			132210	132210	0	1299349
$TNX	-0.37%	1.873	-0.007	1.880			1.880	1.859		
ZF M2	+0.02%	123307	+7	123300	123307	123310	124032	123297	170942	405572
ZT M2	-0.01%	110087	-2	110090	110087	110090	110092	110082	59470	122419
6E M2	-0.28%	13052	-37	13089	13051	13052	13062	12957	190719	237586
6S M2	-0.30%	10868	-33	10901	10867	10868	10877	10778	35792	39567
6B M2	+0.26%	16183	+42	16141	16182	16184	16193	16110	64892	99435
6J M2	+0.01%	12526	+1	12525	12525	12526	12564	12504	48836	78990
6A M2	+0.20%	10160	+20	10140	10159	10160	10165	10065	83072	160266
6C M2	+0.10%	10048	+10	10038	10047	10048	10060	10002	60345	103796
6N M2	+0.05%	7934	+4	7930	7934	7936	7945	7886	11427	18113
DX M2	+0.15%	79705	+118	79587	79705	79710	80135	79660	17688	24757

Mathematical operations

Input

Output

Configurations

Example

QuantAlea

Buy signal

Sell signal

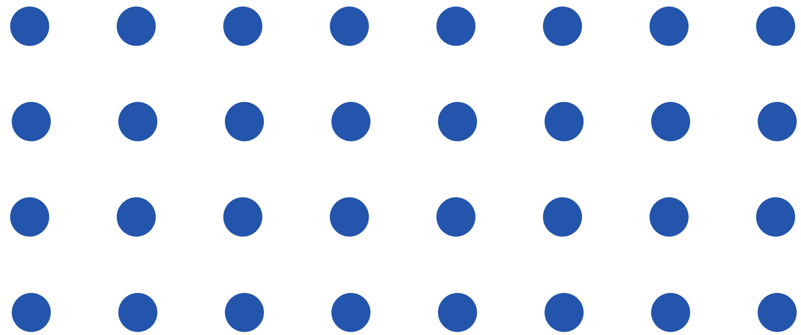
Markets

- Futures market
  - CME 50 liquid futures
  - Other exchanges
- Equity markets
  - World stock indices
- FX markets
  - 10 major currency pairs
  - 30 alternative currency pairs
- Options markets
  - Options on futures
  - Options on indices

Strategy Universe

QuantAlea

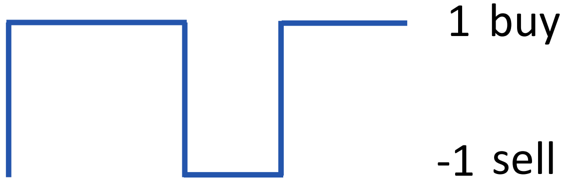
Strategy configurations



Configuration c

Trading decision s(c)

Utility U(s(c))



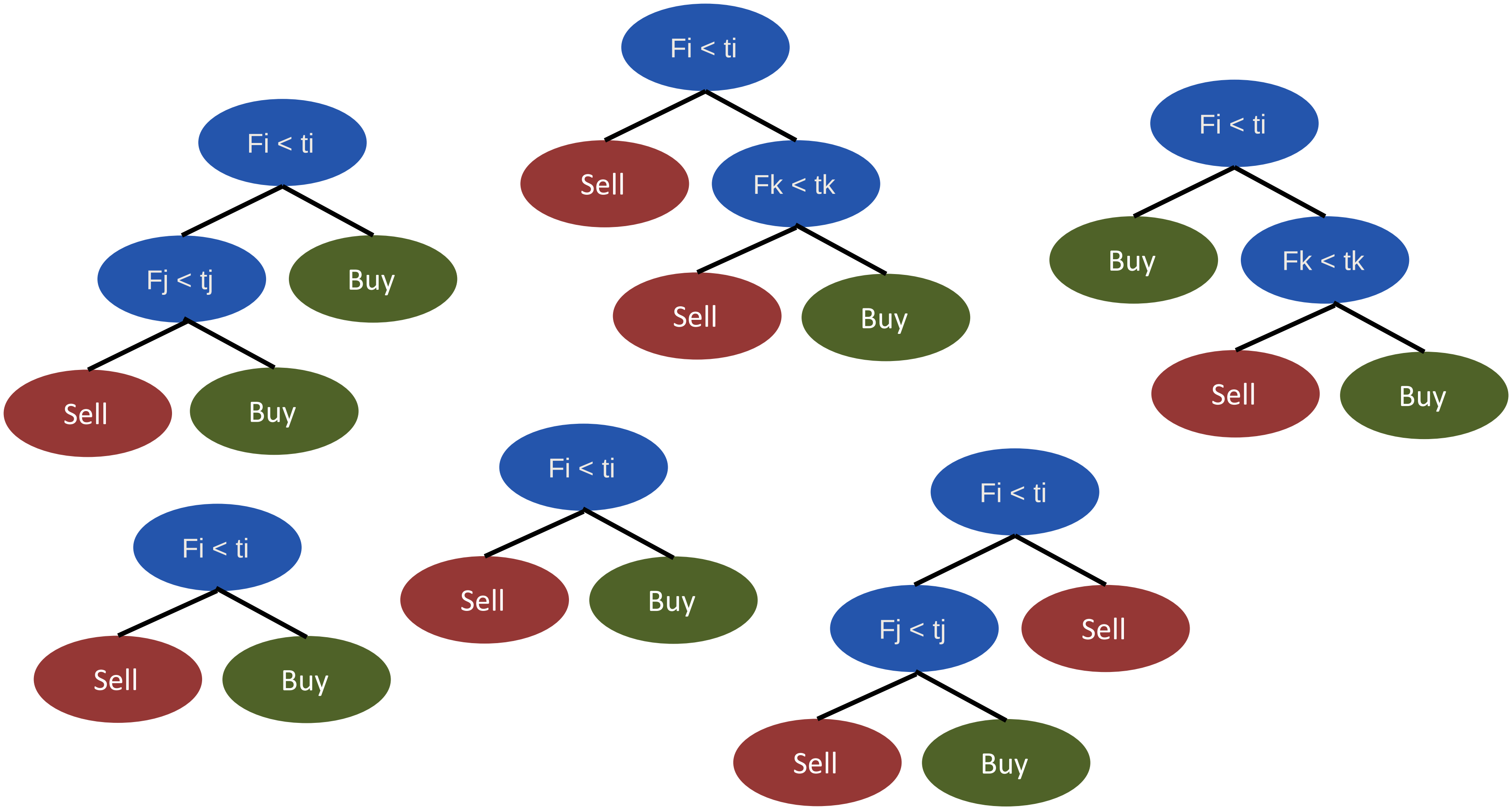
P&L / drawdown

Strategy Engineering

QuantAlea

Challenge 1: How can we engineer a strategy producing buy / sell decisions ?

Random Forests



Random Forests

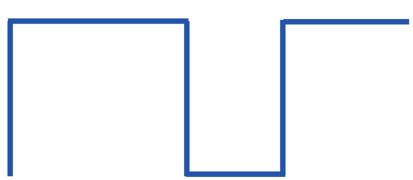
QuantAlea

Strategy configuration c

Features

Random forest

Trading decision s(c)



1 buy

-1 sell

Training Random Forests

Bootstrapping to create training sets

C 4.5 algorithm for individual tree construction

- Selecting subset of features for tree construction
- Each node is associated with a subset of training samples
- Recursive, starting at the root node
- At each node execute divide and conquer algorithm to find locally optimal choice
  - If samples are in same class (or few class) node is a leaf associated with that class
  - If samples are in two or more classes
    - Calculate information gain for each feature
    - Select feature if largest information gain for splitting

Entropy

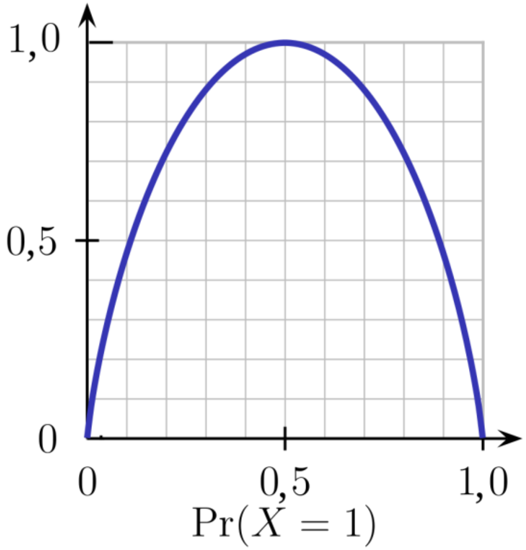
T = set of samples associated with node

$C_1, \ldots, C_n$= classes of samples

Entropy

$$Ent(T) = -\sum_{i=1}^{n} \frac{freq(C_i,T)}{|T|} \; log_2\left(\frac{freq(C_i,T)}{|T|}\right)$$

- Characterizes impurity of samples
- Measure of uncertainty
- Additive: impurity of several subsets is sum of impurities



Information Gain

QuantAlea

$T_1, \ldots, T_s$ = subsets of T generated by splitting on selected attribute

Information gain discrete feature

$$gain(T_1, \ldots, T_s) = Ent(T) - \sum_{i=1}^{s} \frac{|T_i|}{|T|} Ent(T_i)$$

Information gain continuous feature with optimal splitting threshold

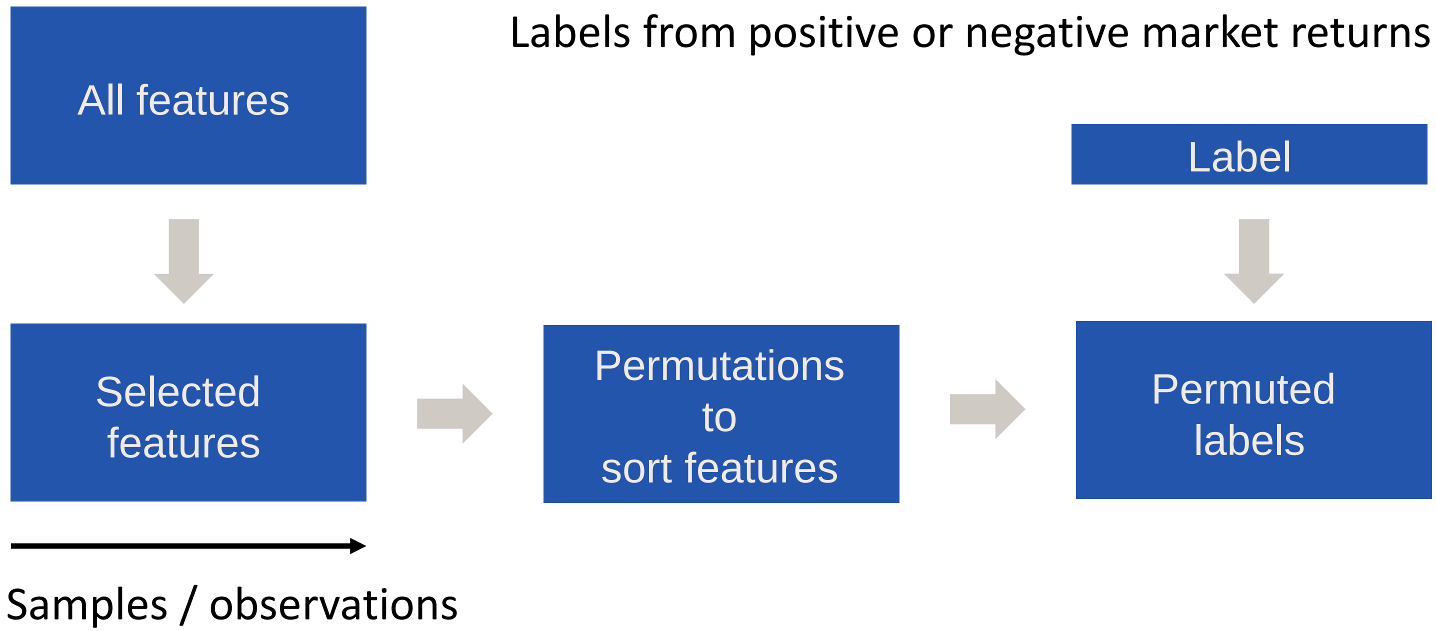
$$gain(t) = gain(T_{\leq t}, T_{>t})$$

$$t_* = argmax\ gain(t)$$

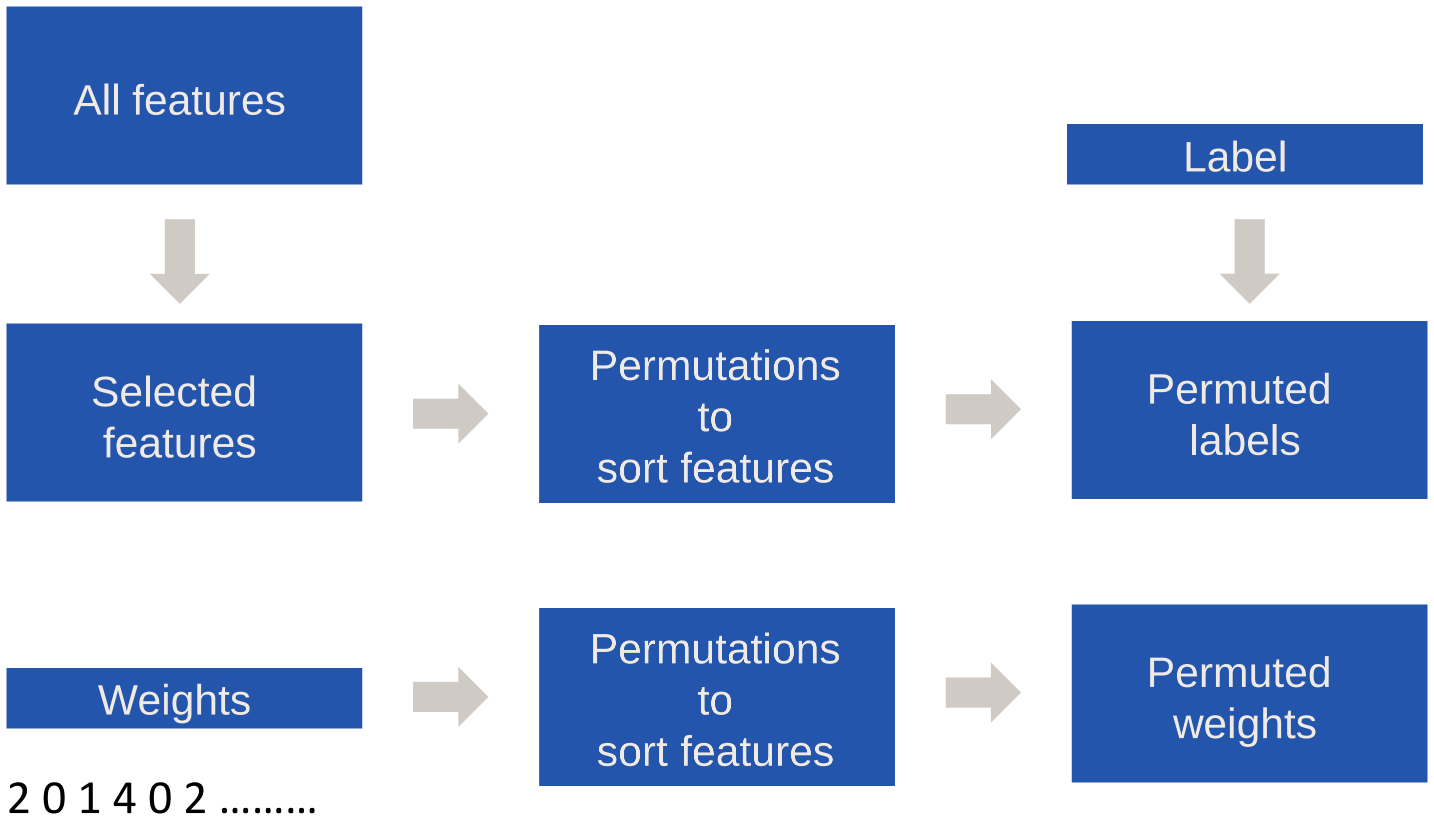
Actual implementation uses ratio information gain over split ratio

Training Individual Trees

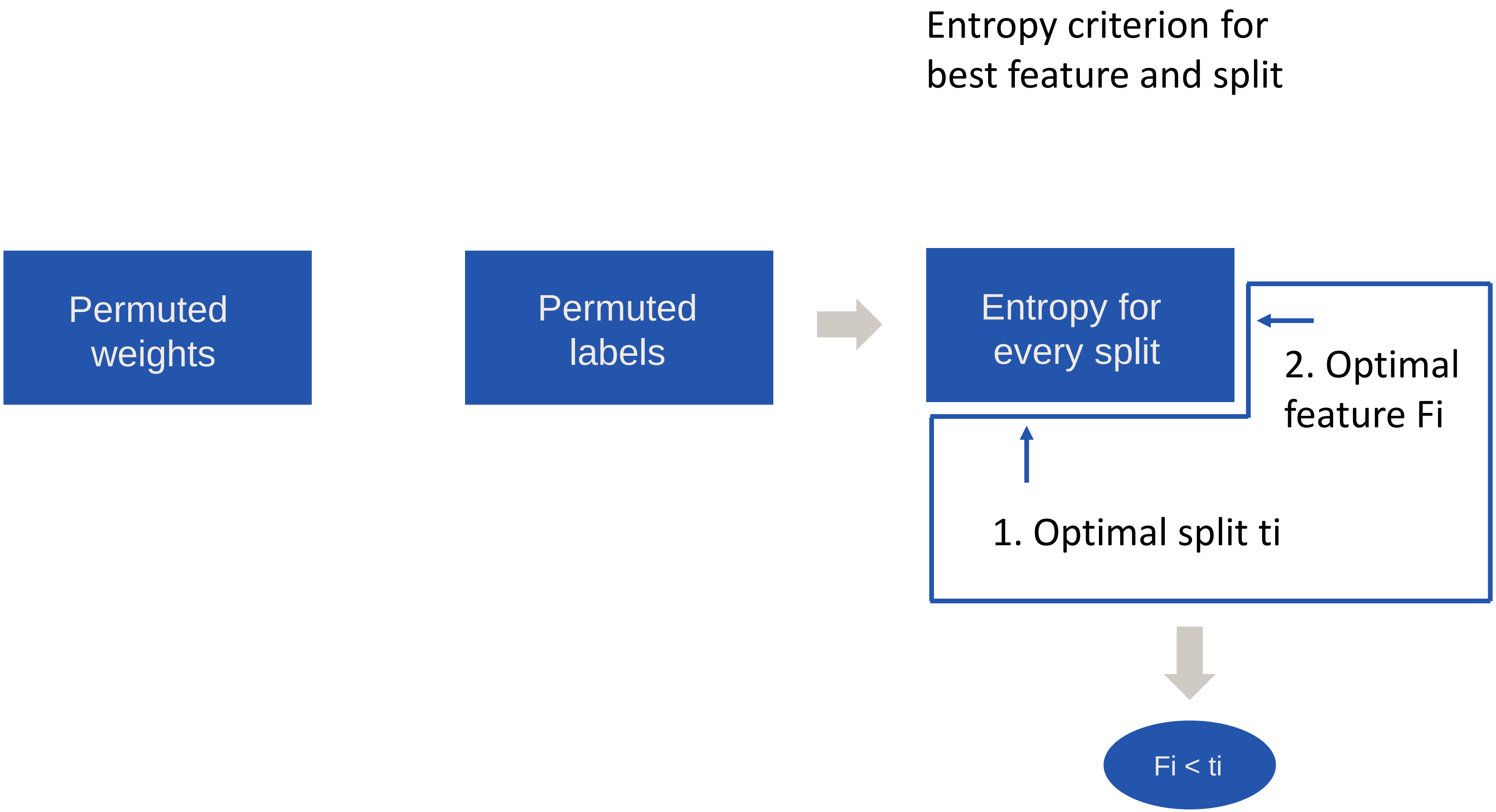
QuantAlea



Training Individual Trees

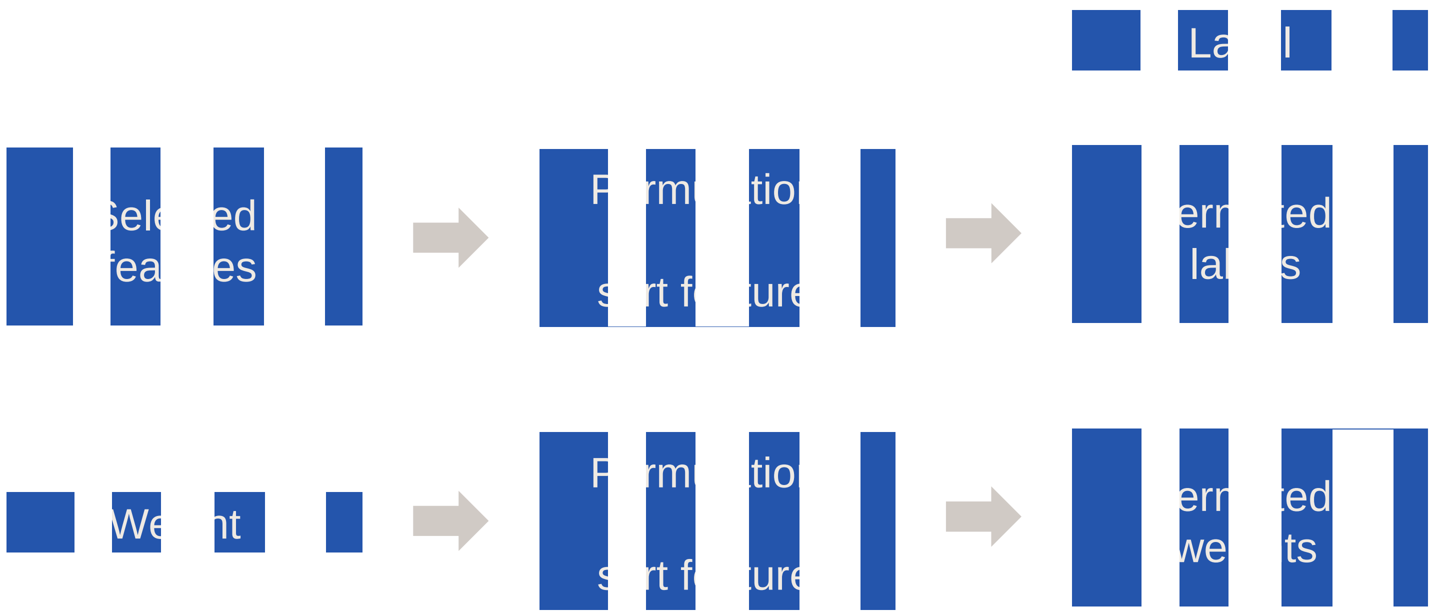


Training Individual Trees



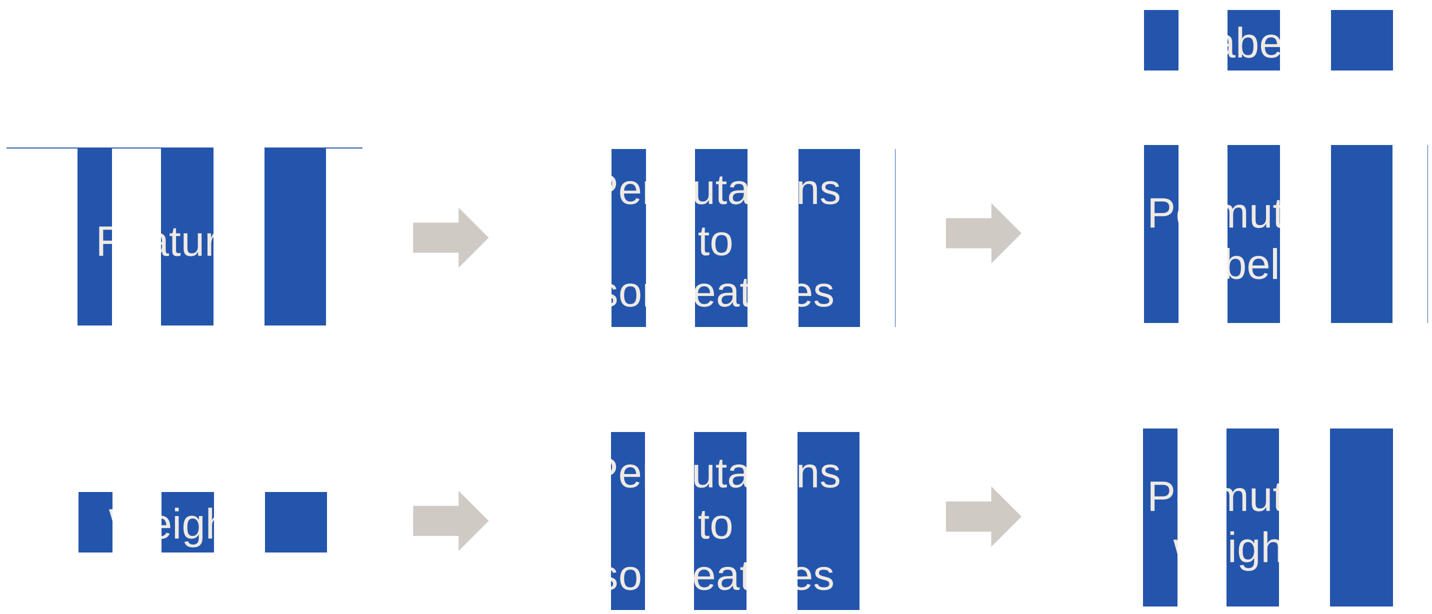
Training Individual Trees

Recursively refine classification: mask data according to classification



Training Individual Trees

Recursively refine classification: mask data according to classification

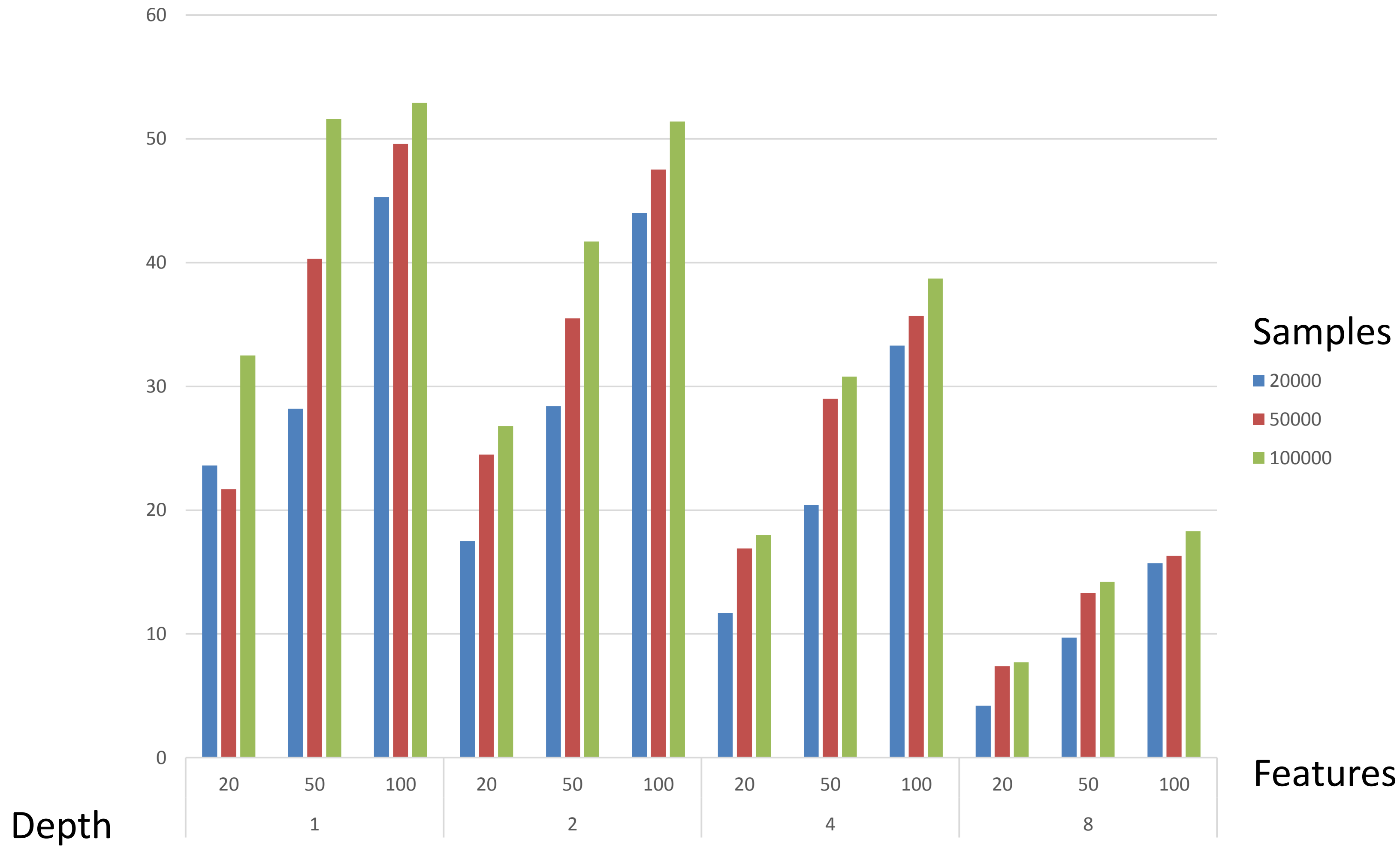


GPU Implementation

QuantAlea

- Parallelism at multiple levels
  - Multiple trees, one for each set of weights
  - Independent features
  - Independent split points
  - Multiple nodes further down the tree
- GPU kernels can be implemented with standard primitives
  - Random number generation for weights
  - Parallel scan (cumulative sum)
  - Parallel map
  - Parallel reduction to find optimal feature and split

Speedup

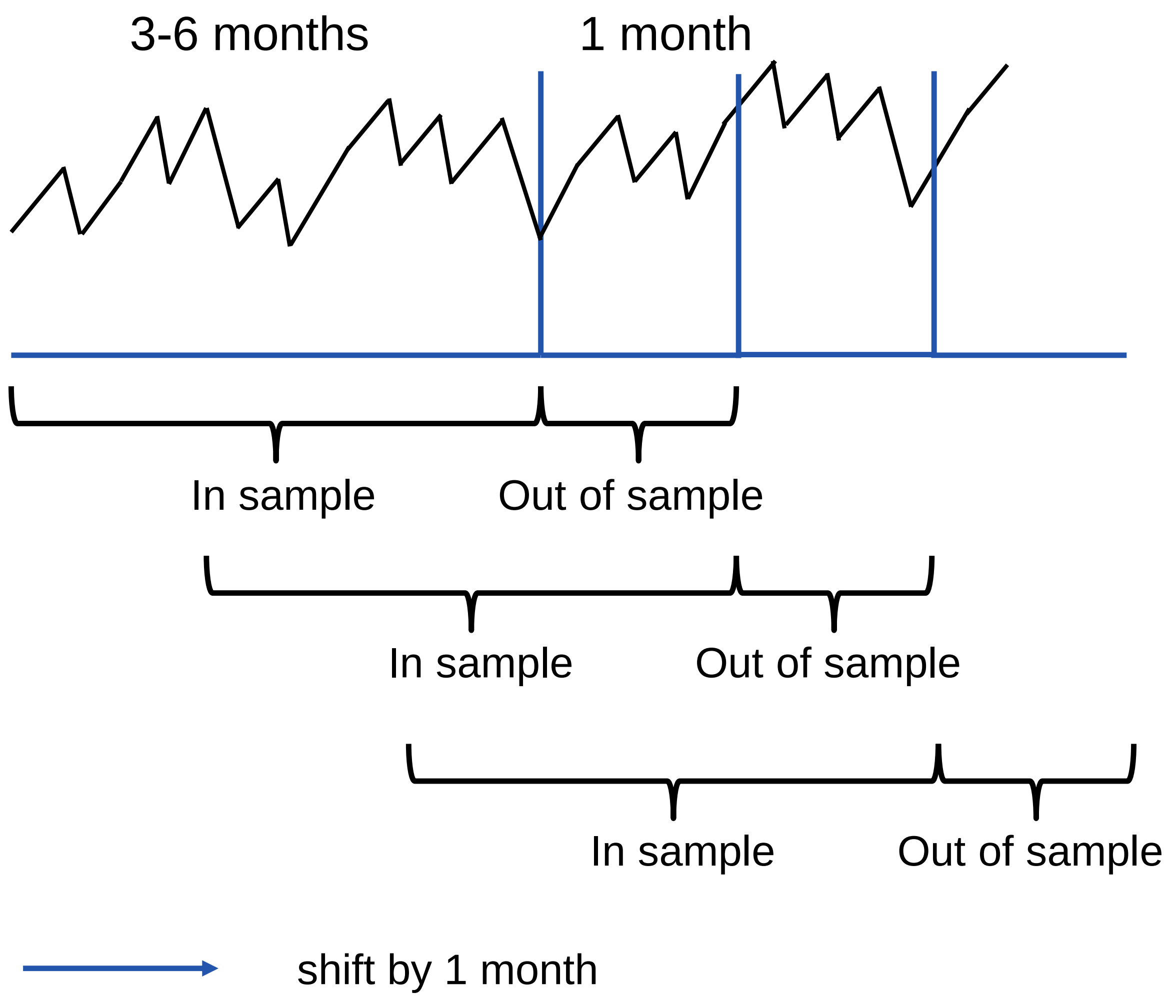


60
50
40
30
20
10
0
20
50
100
20
50
100
20
50
100
20
50
100
1
2
4
8
Depth
Samples
20000
50000
100000
Features

Strategy Backtesting

Challenge2: How to choose best trading strategy ?

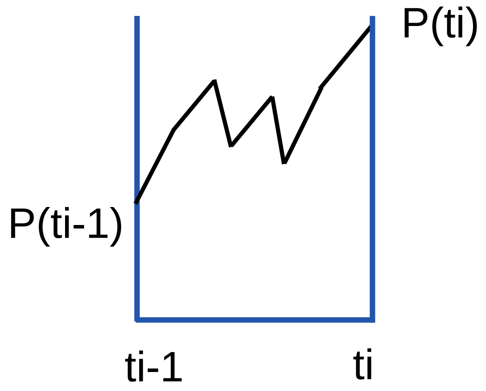
Walk Forward Optimization



Trading P&L

Market prices

Market returns



r(ti) = log(P(ti) / P(ti-1))

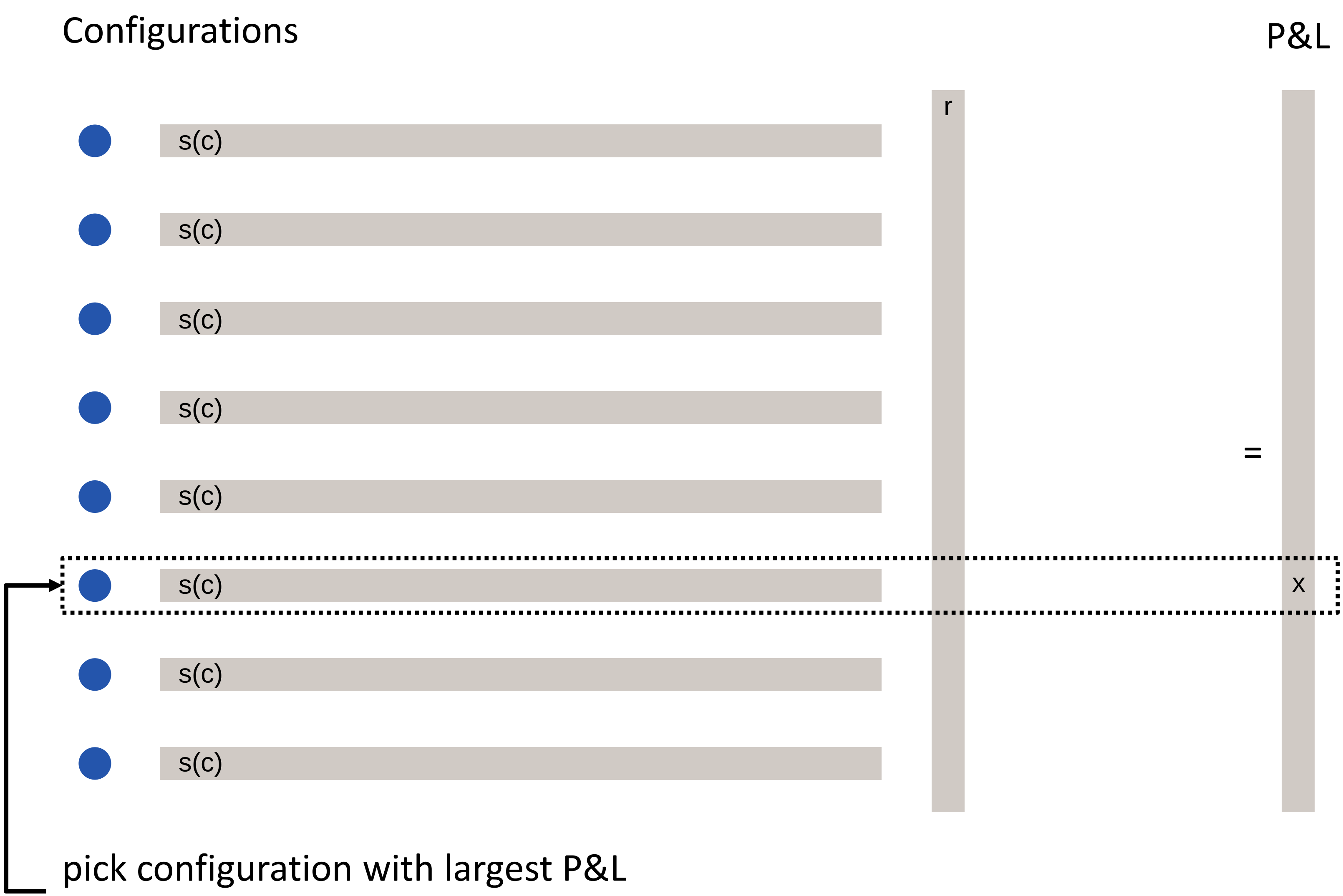
Market returns r(ti) ..... | r | r | r | r | r | .....

Trading decision s(c) ..... | 1 | -1 | 1 | 1 | -1 | .....

time

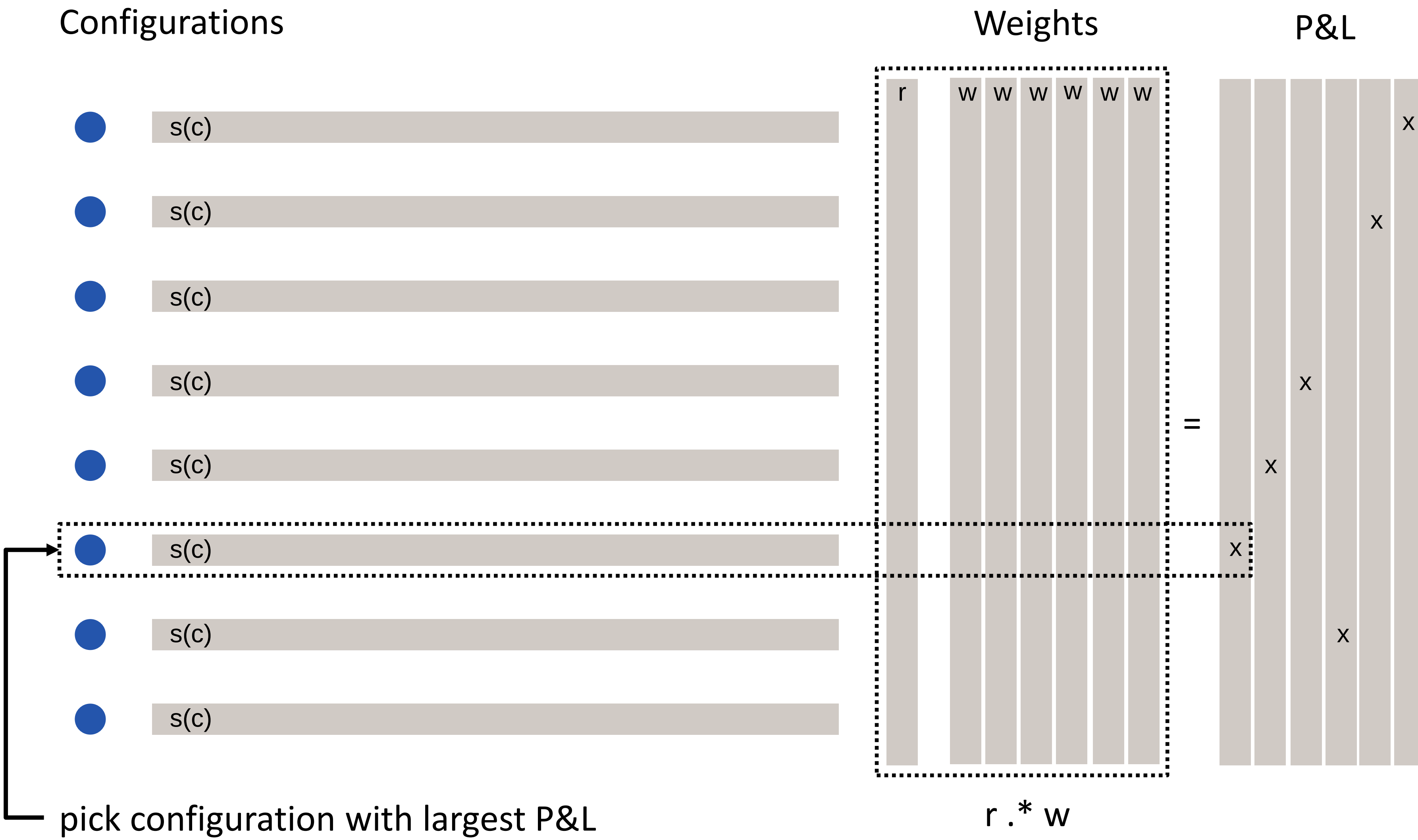
P&L(c) = <s(c), r>

Optimal Configuration



pick configuration with largest P&L

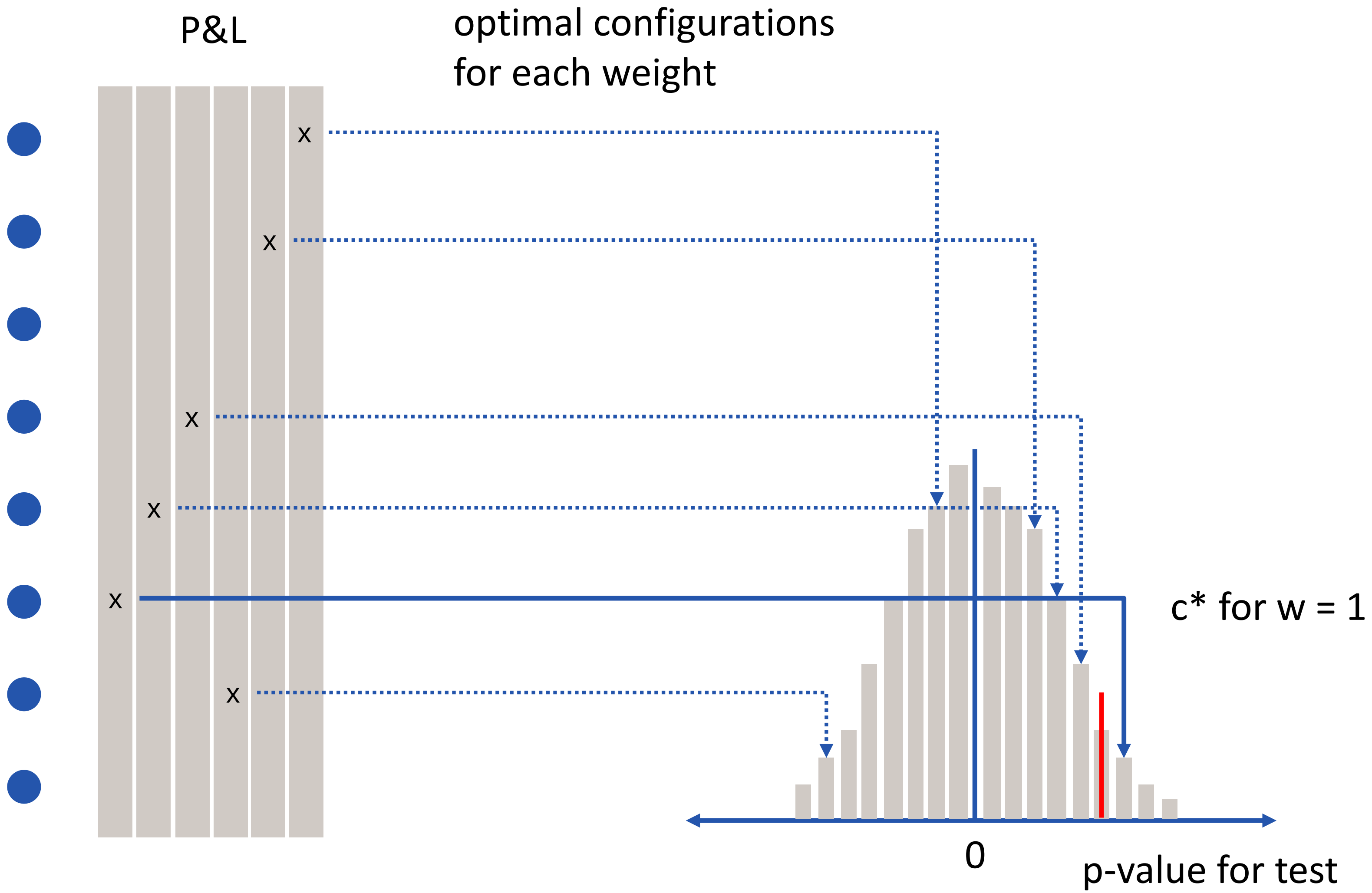
Bootstrapping Trading P&L



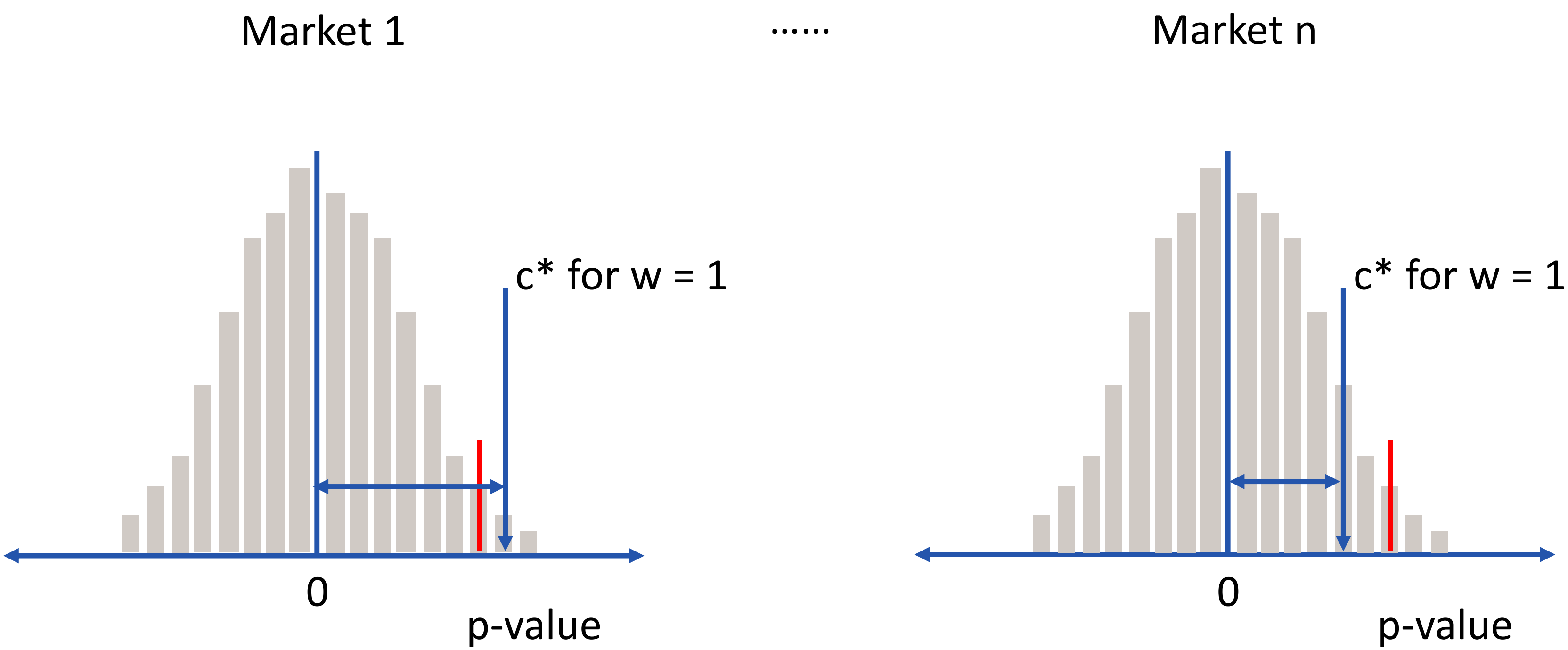
Hypothesis Tests

Null Hypothesis	Alternative Hypothesis
Trading P&L<= 0	Trading P&L > 0

Trading P&L Distribution



White Reality Check



- Scale exposure according to distance from 0
- Do not trade if negative returns

GPU Implementation

- Parallelism at multiple levels
  - Multiple markets
  - Independent in-sample / out-of-sample windows
  - Independent strategy configurations
  - Independent time steps for utility functions such as mean return
- GPU kernels can be implemented with standard primitives
  - Random number generation
  - Matrix multiplication (almost, up to return vector scaling the weights)
  - Parallel reduction

GPU Implementation

QuantAlea

- GPU grid
  - Multiple markets
  - Independent in-sample / out-of-sample windows
- Per GPU
  - Independent strategy configurations
  - Independent time steps for utility functions such as mean return

Questions ?

Dr. Daniel Egloff
daniel.egloff@quantalea.net
March 18, 2015