Dynamic portfolio management with transaction costs

Alberto Suárez[1,2], John Moody[2], Matthew Saffell[2]

(1) Computer Science Department

Universidad Autónoma de Madrid

(2) International Computer Science Institute, Berkeley

Objectives and methodology

- To design **goal-directed agents** that learn to discover **risk-averse strategies** in a **non-stationary environment** with **high levels of uncertainty**.
- **Reinforcement learning**: learn through **trial and error exploration** with only **limited feedback**.
- **Direct Reinforcement** (DR), or **policy gradient** methods enable an agent to discover useful strategies without the need to learn a value function.
- **Apply the methods developed** to an economically important problem: **dynamic portfolio management**.

Dynamic portfolio management

- **Allocate** an initial budget among different **risky assets**.
- The evolution of the asset prices is **uncertain**.
  - Difficult to forecast (efficient market hypothesis)
  - Non-stationary.
- **Multi-period investment problem**: The portfolio can be rebalanced by **trading assets** at fixed, regularly spaced, times, according to a **management strategy**.
- **Assets** can be bought and sold in arbitrary quantities at the posted price (no market impact).
- There are **transaction costs** [0.5 - 2 %].

GOAL: Design good self-financing strategies.

Experiments

- **Data:** Monthly Gross index (total returns without taxes in US $)
  December 31, 1969 - may 29, 2009 [474 months]

- **Source**: MSCI http://www.msci.com/

  - **Europe**: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom
  - **North America**: Canada, USA
  - **Pacific**: Australia, Hong kong, Japan, New Zealand, Singapore
  - **The world index**: Europe + North America + Pacific

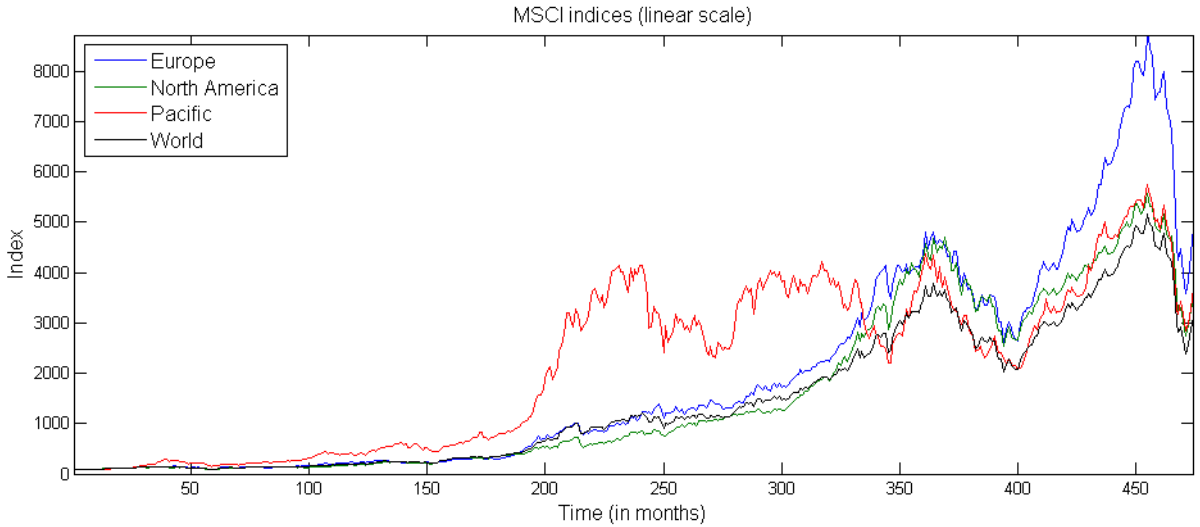
Time series of asset prices

$$S_0^{(i)}, S_1^{(i)}, S_2^{(i)}, \ldots, S_N^{(i)}; \qquad i = 1,2,\ldots,m$$

$$S_n^{(i)} \equiv S^{(i)}(t_n); \quad t_n = n\Delta T, \quad n = 1,2,\ldots,N; \quad T = N\Delta T.$$

E.g. MSCI Gross index (total returns without taxes in US $)

from December 31, 1969 to May 29, 2009, ΔT = 1 month



Time series of returns

$$r_1^{(i)}, r_2^{(i)}, \ldots, r_N^{(i)}; \quad i = 1,2,\ldots,m$$

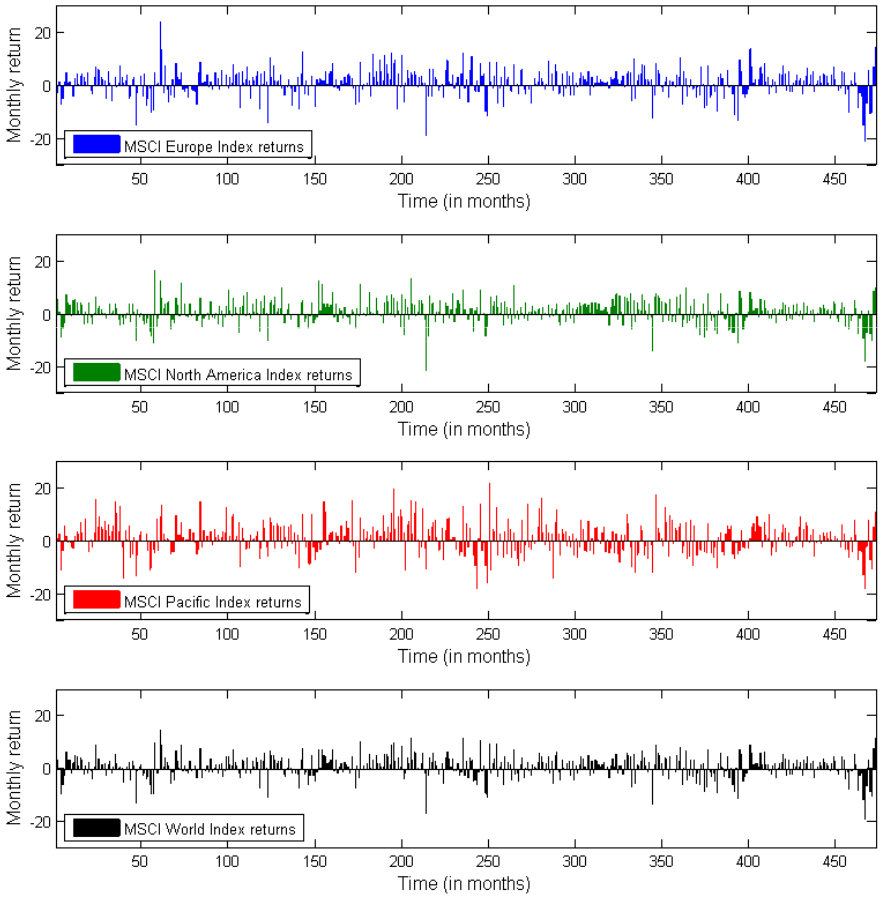
$$r_n^{(i)} \equiv \frac{S_n^{(i)} - S_{n-1}^{(i)}}{S_{n-1}^{(i)}} \quad n = 1,2,\ldots,N$$

$$S_n^{(i)} = S_{n-1}^{(i)}\left(1 + r_n^{(i)}\right) \quad \Rightarrow$$

$$S_n^{(i)} = S_0^{(i)} \prod_{i=1}^{n} \left(1 + r_n^{(i)}\right)$$

Time-series for returns are

- Quasi-stationary
- Short-time memory
- Heteroskedastic



Initial portfolio

- Assume we have an initial capital P_0.
- Allocate this capital among the *m* assets:

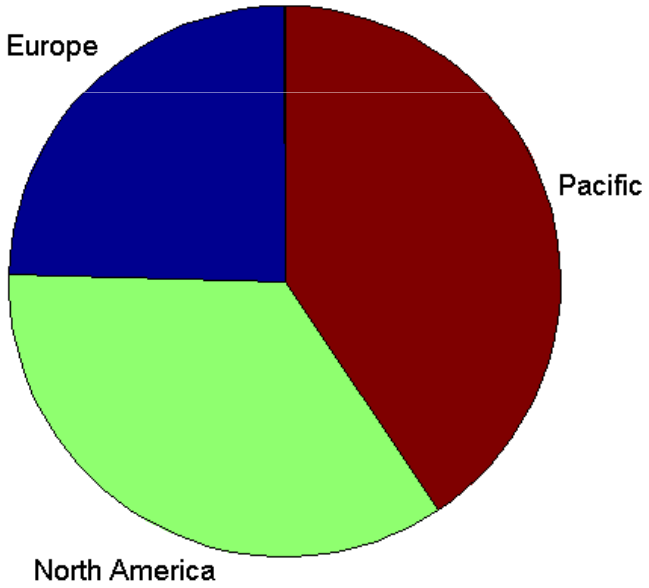
Portfolio composition:

$$\left\{c_0^{(i)}\right\}_{i=1}^{m}; \qquad \sum_{i=1}^{m} c_0^{(i)} S_0^{(i)} = P_0$$

Portfolio weights:

$$\left\{F_0^{(i)}\right\}_{i=1}^{m}; \qquad F_0^{(i)} = \frac{c_0^{(i)} S_0^{(i)}}{P_0}$$

$$\sum_{i=1}^{m} F_0^{(i)} = 1$$



Passive management: Market capitalization

- **The composition of the portfolio is held constant $[t_0, t_N]$**

Portfolio value:

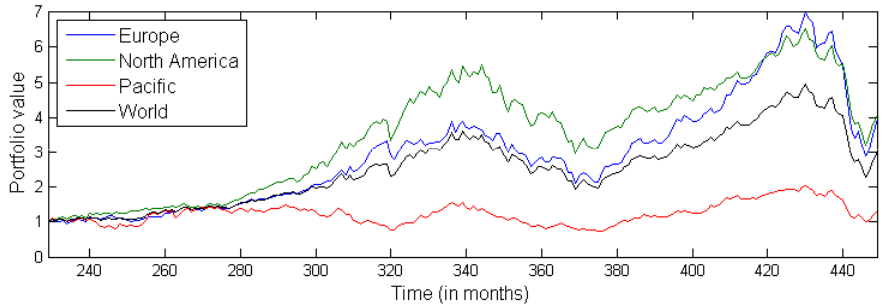
$$P_n = \sum_{i=1}^{m} c_0^{(i)} S_n^{(i)} = P_0 \sum_{i=1}^{m} F_0^{(i)} \frac{S_n^{(i)}}{S_0^{(i)}};$$

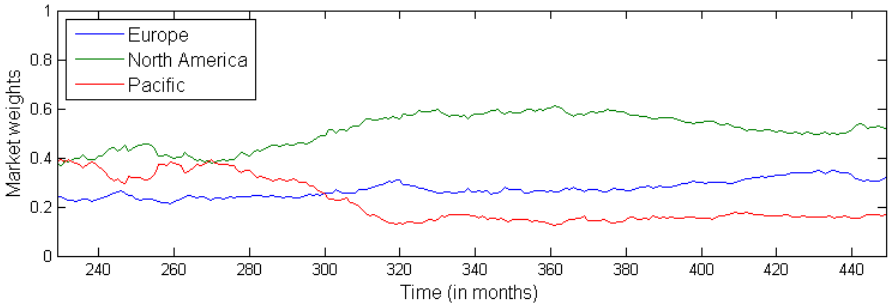
$$n = 1,2,\ldots,N$$

Portfolio weights:

$$F_n^{(i)} = \frac{c_0^{(i)} S_n^{(i)}}{P_n} = \frac{F_0^{(i)} \frac{S_n^{(i)}}{S_0^{(i)}}}{\sum_{j=1}^{m} F_0^{(j)} \frac{S_n^{(j)}}{S_0^{(j)}}};$$

$$\sum_{j=1}^{m} F_n^{(j)} = 1$$





Active management (no transaction costs)

- **The composition of the portfolio is held constant in $[t_{n-1}, t_n)$**

$$t = t_{n-1}: \quad P_{n-1} = \sum_{i=1}^{m} c_{n-1}^{(i)} S_{n-1}^{(i)}$$

Market capitalization

$$t = t_n^{-}: \quad P_n = \sum_{i=1}^{m} c_{n-1}^{(i)} S_n^{(i)}; \quad \widetilde{F}_n^{(i)} = \frac{c_{n-1}^{(i)} S_n^{(i)}}{P_n} = \frac{F_{n-1}^{(i)}\left(1 + r_n^{(i)}\right)}{\sum_{j=1}^{m} F_{n-1}^{(j)}\left(1 + r_n^{(j)}\right)}; \quad \sum_{j=1}^{M} \widetilde{F}_n^{(j)} = 1$$

- **The portfolio is rebalanced at t_n**

Portfolio rebalancing

$$t = t_n \qquad \left\{c_{n-1}^{(i)}; i = 1,2,\ldots,m\right\} \Rightarrow \left\{c_n^{(i)}; i = 1,2,\ldots,m\right\}$$

$$P_n = \sum_{i=1}^{m} c_{n-1}^{(i)} S_n^{(i)} = \sum_{i=1}^{m} c_n^{(i)} S_n^{(i)}; \qquad F_n^{(i)} = \frac{c_n^{(i)} S_n^{(i)}}{P_n} \qquad \sum_{j=1}^{M} F_n^{(j)} = 1$$

Budget constraint

Portfolio value (without transaction costs)

- **Value of the portfolio at t_n**

$$P_n = P_{n-1}(1+r_n); \qquad n = 1,2,\ldots,N$$

with $$r_n = \sum_{i=1}^{m} F_{n-1}^{(i)} r_n^{(i)}$$

- **Value of the portfolio at $t_N = T$**

$$P_N = P_0 \prod_{n=1}^{N} (1+r_n)$$

Portfolio value (with transaction costs)

$$t = t_n^- : \ \tilde{P}_n = P_{n-1}(1+r_n) = P_{n-1}\left(1+\sum_{i=1}^{m} F_{n-1}^{(i)} r_n^{(i)}\right) = \sum_{i=1}^{m} c_{n-1}^{(i)} S_n^{(i)};$$

Before rebalancing

$$t = t_n : \ \ P_n = \sum_{i=1}^{m} c_n^{(i)} S_n^{(i)};$$

After rebalancing

$$\text{costs}_n = \delta \sum_{i=1}^{m} \left| c_n^{(i)} S_n^{(i)} - c_{n-1}^{(i)} S_n^{(i)} \right| = \delta \sum_{i=1}^{m} \left| P_n F_n^{(i)} - \tilde{P}_n \tilde{F}_n^{(i)} \right| \quad =$$

$$= \delta \tilde{P}_n \sum_{i=1}^{m} \left| \frac{P_n}{\tilde{P}_n} F_n^{(i)} - \tilde{F}_n^{(i)} \right| = \delta \tilde{P}_n \sum_{i=1}^{m} \left| F_n^{(i)} - \tilde{F}_n^{(i)} \right| + O\left(\delta^2\right)$$

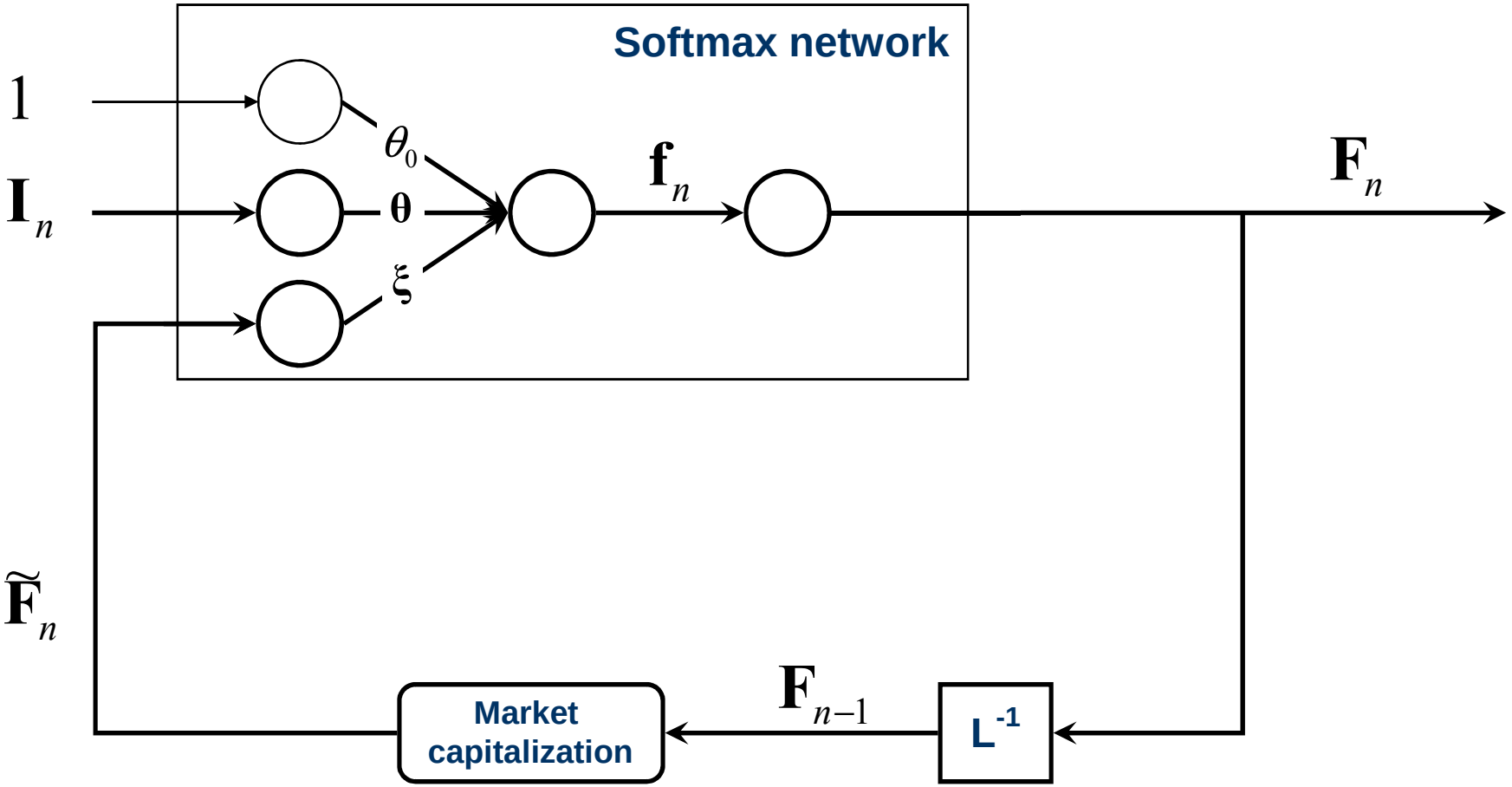
$$P_n = \tilde{P}_n - \text{costs}_n \approx \tilde{P}_n \left(1 - \delta \tilde{P}_n \sum_{i=1}^{m} \left| F_n^{(i)} - \tilde{F}_n^{(i)} \right| \right)$$

Budget constraint

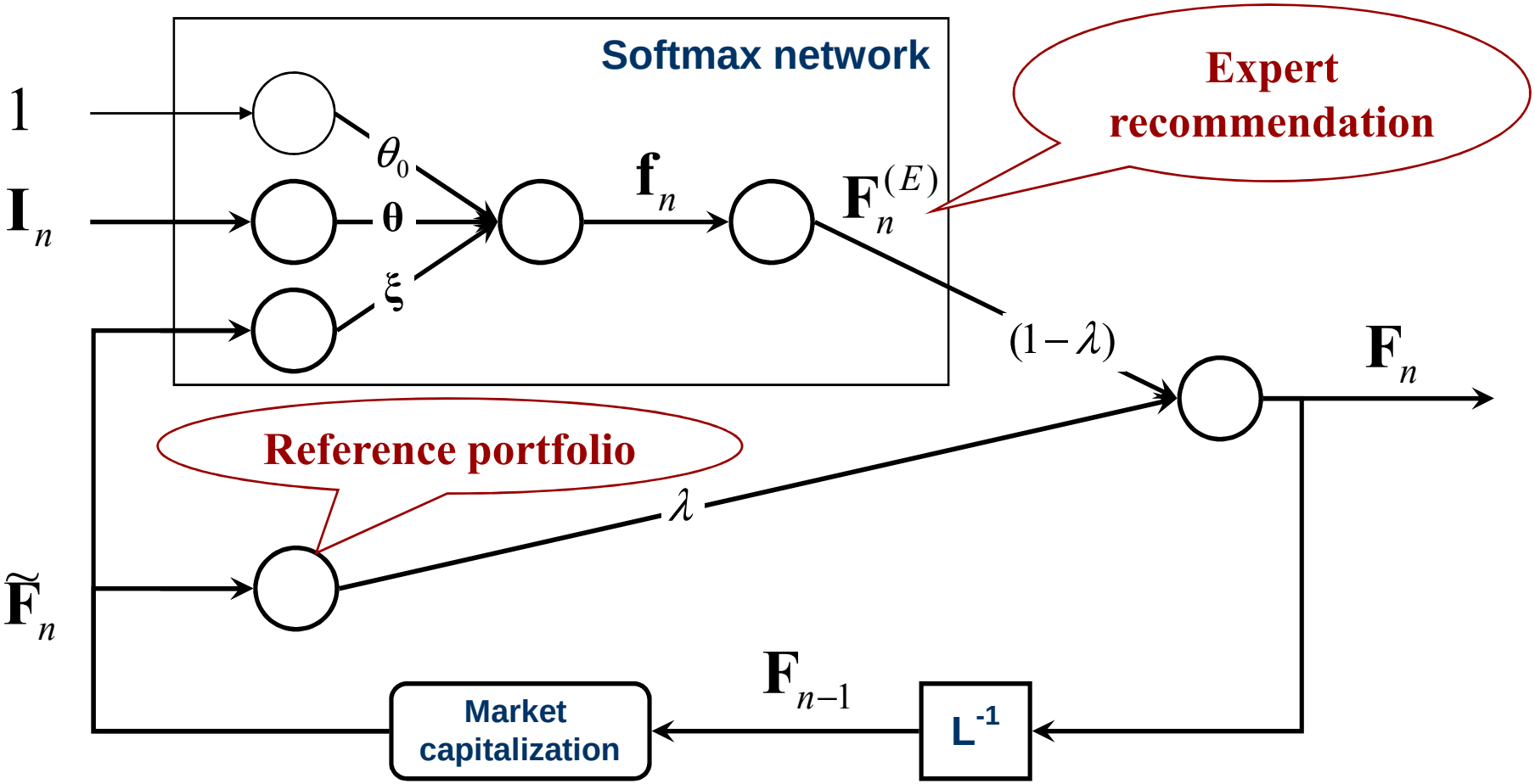
$$P_n \approx P_{n-1}(1+r_n)\left(1-\delta \sum_{i=1}^{m} \left| F_n^{(i)} - \tilde{F}_n^{(i)} \right| \right) = P_{n-1}\left(1+\hat{r}_n\right) \Rightarrow P_N = P_0 \prod_{n=1}^{N} \left(1+\hat{r}_n\right)$$

$$\left(1+\hat{r}_n\right) \equiv (1+r_n)\left(1-\delta \sum_{i=1}^{m} \left| F_n^{(i)} - \tilde{F}_n^{(i)} \right| \right)$$

Recurrent RL architecture (without transaction costs)



Recurrent RL architecture (with transaction costs)



RRL architecture (without transaction costs)

- Use a softmax representation for the portfolio weights
  - Inputs:
    - Non-recurrent: Constant bias term (1) + exogenous variables: $\underline{I_n}$
    - Recurrent (delay 1): Delayed values of the portfolio weights updated by market capitalization
  - Outputs: Portfolio weights

$$\mathbf{w}^{(i)} \equiv \left[\theta_0^{(i)} \quad \boldsymbol{\theta}^{(i)} \quad \boldsymbol{\xi}^{(i)}\right]$$

$$f_n^{(i)}\left(\tilde{\mathbf{F}}_n, \mathbf{I}_n; \mathbf{w}^{(i)}\right) = \theta_0^{(i)} + \boldsymbol{\theta}^{(i)} \cdot \mathbf{I}_n + \boldsymbol{\xi}^{(i)} \cdot \tilde{\mathbf{F}}_n; \quad i = 1,2,\ldots,m.$$

$$F_n^{(i)} = \text{softmax}\left[\mathbf{f}_n\left(\tilde{\mathbf{F}}_n, \mathbf{I}_n; \mathbf{w}\right)\right] = \frac{\exp\left\{f_n^{(i)}\left(\tilde{\mathbf{F}}_n, \mathbf{I}_n; \mathbf{w}^{(i)}\right)\right\}}{\sum_{j=1}^{m} \exp\left\{f_n^{(j)}\left(\tilde{\mathbf{F}}_n, \mathbf{I}_n; \mathbf{w}^{(j)}\right)\right\}}; \quad \sum_{i=1}^{m} F_n^{(i)} = 1$$

RRL architecture (with transaction costs)

- **Use regularization hyperparameter $0 \le \lambda \le 1$**

$$\mathbf{F}_n = \lambda \tilde{\mathbf{F}}_n + (1-\lambda)\mathbf{F}_n^{(E)}$$

$$\tilde{\mathbf{F}}_n = \text{current portfolio weights}$$

$$\mathbf{F}_n^{(E)} = \text{softmax}\left[\mathbf{f}_n\left(\tilde{\mathbf{F}}_n, \mathbf{I}_n; \mathbf{w}\right)\right]$$

$\lambda = 1$ Buy and hold strategy

$\lambda = 0$ Use the recommendation of the softmax network ignoring the composition of the current portfolio

Measures of performance: Portfolio value

- Learning parameter ρ

$$P_n = P_{n-1}\left(1+\hat{r}_n\right)$$

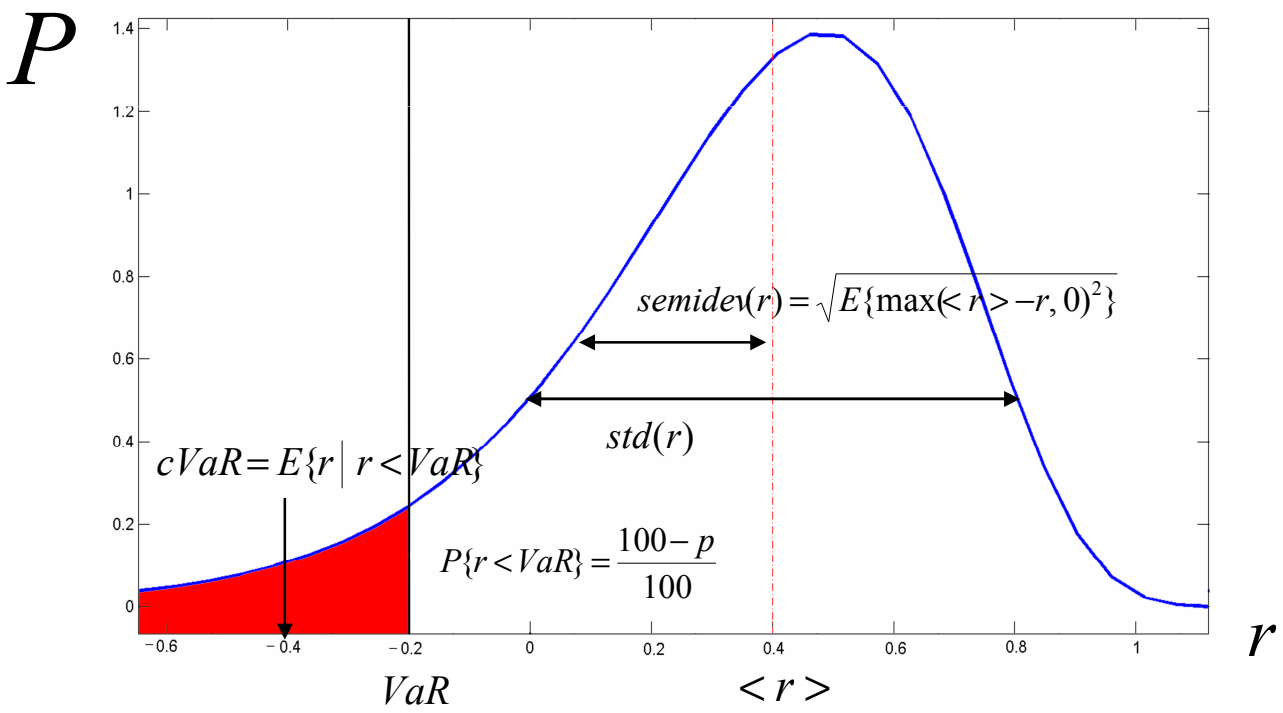
$$\left(1+\hat{r}_n\right)=\left(1+\sum_{i=1}^{m} F_{n-1}^{(i)} r_n^{(i)}\right)\left(1-\delta\sum_{i=1}^{m}\left|F_n^{(i)}-\widetilde{F}_n^{(i)}\right|\right); \qquad \widetilde{F}_n^{(i)}=\frac{F_{n-1}^{(i)}\left(1+r_n^{(i)}\right)}{\sum_{j=1}^{m} F_{n-1}^{(j)}\left(1+r_n^{(j)}\right)};$$

$$\max\left[P_n \mid P_{n-1}\right]=\max\left[\hat{r}_n\right] \quad \Rightarrow \qquad \mathbf{w}_{n+1}=\mathbf{w}_n+\rho\left.\frac{d\hat{r}_n}{d\mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_n};$$

Needs $\dfrac{d\mathbf{F}_{n-1}}{d\mathbf{w}}; \dfrac{d\mathbf{F}_n}{d\mathbf{w}}$ (Recurrent learning)

Measures of risk

- Measures of risk: stdev, Semideviation, VaR, ShortFall (conditional VaR), etc.



Balancing risk and performance: Sharpe ratio

- The Sharpe ratio is a risk-adjusted measure of performance.

$$SR = \frac{E[\hat{r}_n]}{stdev[\hat{r}_n]}$$

- For online optimization it is preferable to use the differential Sharpe-ratio

Exponential moving average Sharpe ratio

- Running average SR

$$A_n = \frac{1}{n}\sum_{k=1}^{n} \hat{r}_n; \qquad A_n = \frac{1}{n}\hat{r}_n + \frac{n-1}{n} A_{n-1}$$

$$B_n = \frac{1}{n}\sum_{k=1}^{n} \hat{r}_n^{\,2}; \qquad B_n = \frac{1}{n}\hat{r}_n^{\,2} + \frac{n-1}{n} B_{n-1}$$

$$SR_n = \frac{A_n}{K_n\left(B_n - A_n^{\,2}\right)}; \qquad K_n = \left(\frac{n}{n-1}\right)^{1/2}$$

- Exponential moving average SR

$$A_n = \eta\,\hat{r}_n + (1-\eta)A_{n-1}; \qquad B_n = \eta\,\hat{r}_n^{\,2} + (1-\eta)B_n$$

$$SR_n = \frac{A_n}{K_\eta\left(B_n - A_n^{\,2}\right)}; \quad K_\eta = \left(\frac{1-\eta/2}{1-\eta}\right)^{1/2}; \qquad \eta \to 0^+$$

Differential Sharpe ratio

$$A_n = \eta\,\hat{r}_n + (1-\eta)A_{n-1}; \qquad B_n = \eta\,\hat{r}_n^{\,2} + (1-\eta)B_{n-1}$$

$$SR_n = SR_{n-1} + \eta \left.\frac{dSR_n}{d\eta}\right|_{\eta=0} + O(\eta^2);$$

$$D_n = \left.\frac{dSR_n}{d\eta}\right|_{\eta=0} = \frac{B_{n-1}\,\Delta A_{n-1} - \frac{1}{2}A_{n-1}\,\Delta B_{n-1}}{\left(B_{n-1} - A_{n-1}^{\,2}\right)^{3/2}};$$

$$\Delta A_n = \hat{r}_n - A_{n-1}; \qquad \Delta B_n = \hat{r}_n^{\,2} - A_{n-1}$$

Maximizing the Differential Sharpe ratio

$$D_n = \left.\frac{dSR_n}{d\eta}\right|_{\eta=0} = \frac{B_{n-1}\left(\hat{r}_n - A_{n-1}\right) - \frac{1}{2}A_{n-1}\left(\hat{r}_n^{\,2} - A_{n-1}\right)}{\left(B_{n-1} - A_{n-1}^{\,2}\right)^{3/2}};$$

$$\frac{dD_n}{d\mathbf{w}} = \frac{dD_n}{d\hat{r}_n}\frac{d\hat{r}_n}{d\mathbf{w}} = \frac{B_{n-1} - A_{n-1}\hat{r}_n}{\left(B_{n-1} - A_{n-1}^{\,2}\right)^{3/2}}\frac{d\hat{r}_n}{d\mathbf{w}}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \rho\left.\frac{dD_n}{d\mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_n} = \mathbf{w}_n + \rho\frac{B_{n-1} - A_{n-1}\hat{r}_n}{\left(B_{n-1} - A_{n-1}^{\,2}\right)^{3/2}}\left.\frac{d\hat{r}_n}{d\mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_n}$$

Learning protocol

- **Input indicators**

  Delayed moving averages for [3 6 12 24] months

- **Training / testing for RRL**

  {1} Learn weights in window: $[t_n - t_{Learn}, t_n - 1]$;

  {2} Test on single point: t_n

  - Optimize either the portfolio value or the differential Sharpe ratio.
  - Fix the values of the differential Sharpe-ratio parameter η, of the learning parameter ρ and of the regularization parameter λ by cross-validation.
  - Learn the policy using $t_{Learn} = 120$ months (10 years).
  - Use 2/3 of the data (training set = 80 points) for weight optimization and 1/3 of the data (validation set = 40 points) for early stopping.
  - Average the output of 10 networks trained on different random partitions of the data available for learning into training and validation sets.

Results (without costs)

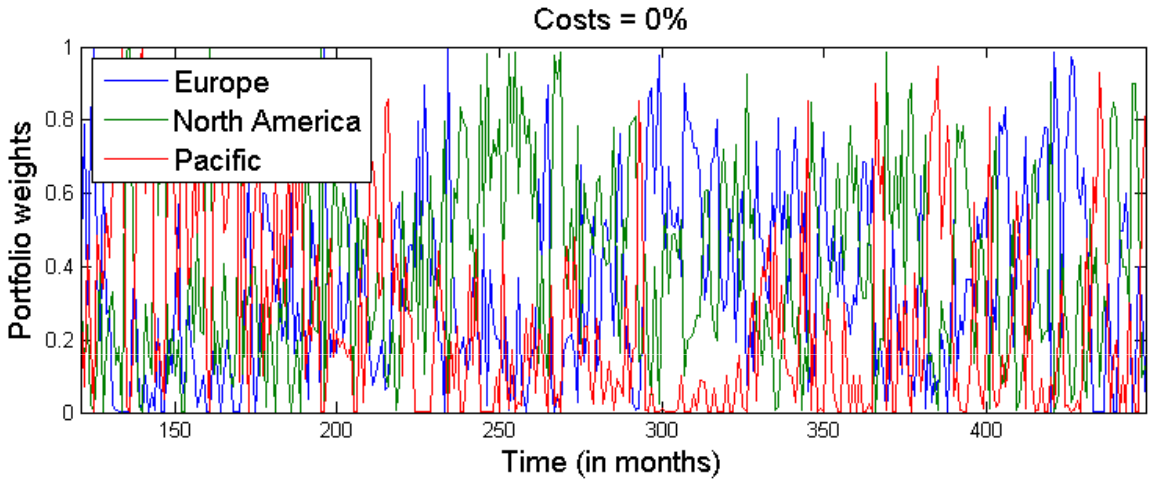
Performance on the last 220 months

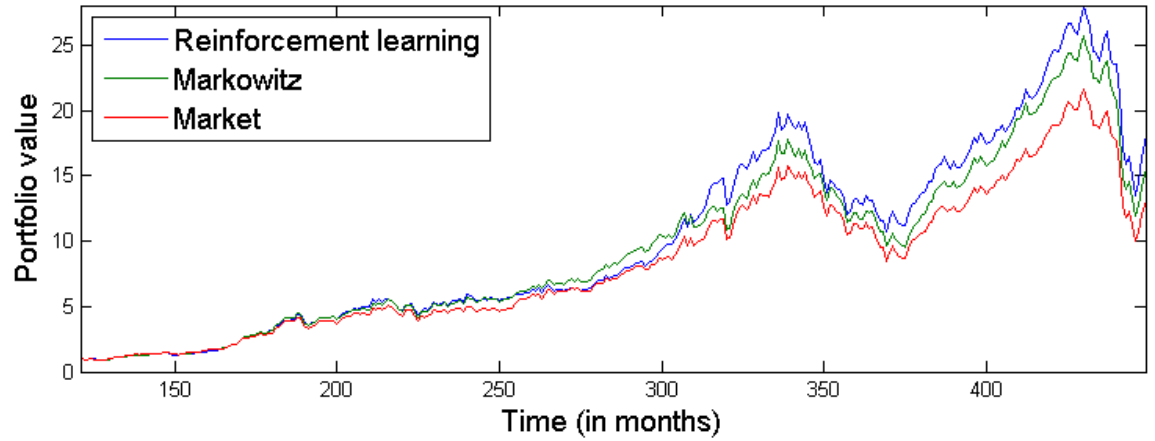
Profit

RRL:	3.6710
Markowitz :	3.2559
Market:	3.0001

Sharpe ratio

RRL:	0.5506
Markowitz:	0.5092
Market:	0.4773





Results (with costs)

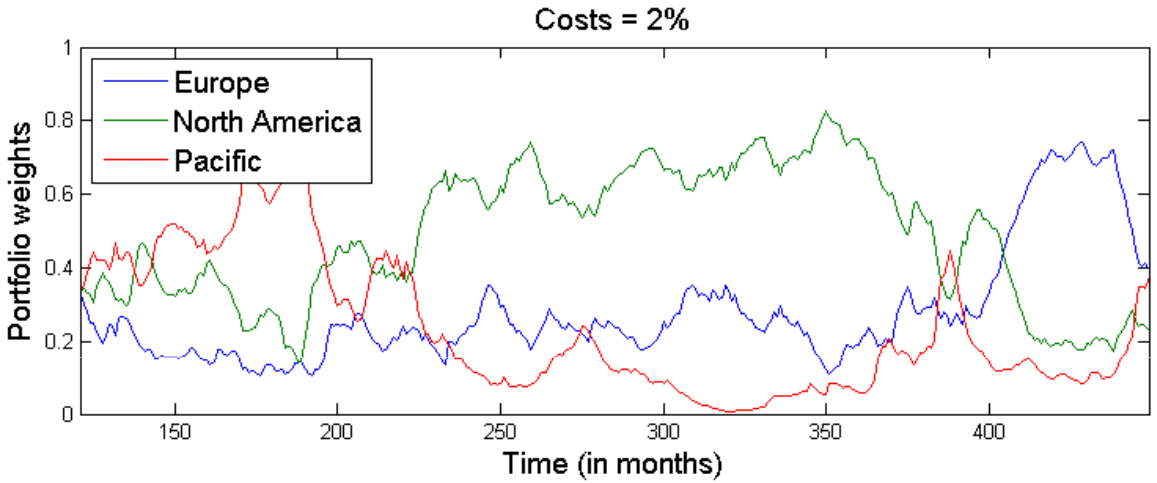
Performance on the last 220 months

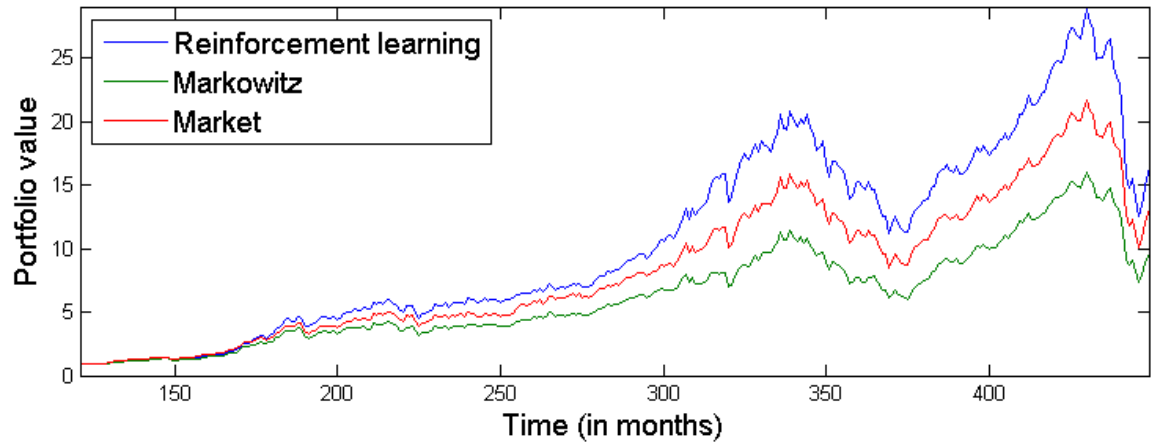
Profit

RRL:	3.2120
Markowitz :	2.7349
Market:	3.0001

Sharpe ratio

RRL:	0.4969
Markowitz:	0.4459
Market:	0.4773





Conclusions (I)

- **Without transaction costs**
  - It is possible to discover active management strategies that outperform the market.
  - Most of these strategies exhibit **switching behavior,** which is undesirable from the point of view of robustness and stability.
  - The frequent and large transfers of capital among assets makes it difficult to implement the investment policy.
- **With transaction costs**
  - It is **more difficult** to design strategies that beat the market portfolio
  - It is important to use a **combined strategy** that uses
    - A pool of learned strategies (**recurrent reinforcement learning**)
    - A **reference strategy** with a sufficiently good performance
      We have used passive management as a reference.
      Can other strategies be used (e.g. log-optimal, based on forecasts, etc.)?
    - Investment policies are **smooth** (easier to manage) & **more diversified**
- More domain-specific information is needed to design successful strategies!

Conclusions (II)

- **Recurrent Reinforcement Learning** (RRL) provides a computationally efficient solution to the problem of **dynamic financial asset allocation**
  - **RRL** is an **adaptive policy search** algorithm that can **learn** a strategy **on-line** in a non-stationary environment that is uncertain and/or risky.
  - It is not based on forecasts and does not require learning a value function.
  - Optimization of risk-adjusted measures of performance (e.g. Sharpe ratio).
  - Takes into account transaction costs.
- The **investment strategies** discovered by the **Direct Reinforcement Agent** are
  - Interpretable and robust.
  - Well-diversified.
  - Avoid switching-behavior.
- The methods developed can be employed in **other areas of application** characterized by large uncertainty and a changing environment, in which a risk-averse behavior is desirable: design of policies (e.g. energy), robotics, autonomous vehicles, industrial control, telecommunications, etc.