Machine Learning For Trading: From Automated Decisions to Decision Support

Charles-Albert Lehalle,

Senior Research Advisor (Capital Fund Management, Paris)
Visiting Researcher (Imperial College, London)

Applied Machine Learning Days – Lausanne, Jan 2019

Outline

Outline

What Is It About?

Let's Try To Identify a Clear Set of Questions

There is a lot of buzz around:

- Machine Learning
- Artificial Intelligence
- TensorFlow and Keras...

What Is It About?

Let's Try To Identify a Clear Set of Questions

There is a lot of buzz around:

- Machine Learning
- Artificial Intelligence
- TensorFlow and Keras...

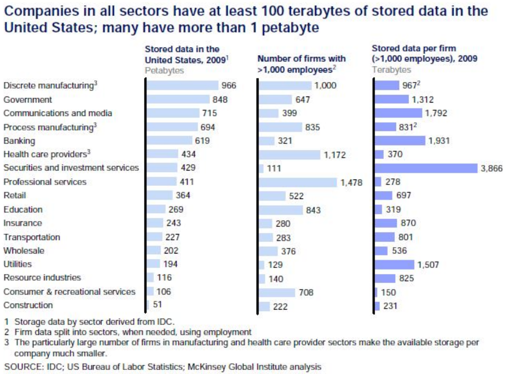
Are they related? Is it really important for us to understand them?

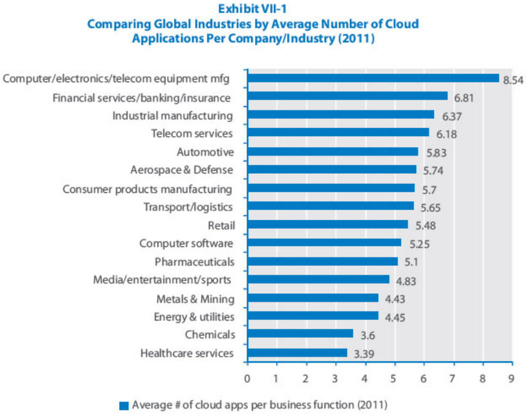
- The Louis Bachelier Institute just launched a transverse research program to work on the impact of these new technologies on the financial and insurance sectors (bit.ly/ILBFaIR: Finance and Insurance Reloaded),
- Today my talk will be on a small subset of this: **two applications of ML techniques to optimal trading**.

Is The Financial Sectors Ready?

cf: Les données au service de la mesure des risques économiques et financiers

The Financial Sector is One of The Largest User of CPU and Cloud





source: blogs.saphana.com

source: tcs.com

Focus on Trading

Two Examples of Applications to Trading

Trading evolved these last 20 years to be mostly electronic of a lot of instruments. It means that large investors are using trading algorithms to buy or sell a large amount of shares or contracts, and that in Europe, event retail investors use the “Smart Order Routers” of their brokers [Lehalle et al., 2018].

- **Fragmentation** is at the root of disintermediation (think about `youtube.com` or `uber.com`).
  There is no more one financial market place but a collection of electronic trading venues (compared to OTC trades). I will show **how to optimally fragment and route an order** to obtain the needed liquidity at the best price? [Laruelle et al., 2011]

- **Human-Machine Interface** will be more needed than ever.
  Humans will need decision support tools in an automated environment, and machines will have to take profit of humans’ understanding of the context. I will present **a decision-support system** to allow few traders to monitor thousands of trading algorithms. [Azencott et al., 2014]

Outline

Fragmentation is at The Root of Disintermediation

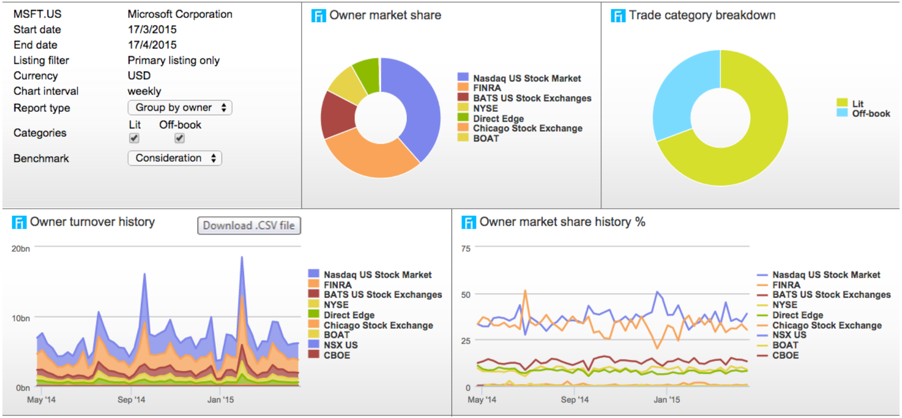
Have you ever seen a mailbox in France?

Fragmentation is at The Root of Disintermediation

Have you ever seen a mailbox in France?

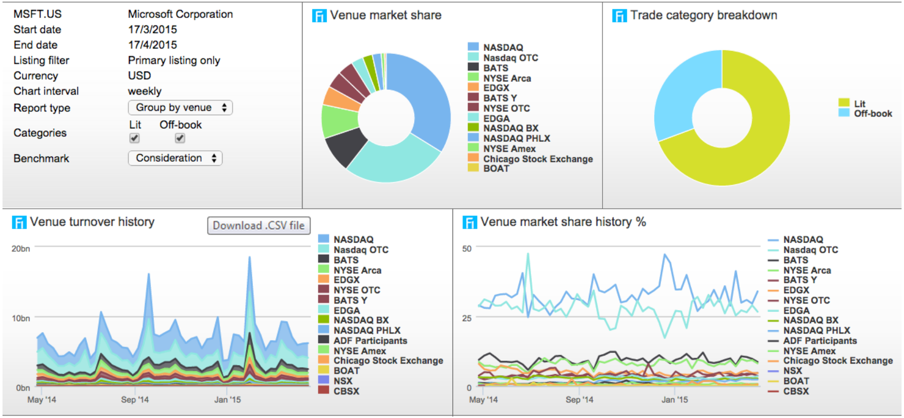
On Berkeley's Campus: This is competition (and fragmentation)

A typical fragmented stock



The Fragmentation of Microsoft the last 20 days (Source: Fidessa's fragulator)

A typical fragmented stock



The Fragmentation of Microsoft the last 20 days (Source: Fidessa's fragulator) more...

The Stakes of Optimal Routing

When a human or robot trader wants to buy or sell few shares, he has to split his order and send it to available venues in the hope to obtain the desired size.

- one the one hand you have to split according to information you have
- be sure to be kept updated when information changes...

It is a typical **Exploration-Exploitation problem** [Lamberton and Pagès, 2008], especially in Dark Pools. We (joint work with S. Laruelle and G. Pagès [Laruelle et al., 2011]) solved it using a **stochastic algorithm**.

Documented approaches:

- [Ganchev et al., 2010] estimates the liquidity in each pool and implements a deterministic optimization;
- [Agarwal et al., 2010] uses a minimum regret approach;
- We implement the stochastic version of an optimal trading scheme.

The first approach is goog for on opportunistic trading (hedge fund), the second for a rare and not really flexible flow (investor), the last one is good for very large systematic flow (broker).

Expected Execution Cost

Minimizing the expected execution cost, *given the price S*, amounts to:

Maximization problem to solve

$$\max \left\{ \sum_{i=1}^{N} \rho_i \mathbb{E}\left(S \left(r_i V \wedge D_i \right) \right), r \in \mathcal{P}_N \right\}$$

where $\mathcal{P}_N := \left\{ r = (r_i)_{1 \leq i \leq N} \in \mathbb{R}_+^N \mid \sum_{i=1}^N r_i = 1 \right\}$.

It is then convenient to **include the price S into both random variables V and D_i** by considering $\widetilde{V} := V\,S$ and $\widetilde{D}_i := D_i S$ instead of V and D_i. Assume that the distribution function of D/V is continuous on $\mathbb{R}_+$. Let $\varphi(r) = \rho\,\mathbb{E}\left(\min\left(rV, D\right)\right)$ be the mean execution function of a single dark pool ($\Phi = \sum_i \varphi_i(r_i)$), and assume that $V > 0\ \mathbb{P} - a.s.$ and $\mathbb{P}(D > 0) > 0$

A Generic Method to Perform Online Optimization

Let's take 2 slides to understand a generic method to be applied to **any online trading algorithm**.

Few preliminary remarks:

- The stationary solutions of the ODE: $\dot{x} = h(x)$ contains the extremal values of $F(x) = \int_0^x h(x)\,dx$
- A discretized version of the ODE is (γ is a step):

$$x_{n+1} = x_n + \gamma_{n+1}\, h(x_n) \tag{1}$$

- A stochastic version of this being (ξ_n are i.i.d. realizations of a random variable, $h(X) = \mathbb{E}(H(X, \xi_1))$, $F(x) = \int_0^x \mathbb{E} H(X, \xi_1)\,dx$):

$$X_{n+1} = X_n + \gamma_{n+1}\, H(X_n, \xi_{n+1}) \tag{2}$$

- the stochastic algorithms theory is a set of results describing the relationship between these 3 formula and the nature of γ, H, h and ξ [Hirsch and Smith, 2005], [Kushner and Yin, 2003], [Doukhan, 1994]

The Theory of Stochastic Algorithms

Now we can do the reverse. This theory can be used when you only have a sequential access to a functional you need to minimize. It is clearly the case in trading.

- To minimize a criteria $\mathbb{E}(F(X, \xi_1))$ of a state variable X
- if it is possible to compute:

$$H(X_n, \xi_{n+1}) := \frac{\partial F}{\partial X}(X_n, \xi_{n+1})$$

- now you can implement the following sequencial algorithm (no more expectation in it):

$$X_{n+1} = X_n + \gamma_{n+1}\, H(X_n, \xi_{n+1})$$

- the results of the stochastic algorithms theory (like the Robbins-Monro theorem [Robbins and Monro, 1951], [Pagès et al., 1990]) gives conditions under which it converges (especially on the time steps γ).
- Moreover they give you central limit theorem like results, i.e. **you can control the variance** (i.e. the speed at which it converges).

Back to Dark Pool splitting

We aim at solving the following maximization problem:

$$\max_{r \in \mathcal{P}_N} \Phi(r), \;\; \Phi(r) := \sum_{i=1}^{N} \rho_i \mathbb{E}\left(S\left(r_i V \wedge D_i\right)\right).$$

The Lagrangian associated to the sole affine constraint is

$$L(r, \lambda) = \Phi(r) - \lambda \left(\sum_{i=1}^{N} r_i - 1 \right)$$

So,

$$\forall i \in \mathcal{I}_N, \quad \frac{\partial L}{\partial r_i} = \varphi_i'(r_i) - \lambda.$$

This suggests that any $r^* \in \arg\max_{\mathcal{P}_N} \Phi$ iff $\varphi_i'(r_i^*)$ is constant when i runs over $\mathcal{I}_N$ or equivalently if

$$\forall i \in \mathcal{I}_N, \quad \varphi_i'(r_i^*) = \frac{1}{N} \sum_{j=1}^{N} \varphi_j'(r_j^*).$$

Existence of maximum

To ensure that the candidate provided by the Lagragian approach is the true one, we need an additional assumption on φ to take into account the behaviour of Φ on the boundary of $\partial\mathcal{P}_N$.

Proposition 1

Assume that (V, D_i) satisfies upper assumptions for every $i \in \mathcal{I}_N$. Assume that the functions φ_i satisfy the following assumption

(3) $$(\mathcal{C}) \equiv \min_{i\in\mathcal{I}_N} \varphi_i'(0) > \max_{i\in\mathcal{I}_N} \varphi_i'\left(\frac{1}{N-1}\right).$$

Then $\arg\max_{\mathcal{H}_N} \Phi = \arg\max_{\mathcal{P}_N} \Phi \subset int(\mathcal{P}_N)$ where

$$\arg\max_{\mathcal{P}_N} \Phi = \left\{r \in \mathcal{P}_N \mid \varphi_i'(r_i) = \varphi_1'(r_1), i = 1, \ldots, N\right\}.$$

Design of the stochastic algorithm

Characterization of the solution

$$r^* \in \arg\max_{\mathcal{P}_N} \Phi \Leftrightarrow \forall i \in \{1,\dots,N\}\,,\ \mathbb{E}\left(V\left(\rho_i\delta_{\{r_i^* V < D_i\}} - \frac{1}{N}\sum_{j=1}^{N}\rho_j\delta_{\{r_j^* V < D_j\}}\right)\right) = 0.$$

Consequently, this leads to the following recursive zero search procedure

$$r_i^{n+1} = r_i^n + \gamma_{n+1} H_i(r^n, Y^{n+1}),\ r^0 \in \mathcal{P}_N,\ i \in \mathcal{I}_N, \tag{4}$$

where for $i \in \mathcal{I}_N$, every $r \in \mathcal{P}_N$, every $V > 0$ and every $D_1,\dots,D_N \geq 0$,

$$H_i(r,Y) = V\left(\rho_i\delta_{\{r_i V < D_i\}} - \frac{1}{N}\sum_{j=1}^{N}\rho_j\delta_{\{r_j V < D_j\}}\right)$$

with $(Y^n)_{n\geq 1}$ a sequence of random vectors with non negative components such that, for every $n \geq 1$, $(V^n, D_i^n, i = 1,\dots,N) \stackrel{d}{=} (V, D_i, i = 1,\dots,N)$.

And More...

The underlying idea of the algorithmis to reward the dark pools which outperform the mean of the N dark pools by increasing the allocated volume sent at the next step (and conversely).

Theorem 1: Convergence

Assume that (V, D) satisfy upper assumptions, that $V \in L^2(\mathbb{P})$ and that Assumption $(\mathcal{C})$ holds. Let $\gamma := (\gamma_n)_{n\geq 1}$ be a step sequence satisfying the usual decreasing step assumption

$$\sum_{n\geq 1} \gamma_n = +\infty \quad \text{and} \quad \sum_{n\geq 1} \gamma_n^2 < +\infty.$$

Let $(V^n, D_1^n, \ldots, D_N^n)_{n\geq 1}$ be an i.d.d. sequence defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Then, there exists an $\mathrm{argmax}_{\mathcal{P}_N} \Phi$-valued random variable r^* such that

$$r^n \longrightarrow r^* \quad a.s.$$

Rate of convergence (Central Limit Theorem)

To establish a *CLT*, we need to ensure the existence of the Hessian of the objective function Φ. This needs further assumption on a couple (V, D) which is that its distribution function given $\{D > 0\}$ is absolutely continuous with a density f defined on $(0, +\infty)^2$. Furthermore , for every $v > 0$, $u \mapsto f(v, u)$ is cont. and pos. on $\mathbb{R}_+$, and $\forall\, \varepsilon \in (0, 1), \sup_{\varepsilon V \le u \le V/\varepsilon} f_D(V, u)V^2 \in L^1(\mathbb{P})$. The conditional distribution function of D given $\{D > 0\}$ and V is given by for $u \ge 0,\ v > 0$,

$$F_D(u \,|\, V = v, \delta_{\{D>0\}}) := \mathbb{P}(D \le u \,|\, V = v, \delta_{\{D>0\}}) = \int_0^u f(v, u')du'$$

Theorem 2: Central Limit Theorem

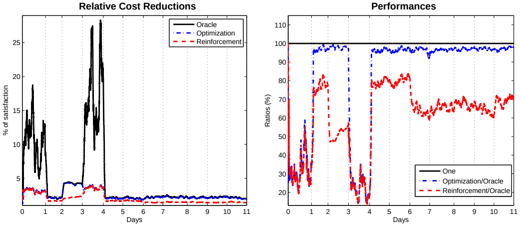
Assume that $\operatorname{argmax} \Phi = \{r^*\}$, $r^* \in \mathcal{P}_N$ so that $r^n \xrightarrow{n\to\infty} r^*$ $\mathbb{P}$-*a.s.* and that Assumption (**??**) holds for every (V, D_i), $i \in \mathcal{I}_N$ and $V \in L^{2+\delta}(\mathbb{P})$, $\delta > 0$. Set $\gamma_n = \frac{c}{n}$, $n \ge 1$ with $c > 1/2\Re e(\lambda_{\min})$ where $\lambda_{\min}$ denotes the eigenvalue of $A^\infty := -Dh(r^*)_{|\mathbf{1}^\perp}$ with the lowest real part. Then $\sqrt{\gamma_n}^{-1}(r^n - r^*) \xrightarrow{\mathcal{L}} \mathcal{N}(0; \Sigma^\infty)$, where the asymptotic covariance matrix Σ^∞ is given by

$$\boxed{\Sigma^\infty = \int_0^\infty e^{u(A^\infty - \frac{Id}{2c})} C^\infty e^{u(A^\infty - \frac{Id}{2c})^t} du}$$

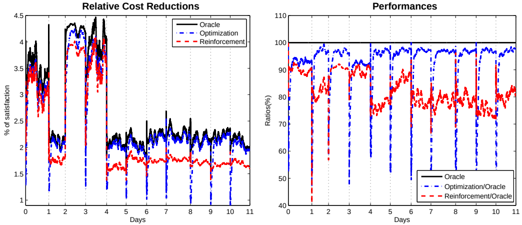
where $C^\infty = \mathbb{E}\left(H(r^*, V, D_1, \ldots, D_N)H(r^*, V, D_1, \ldots, D_N)^t\right)_{|\mathbf{1}^\perp}$ and $(A^\infty - \frac{Id}{2c})^t$ stands for the *transpose operator* of $A^\infty - \frac{Id}{2c} \in \mathcal{L}(\mathbf{1}^\perp)$.

In Practice

Some “backtests” using reconstructed data: we implemented an “Oracle” (it knows the future) and a simpler “reinforcement” policy (at left). We tested different market impact functions $\kappa = 0$ means no impact (at right).



Maintaining the learning during 10 day



Reinitialization each morning

Of course it is possible to do better, for instance by implementing a “smart reset”, etc.

Outline

Decision Support For Thousands of Trading Algorithms

Each trader monitors 150 to 700 trading algorithms. Algorithms react:

- to realtime feeds,
- estimates,
- market state.

Algo have “meta parameters” that can be tuned by traders.

In real-time, we (joint work with R. Azencott, A. Beri, Y. Gadhyan, N. Joseph and M. Rowley [Azencott et al., 2014]) will

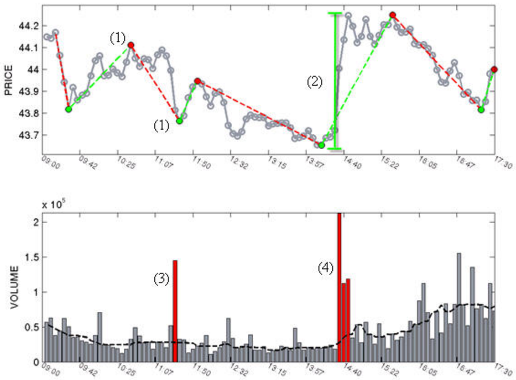
- attempt to predict on the fly the quality of trading of the thousands of algos using potential explanatory variables (i.e. “features”),
- that for we will need to extract features on price formation at high frequency,
- if a feature explains successfully bad performance, we will
  1. announce the feature is potentially at the root of the anomaly,
  2. group algorithms using this feature to help traders to take decisions.

How to monitor all this in real-time?

We would like to **automatically find common causes of bad performances**.

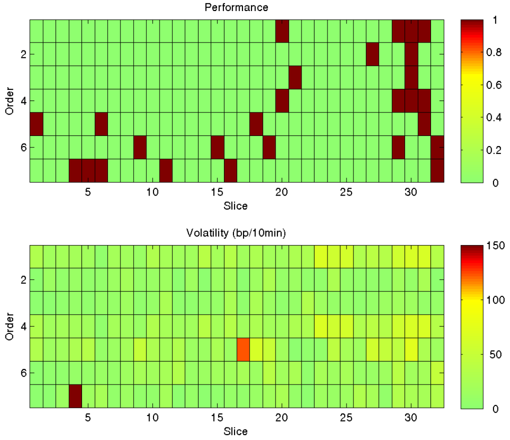
- We define some **efficiency criteria** Y_t (like performance) and some **potential explanatory variables** $X_t^1, \ldots, X_t^N$ (like a sector, an increase of volatility, a change in liquidity).
- On the fly (for instance every five minutes), we will **build predictors** $\phi(X) = \mathbb{E}(Y|X)$ of the current performance of all the trading algorithms of a trader using the sector, the volatility level, the liquidity, etc.
- The variables succeeding to explain bad performances will be said to be the **causes of bad performance**. That for, we will define the *predicting power* $\pi(t)$ of each variable X^i.

Performances and explanatory variables



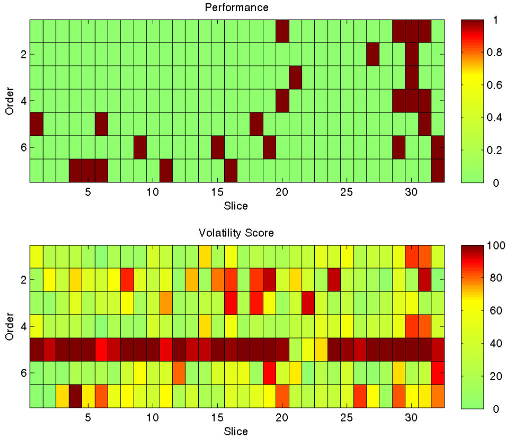
- We use the PnL (in bid ask spread) as a performance criterion;
- We use market descriptors: volatility (risk), bid-ask spread (liquidity), and momentum (directionality);
- We renormalize them using their *scores* (i.e. their empirical likelihood);
- We add patterns: price trends, price jumps and volume peaks.

Scoring



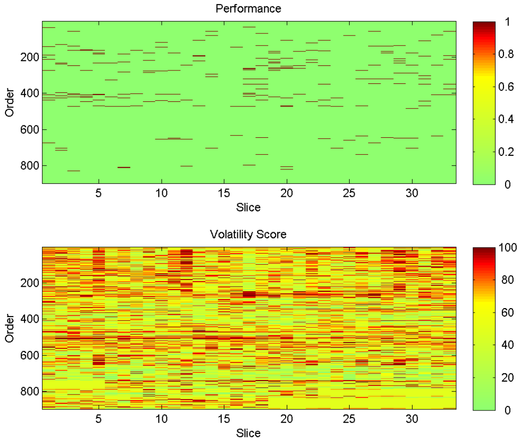
Scoring increases the “contrast” of the figure.

Scoring



Scoring increases the “contrast” of the figure. It is performed using the past values of the variable and using its empirical distribution function (roughly: replace *x* by its quantile).

Scoring



Scoring increases the “contrast” of the figure. It is performed using the past values of the variable and using its empirical distribution function (roughly: replace *x* by its quantile).

Binary prediction

- To be fast and take into account the number of possible predictors given the number of data,
- at each t, we select the 5% worst performances (i.e. Y is now zero or one) and try to explain them
- using two-sided binary predictors:

$$\phi(x) = \begin{cases} 0 & \text{if } x \in [\theta^-, \theta^+] \\ 1 & \text{otherwise} \end{cases}$$

- we choose the thresholds $(\theta^-(i), \theta^+(i))$ to obtain the best possible predictor for each X^i.

We have some guarantee

Generic Optimal Randomized Predictors

Fix a random vector $\mathbb{X} \in R^N$ of explanatory factors and a target binary variable Y. Let $0 \leq v(x) \leq 1$ be any Borel function of $x \in R^N$ such that $v(\mathbb{X}) = \Pr(Y = 1 \mid X)$ almost surely.
For any Borel decision function $\phi \in \Phi$, define the predictive power of the randomized predictor $\hat{Y}_\phi$ by $\pi(\phi) = Q(\mu, P^1(\phi), P^0(\phi))$, where Q is a fixed continuous and increasing function of the probabilities of correct decisions P^1, P^0 . Then there exists $\psi \in \Phi$ such that the predictor $\hat{Y}_\phi$ has maximum predictive power

$$\pi(\psi) = \max_{\phi \in \Phi} \pi(\phi)$$

Any such optimal Borel function $0 \leq \psi(x) \leq 1$ must almost surely verify, for some suitably selected constant $0 \leq c \leq 1$.

(5) $$\psi(X) = 1 \text{ for } v(X) > c \,; \quad \psi(X) = 0 \text{ for } v(X) < c.$$

Meaning that our two-sided predictors are not bad at all when it comes to do something simple. Moreover we have confidence intervals too (see in the paper).

Influence of a variable via a predictor

Influence of explanatory variables

We define the influence of $\mathfrak{X}$ a subset of explanatory variables as the **predictive power** of the best predictor:

$$\mathcal{I}_t(\mathfrak{X}, Y) = \pi(\psi) = \max_{\phi \in \Phi} \pi(\phi).$$

Remind we do not use the past of the variables X (except to build their *score* and for the *pattern matching* detectors).

We just rely here on the **joint distribution** of (Y, X) over all the instrument currently traded. It means we will use the states of all algorithms to try to establish a relation, now, between bad performances and variables of interest.

Simultaneous prediction as a clustering mechanism

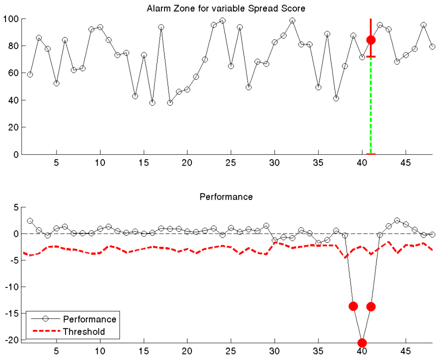
At the end of this process:

- at each update,
- we build optimal predictors and combinations of predictors explaining at most current bad performances.

- Implicitly we selected **hyperplanes in the space of combinations of our explanatory variables** separating trading algos with good perf. vs. bad ones.
- Some subsets of predictors are good (i.e. they allow hyperplanes to be efficiently positioned), others are not.
- This allows us to **identify variables currently influencing the performances**. They are said to be the **causes of bad performances**.
- We present to the trader the summarized information: "*sort by this variable if you want to understand what is happening to your algos*".

Monitoring results

Seen from one trading algo

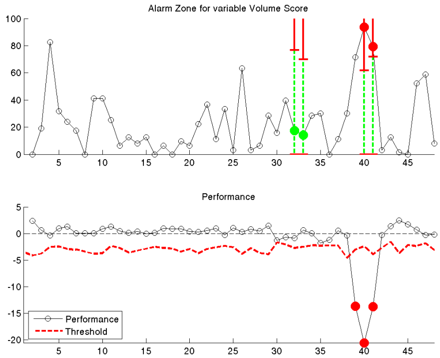


- top: the explanatory variable.
- bottom: the performance.

- The performance quantile is in dotted red; around update 40 this algo is 3 times among the 5% worst performers;
- on update 41, the spread score is selected by the good predictors to be used: $\theta^- = 0$, $\theta^+ \simeq 70\%$.

Monitoring results

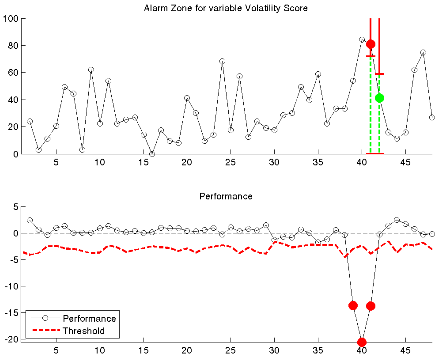
Seen from one trading algo



- top: the explanatory variable.
- bottom: the performance.
- The performance quantile is in dotted red; around update 40 this algo is 3 times among the 5% worst performers;
- around update 32, the volume score is selected to predict bad perf. of other algos.

Monitoring results

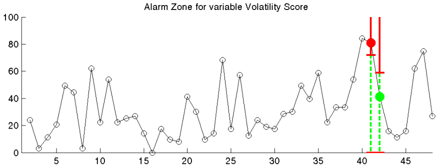
Seen from one trading algo

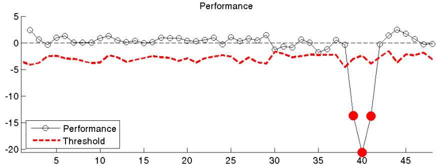


- top: the explanatory variable.
- bottom: the performance.
- The performance quantile is in dotted red; around update 40 this algo is 3 times among the 5% worst performers;
- the volatility score is selected at update 42, but the associated predictors says it is ok for this algo.

Monitoring results

Seen from one trading algo





- top: the explanatory variable.
- bottom: the performance.
- The performance quantile is in dotted red; around update 40 this algo is 3 times among the 5% worst performers;
- the volatility score is selected at update 42, but the associated predictors says it is ok for this algo.

Now you can tell traders the volatility is a potential source of bad performance of algos at time $t = 42$ (i.e. 12h30, Paris time) and point out the algorithms affected by this anomaly.

Outline

Conclusion

What is next?

We have seen two straightforward applications of machine learning to trading.

- **Online adaptiveness** is clearly a very nice features of ML techniques, that are very interesting to **digest and react to large flows of data**. Trading is clearly a good domain of application.
- **Humain monitoring** is needed in regulated applications of ML. You can replace dozens of trader by algorithms, but you always need to have humains monitoring them. **How to make sense of the outputs of hundreds or machines?** This is a challenge we need to answer to.

Thank You For Your Attention – Any Question?

References I

Agarwal, A., Bartlett, P. L., and Dama, M. (2010).
Optimal Allocation Strategies for the Dark Pool Problem.
In Teh, Y. W. and Titterington, M., editors, *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 9–16.

Azencott, R., Beri, A., Gadhyan, Y., Joseph, N., Lehalle, C.-A., and Rowley, M. (2014).
Realtime market microstructure analysis: online Transaction Cost Analysis.
Quantitative Finance, pages 0–19.

Doukhan, P. (1994).
Mixing: Properties and Examples (Lecture Notes in Statistics).
Springer, 1 edition.

Ganchev, K., Nevmyvaka, Y., Kearns, M., and Vaughan, J. W. (2010).
Censored exploration and the dark pool problem.
Commun. ACM, 53(5):99–107.

Hirsch, M. W. and Smith, H. (2005).
Monotone dynamical systems, volume 2, pages 239–357.

Kushner, H. J. and Yin, G. G. (2003).
Stochastic Approximation and Recursive Algorithms and Applications (Stochastic Modelling and Applied Probability) (v. 35).
Springer, 2nd edition.

Lamberton, D. and Pagès, G. (2008).
A penalized bandit algorithm.
Electronic Journal of Probability, 13(0).

Laruelle, S., Lehalle, C.-A., and Pagès, G. (2011).
Optimal split of orders across liquidity pools: a stochastic algorithm approach.
SIAM Journal on Financial Mathematics, 2:1042–1076.

References II

Lehalle, C.-A., Laruelle, S., Burgot, R., Pelin, S., and Lasnier, M. (2018).
Market Microstructure in Practice.
World Scientific publishing.

Pagès, G., Lapeyre, B., and Sab, K. (1990).
Sequences with low discrepancy. Generalization and application to Robbins-Monro algorithm.
Statistics, 21(2):251–272.

Robbins, H. and Monro, S. (1951).
A Stochastic Approximation Method.
The Annals of Mathematical Statistics, 22(3):400–407.