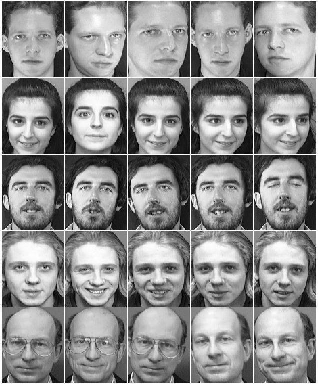
Application of Machine Learning to Finance

Zélia Cazalet & Tung-Lam Dao

ASSET MANAGEMENT BY
LYXOR

Introduction

Figure: A subset of the database



Introduction

Figure: PCA of faces

Introduction

Figure: ICA of faces

Outline

1. Hedge fund replication: factor selection and the lasso method
2. Nonnegative matrix factorization
3. Learning algorithms
4. Trend forecasting with L_1 and L_2 filterings
5. Support Vector Machine and financial applications

Hedge Fund replication

It is principally done using factor-based models: **rolling least squares** or **Kalman filtering algorithms**.

HF replication

$$R_t^{\mathrm{HF}} = \sum_{i=1}^{m} \beta_{i,t} R_t^i + \varepsilon_t$$

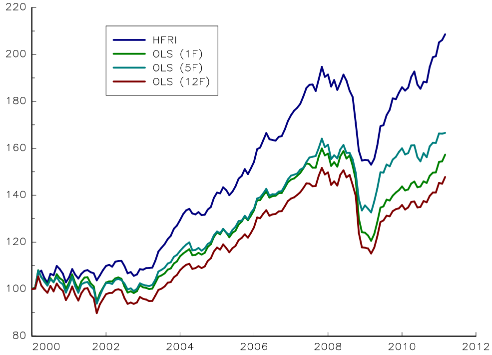
Define the tracker portfolio as:

$$R_{t+1}^{\mathrm{Tracker}} = \sum_{i=1}^{m} \beta_{i,t} R_{t+1}^i$$

Problem of factor selection

Considering the problem of factor selection is necessary: the universe of factor selection influences the tracker's performance. A solution: **the lasso method**.

Trackers with different universes of factors



Lasso regression (Tibshirani, 1996)

It corresponds to a linear regression with regularization of coefficient estimates: L^1 **norm constraint of exposures**.

Lasso regression

After the standardization of returns, we have:

$$\begin{aligned} \hat{\beta} &= \arg\min \left(R^{\mathrm{HF}} - R\beta\right)^\top \left(R^{\mathrm{HF}} - R\beta\right) \\ &\quad \text{u.c. } \sum_{i=1}^{m} \beta_i^2 \leq \tau^\star \end{aligned}$$

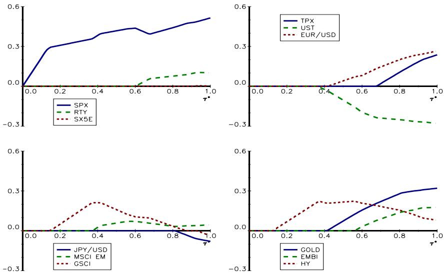
where $\tau^\star$ is the shrinkage measure of the lasso model with respect to the OLS model.

Ranking of factors

Ranking of the lasso exposures (Feb. 28, 2011)

1. SPX	2. HY	3. GSCI	4. UST	5. MSCI EM	6. EUR/USD
7. GOLD	8. EMBI	9. RTY	10. TPX	11. JPY/USD	12. SX5E

Factors selection (Feb. 28, 2011)



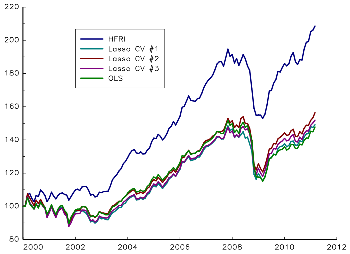
Cross-validation procedure

We define an **out-of-sample procedure** to choose the optimal value of $\tau^\star$.

Principle

1. We build training and test samples from the lag window p.
2. For one sequence of different $\tau^\star \in [0, 1]$, we estimate the exposures $\beta_{i,t}$ on the training sample.
3. We compute a statistic of interest on the test sample: performance, TE or MSE.
4. We find the value of $\tau^\star$ which permits to optimize the statistic of interest.

Trackers with cross-validation lasso regression



Results of replicating the HFRI index using different methods

Model	μ	σ	sh	$\mathcal{MDD}$	π_{AB}	σ_{TE}	ρ
HFRI	6.80	6.81	0.57	21.42			
CV #1	3.64	7.59	0.09	22.32	71.50	3.52	0.89
CV #2	4.09	7.77	0.15	21.56	74.99	3.29	0.91
CV #3	3.81	7.68	0.11	20.20	72.82	3.43	0.89
OLS	3.56	7.66	0.08	24.07	70.85	3.51	0.89

NMF principle and financial interpretation

NMF is an alternative approach to decomposition methods like PCA and ICA with the special feature to consider nonnegative matrices:

NMF decomposition

Let A be a nonnegative matrix $m \times p$:

$$A \approx BC$$

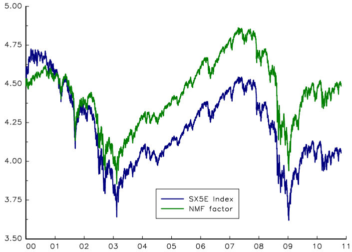
with B and C nonnegative matrices of dimensions $m \times n$ and $n \times p$.

Considering a variable/observation storage in A, interpret B as a matrix of weights called **loading matrix** and C as a **factor matrix**.

Factor extraction of an equity universe

Using the composition at the end of 2010, we compute NMF on the logarithm of the stock prices.

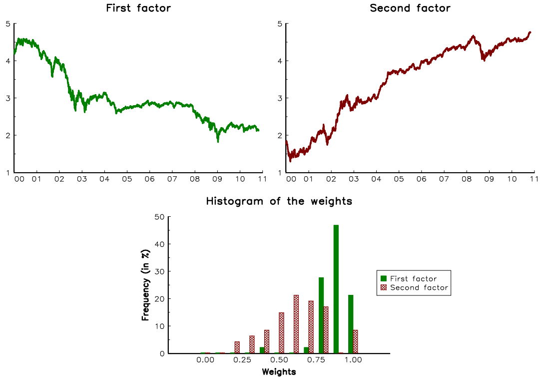
Comparison between the EuroStoxx 50 and the first NMF factor



The first NMF factor is highly correlated with the index.

Factor extraction of an equity universe

NMF with two factors



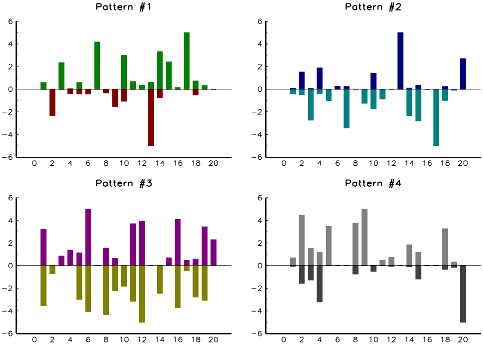
We may interpret them as a factor of bear market and a factor of bull market

Pattern recognition of asset returns

Data: weekly returns of 20 stocks.

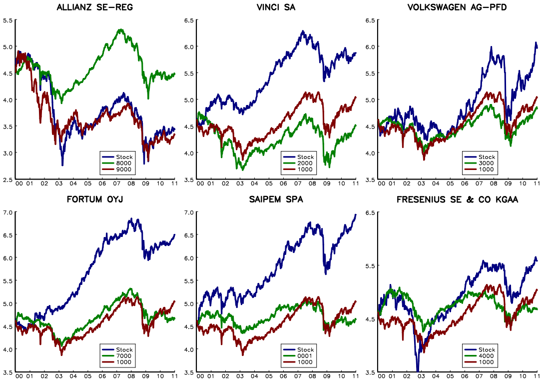
Period: January 2000 - December 2010.

NMF on positive and negative returns (four patterns)



Stock classification

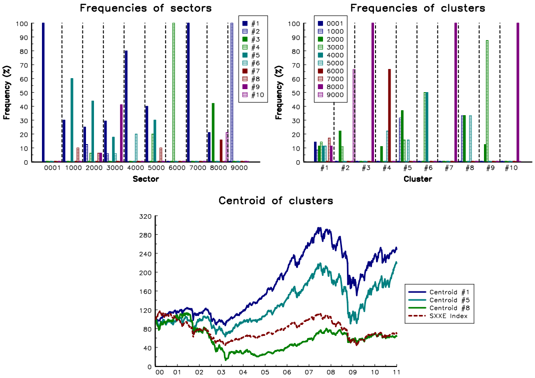
Some stocks are more sensible to the representative NMF factor than to their corresponding sectors.



Classification of stocks: NMF classifiers

Apply the K-means procedure directly on the stocks returns.

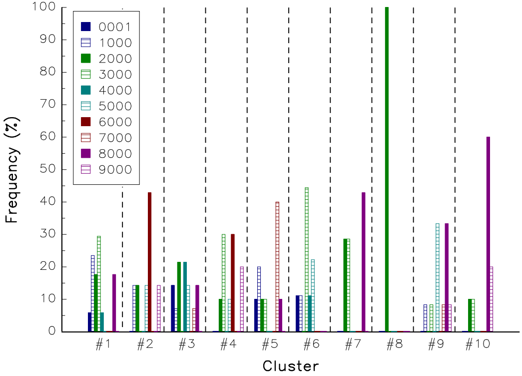
Results of the cluster analysis



Classification of stocks: NMF classifiers

Can NMF classifiers represent an alternative sector classification?

Frequencies of sectors in each cluster



Bagging and Boosting algorithms

- Bagging and boosting algorithms are recent powerful techniques which permit to reduce the error of any learning algorithms.
- These two methods consist in determining several classifiers before aggregating them by voting.

Difference between the two algorithms

- bagging uses bootstrap samples to construct classifiers,
- boosting adjusts the weights of the training instances considering errors of classification.

Application to stock picking: scores

We work on the improvement of a score used in a stock picking model. We use the current score based on a discrete optimization and a score built with a probit model.

Probit score

$$S = \Phi(X^\top \beta + \alpha)$$

with $\Phi(x)$ the cumulative distribution function of the standard normal distribution and (α, β) two vectors estimated using the estimator of the maximum likelihood.

Application to stock picking: index tilting

The objective of index tilting is to maximize the score of the portfolio compared to the score of a benchmark. This optimization is under constraint of tracking error.

Optimization problem

$$x^\star = \arg\max \, (x - b)^\top s$$

$$\text{u.c. } 1^\top x = 1^\top b = 1 \text{ and } \sigma \le \sigma^\star$$

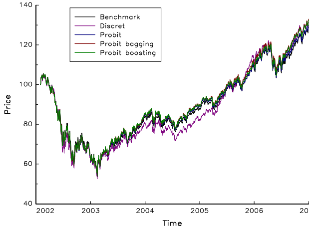
with:

$$\sigma^2 = (x - b)^\top \Sigma \, (x - b)$$

where x and b are respectively the portfolio and the benchmark weights, s is the vector of score, Σ the variance-covariance matrix of stocks and $\sigma^\star$, the constraint of tracking error.

Application to stock picking: backtests

Backtests of the stock picking model (2002-2006)

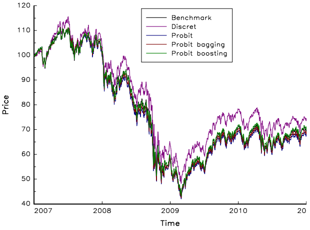


Reporting of the stock picking model (2002-2006)

Models	μ	σ	sh	$\mathcal{MDD}$	IR	σ_{TE}	ρ
Benchmark	5.34	20.61	0.26	48.76			
Discret Score	5.74	21.38	0.27	50.01	0.09	4.67	0.98
Probit Score	5.51	20.57	0.27	49.25	0.09	1.91	0.99
Probit Score bagging	5.92	20.59	0.29	48.86	0.33	1.73	0.99
Probit Score boosting	6.00	20.57	0.29	49.07	0.33	1.98	0.99

Application to stock picking: backtests

Backtests of the stock picking model (2007-2011)



Reporting of the stock picking model (2007-2011)

Models	μ	σ	sh	$\mathcal{MDD}$	IR	σ_{TE}	ρ
Benchmark	-7.71	27.30	-0.28	61.04			
Discret Score	-6.06	28.50	-0.21	58.27	0.29	5.63	0.98
Probit Score	-8.36	27.09	-0.31	62.18	-0.23	2.80	0.99
Probit Score bagging	-7.46	27.12	-0.27	61.11	0.10	2.42	0.99
Probit Score boosting	-8.09	27.10	-0.30	61.84	-0.14	2.70	0.99

Trend filtering

- Noisy signal y_t can be decomposed into trend x_t and noise z_t:

$$y_t = x_t + z_t$$

- L_2 filter (Hodrick-Prescott filter) detects x_t by minimizing:

$$\frac{1}{2}\|y - x\|_{L_2}^2 + \lambda \|Dx\|_{L_2}^2$$

with second derivative D:

$$D = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & -2 & 1 \end{bmatrix}$$

- L_2 filter allows explicit solution $x^\star = \left(I + 2\lambda D^\top\right)^{-1} y$

L_1 filtering

- Minimize the objective function with L_1 pernalty:

$$\frac{1}{2}\|y - x\|_{L_2}^2 + \lambda \|Dx\|_{L_1}$$

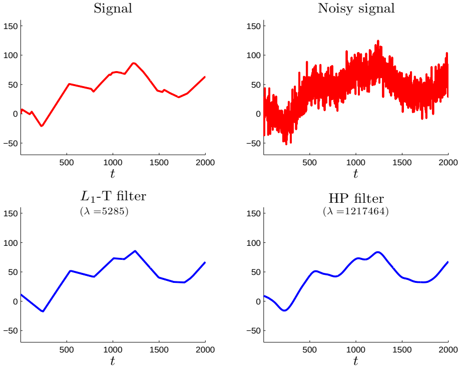
where D is discrete form of the first or second derivative.

- Similar problems: Lasso regression (Tibshirani, 1996) or the L_1 regularized least square problem (Daubechies, 2004)
- Properties of L_1 filtering:
  - Using L_1 norm $\Rightarrow$ 2^{nd} derivation of x_t must be zero.
  - L_1 norm allows x_t change the trend without two much cost.
  - Trade-off between: residual noise and number of breaks.
  - Determine λ by minimizing prediction error within caliration procedure.

Linear trend model

Stochastic linear trend

$$\begin{cases} y_t = x_t + z_t \\ z_t \sim \mathcal{N}\left(0, \sigma^2\right) \\ x_t = x_{t-1} + v_t \\ \Pr\left\{v_t = v_{t-1}\right\} = 1 - p \\ \Pr\left\{v_t = b\mathcal{U}_{[-1,1]}\right\} = p \end{cases}$$



Remarks

- L_1 filter gives hidden trend
- Direct trend prediction

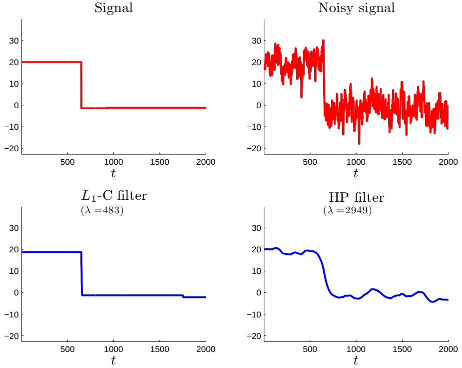
Ornstein-Uhlenbeck process

OU with switching regim

$$
\begin{cases}
y_t = y_{t-1} + \theta(\mu_t - y_{t-1}) + z_t \\
z_t \sim \mathcal{N}\left(0, \sigma^2\right) \\
\Pr\left\{\mu_t = \mu_{t-1}\right\} = 1 - p \\
\Pr\left\{\mu_t = b\mathcal{U}_{[-1,1]}\right\} = p
\end{cases}
$$

Remarks

- L_1 is better than L_2
- simple for application

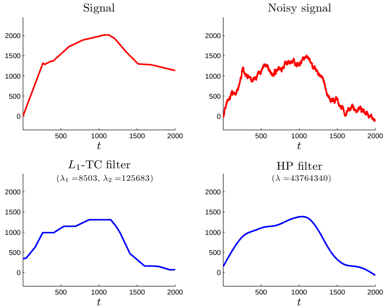


Mixing trend and mean-reverting

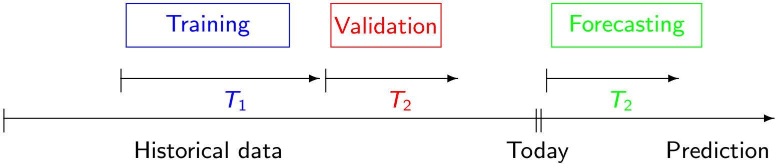
Use two penalty conditions:

$$\frac{1}{2}\|y-x\|_2^2+\lambda_1\|D_1x\|_1+\lambda_2\|D_2x\|_1$$

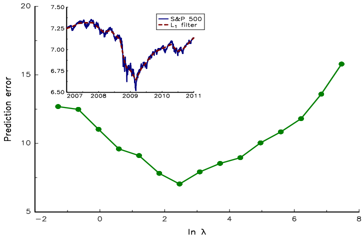
D_1 and D_2 are respectively the 1^{st} and 2^{nd} derivatives.



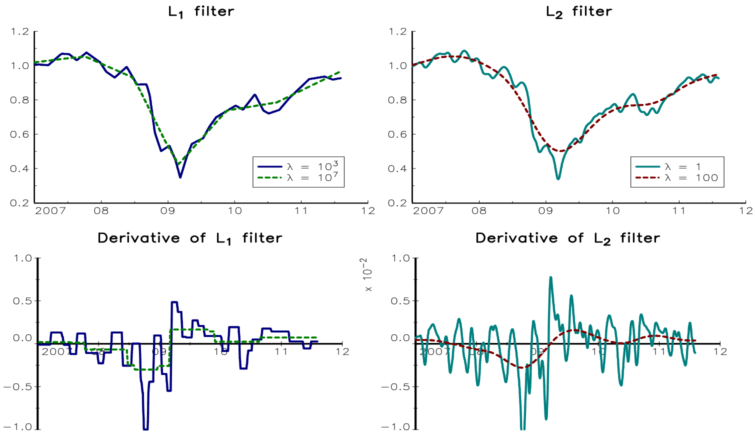
Cross validation: Algorithm



procedure CV_FILTER(T_1, T_2)
Compute an array of $(\lambda_n^{\max})$ of N training sets T_1
Compute $\bar{\lambda}$, $\Delta\lambda$ the average and variance of (λ_n)
Compute $\lambda_1 = \bar{\lambda} + \Delta\lambda$ and $\lambda_2 = \bar{\lambda} - \Delta\lambda$
for $i = 1 : N_p$ **do**
Compute $\lambda_i = \lambda_2 \left(\lambda_2 / \lambda_1\right)^{(i/N_p)}$
Scan data by the window T_1
Compute the total error $e(\lambda_i)$
end for
Minimize the error $e(\lambda)$ to find the optimal value $\lambda^\star$
Run the L_1 filter with $\lambda = \lambda^\star$
end procedure



Comparison between L_1 and L_2 filters



History and Financial applications

History

- SVM first introduced in 1992 as classification method
- SVM next interpreted as regression technique (Vapnik 1998)
- SVM applications in various fields: pattern recognition, bioinformation

Financial applications

- SVM score: Score Binary classification
- SVM sector recognition: supervision method to classify stocks
- SVM filtering: trend extraction
- SVM multi-regression: trend prediction based on multi-factors

Principle and Score construction

Example of SVM via the score construction

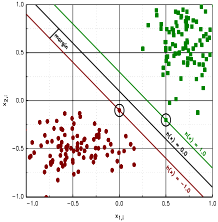
- Universe of n stocks characterized by d economic factors $\mathbf{x} \in \mathbb{R}^d$
- Classify the stocks subjected to their performance indicator $y = \pm 1$
- SVM score is defined as the distance to the frontier

Hard margin principle

- Hyperplane defined by $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = 0$
- Maximize the margin:

$$m_{\mathcal{D}}(h) = \hat{\mathbf{w}}^T \left(\mathbf{x}_+ - \mathbf{x}_-\right)/2 = 1/\|\mathbf{w}\|$$

under constraints: $y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) > 1 \quad i = 1 \ldots n$



Employ SVM score without overfitting

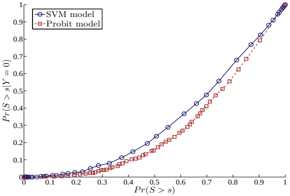
Selection curve

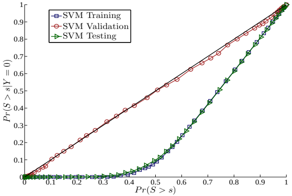
We construct:

- High score:
  $Q(s) = Pr(S \geq s)$
- Selection error:
  $E(s) = Pr(S \geq s \,|\, Y = -1)$

Corss validation

- Training set: Define SVM classifier
- Validation set: Minimize predicting error and SVM error
- SVM score constructed on both Training+Validation





SVM as trend filtering

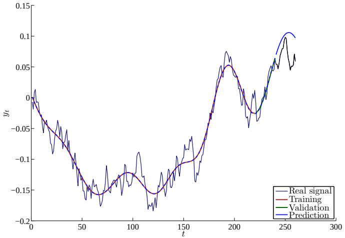
Principle

- Filter y_t by a trend of the form:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

- Minimize the following fitting error:

$$R = \sum_{i=1}^{n} |f(\mathbf{x}_i) - y_i|^2 + n\sigma^2 \|\mathbf{w}\|^2$$



Remarks

- Equivalent to SVM classification.
- Non-linear filtering solved by kernel approach $K = \phi(\mathbf{x})^\top \phi(\mathbf{x})$

Example on S&P 500 index

Cross validation procedure

- Divide data into: training, validation and testing
- Learn on training, optimize parameters on validation, predict on testing