Multivariate Dynamic Kernels for Financial Time Series Forecasting

ICANN16: 25th Int. Conference on Artificial Neural Networks, Barcelona, Sept. 2016

Argimiro Arratia, Lluís Belanche and Mauricio Peña-Grass

Universitat Politècnica de Catalunya - CS

Keep in mind our forecasting challenge

- One output variable
  - Standard & Poor's 500 Index (S&P500)
- Three exogenous variables
  - Volatility index (VIX)
  - U.S. 10-year treasury bond (US10Yr)
  - Cooper 3-month future price (LME3m)

> **Challenge**: **predict** next month performance of S&P500 given its own history and the exogenous variables measured on a daily basis (2006–2014).

- In practice, find a function $X \to Y$ for which $f(\boldsymbol{x})$ is not too different from y on average

Outline

Support Vector Machines (for regression)

An SVM approximates dataset $\mathcal{G} = \{(\boldsymbol{x}_k, y_k) : k = 1, \ldots, N\}$ by multiple regressions of the form:

$$f(\boldsymbol{x}_k, \boldsymbol{w}) = \sum_{i=1}^{D} w_i \phi_i(\boldsymbol{x}_k) + b$$

where $\{\phi_i(\boldsymbol{x}_k)\}_{i=1}^{D}$ are the features of inputs,
$\boldsymbol{w} = \{w_i\}_{i=1}^{D}$ and b are coefficients estimated from data by minimizing the risk functional:

$$R(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(\boldsymbol{x}_i, \boldsymbol{w})|_\epsilon + \frac{1}{2}||\boldsymbol{w}||^2$$

with respect to the ϵ-insensitive loss function

$$|y_i - f(\boldsymbol{x}_i, \boldsymbol{w})|_\epsilon = \begin{cases} 0 & \text{if } |y_i - f(\boldsymbol{x}_i, \boldsymbol{w})| < \epsilon \\ |y_i - f(\boldsymbol{x}_i, \boldsymbol{w})| & \text{otherwise} \end{cases}$$

Interpreting the approximations $f(\boldsymbol{x}_k, \boldsymbol{w}) = \sum_{i=1}^{D} w_i \phi_i(\boldsymbol{x}_k) + b$ as hyperplane in D-dimensional feature space defined by $\{\phi_i(\boldsymbol{x})\}$
The Goal: to find a hyperplane $f(\boldsymbol{x}, \boldsymbol{w})$ that minimizes $R(\boldsymbol{w})$

Vapnik (1995) shows

such minimum is attained by functions of the form:

$$f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\alpha}') = \sum_{i=1}^{N} (\alpha_i' - \alpha_i) K(\boldsymbol{x}, \boldsymbol{x}_i) + b$$

where $K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{D} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{y})$ is the kernel function (and inner product in D-dim. feature space),

Kernel trick

One does not need to compute the features $\phi_i(\boldsymbol{x})$ and their inner product, since kernel can be computed alternatively through analytical functions not involving them.

Common choices for kernel

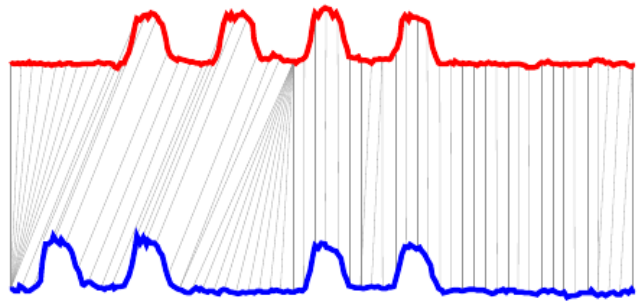
- polynomial kernel (with degree d): $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^d$;
- Gaussian radial basis function (RBF)
  $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-||\boldsymbol{x} - \boldsymbol{y}||^2/\sigma^2)$, with bandwidth σ^2, and
- sigmoid kernel: $K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\kappa \boldsymbol{x} \cdot \boldsymbol{y} - \sigma)$.

Multivariate Dynamic Kernels for Time Series

Measuring similarity between time series

Use an old idea: Dynamic Time Warping (Sakoe & Chiba, 1970)

- Objective: find a good **alignment** between **x** and **y** before computing $d_{\text{Euclidean}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} d(x_i, y_i)$



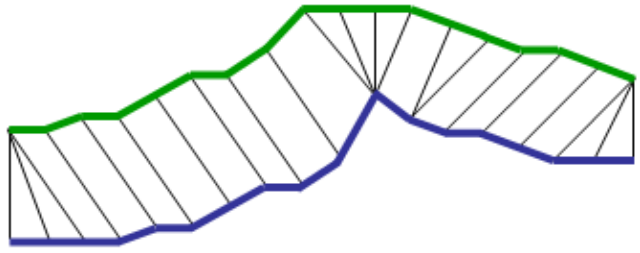
DTW

$$d_{\text{DTW}}(\boldsymbol{x}, \boldsymbol{y}) = \min_{\pi \in \mathbf{A}(\boldsymbol{x}, \boldsymbol{y})} \sum_{i=1}^{|\pi|} d(x_{\pi_1(i)}, y_{\pi_2(i)}) = \min_{\pi \in \mathbf{A}(\boldsymbol{x}, \boldsymbol{y})} D_\pi(\boldsymbol{x}, \boldsymbol{y})$$

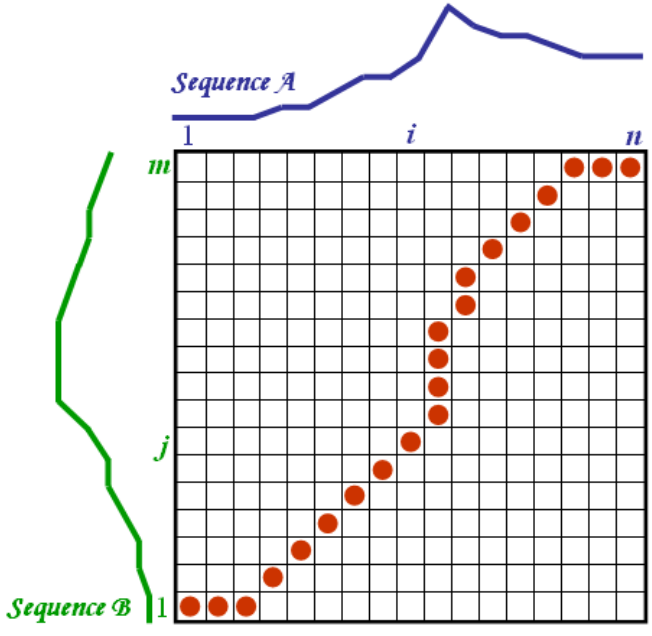
where $d(\boldsymbol{x}, \boldsymbol{y})$ is usually $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$

Alignments

- Here are two sequences aligned



- An alignment is an increasing path on a grid



Problems with the DTW distance

- The DTW is not rigorously a distance and is known not to be n.d. (not satisfy the triangle inequality). Then the similarity

$$k_{\mathrm{DTW}}(\boldsymbol{x}, \boldsymbol{y}) = e^{-d_{\mathrm{DTW}}(\boldsymbol{x}, \boldsymbol{y})}$$

is NOT p.d. in general

- The lack of p.d. contradicts most of the mathematical foundations of kernel methods
- Moreover, DTW is a arbitrary choice in terms of defining the distance based exclusively on the optimal alignment:
$\min_{\pi \in \mathbf{A}(\boldsymbol{x}, \boldsymbol{y})} D_{\pi}(\boldsymbol{x}, \boldsymbol{y})$
- All these weaknesses lead to unexpected and counter-intuitive behaviors in some cases

Global alignment kernel

- Instead of the minimum, GA (Cuturi, Vert, Birkenes and Matsui, 2009) consider the soft-minimum of $D_\pi(\boldsymbol{x}, \boldsymbol{y})$:

$$\underset{\pi \in \mathbf{A}(\boldsymbol{x},\boldsymbol{y})}{\text{soft-minimum}}(D_\pi(\boldsymbol{x}, \boldsymbol{y})) = -\log \sum_{\pi \in \mathbf{A}(\boldsymbol{x},\boldsymbol{y})} e^{-D_\pi(\boldsymbol{x},\boldsymbol{y})}$$

- To get a similarity, we take exp(soft-minimum) :

$$k_{\text{GA}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\pi \in \mathbf{A}(\boldsymbol{x},\boldsymbol{y})} e^{-D_\pi(\boldsymbol{x},\boldsymbol{y})}$$

- The GA kernel takes advantages of all distances spanned by all possible alignments

This is our gold-standard on kernels for TS!

Multivariate dynamic euclidean distance kernel

- Given that financial time series follow a filtration process, we propose the MDED alignment that shortens the longer time series up to become equal in length to the shorter time series
- Thus, the MDED alignment between time series $\boldsymbol{x}$ and $\boldsymbol{y}$ with lengths $N \geq M$ is $\pi_{\text{MDED}} = \{(N-(M-1),1),(N-(M-2),2),\ldots,(N-1,M-1),(N,M)\}$
- We define the MDED as

$$MDED(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{M}\sum_{i=1}^{M} d(x_{\pi_{\text{MDED}_1}(i)}, y_{\pi_{\text{MDED}_2}(i)})$$

- For turning it into similarity, we use the RBF function

$$k_{\text{MDED}} = \exp\left\{\frac{-MDED(\boldsymbol{x},\boldsymbol{y})}{2\sigma}\right\}$$

- We recommend $\sigma = \text{median } MDED(\boldsymbol{x},\boldsymbol{y})$

Vector autoregressive kernel

- We propose a *VAR kernel* (much simpler than (Cuturi 2011)) based on comparing the similarity of VAR models parameters.
- A VAR(L) model is such that $x(t) = \sum_{l=1}^{L} A_l x(t-l) + b + \varepsilon_t$
- We propose to append the $\hat{A}$'s and $\hat{b}$'s into a single matrix denoted as $\hat{B} = (\hat{A}_1|\hat{A}_2|\ldots|\hat{A}_L|[\hat{b}])$. Then, compute a distance using the Frobenious norm

$$FD(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\text{Trace}\left\{(\hat{B}_{\boldsymbol{x}} - \hat{B}_{\boldsymbol{y}})(\hat{B}_{\boldsymbol{x}} - \hat{B}_{\boldsymbol{y}})^T\right\}}$$

- Since we need a similarity, we use a radial basis function

$$k_{\text{VAR}} = \exp\left\{\frac{-FD(\boldsymbol{x}, \boldsymbol{y})}{2\sigma}\right\}$$

- We recommend $\sigma = \text{median } FD(\boldsymbol{x}, \boldsymbol{y})$

Cuturi's VAR-kernel

Cuturi 2011 rely on VAR model for multivariate processes but avoid the two-step approach by using a matrix normal-inverse Wishart prior. In short, a VAR(L) model on time series $X \in \mathbb{R}^{P \times T_X}$ is summarise into two matrices $Z_X \in \mathbb{R}^{P \times (T_X - L)}$ and $W_X \in \mathbb{R}^{(PL+1) \times (T_X - L)}$

Then a VAR-related covariance kernel is formulated using the Z and W matrices, by applying a bayesian linear regression framework with non-informative prior. The kernel function takes the form of

$$k(X, Y) = (|W^T W \triangle + I_C|^{1-\alpha} + |W^T W \triangle + Z^T Z \triangle + I_c|^{\alpha})^{-P/2}$$

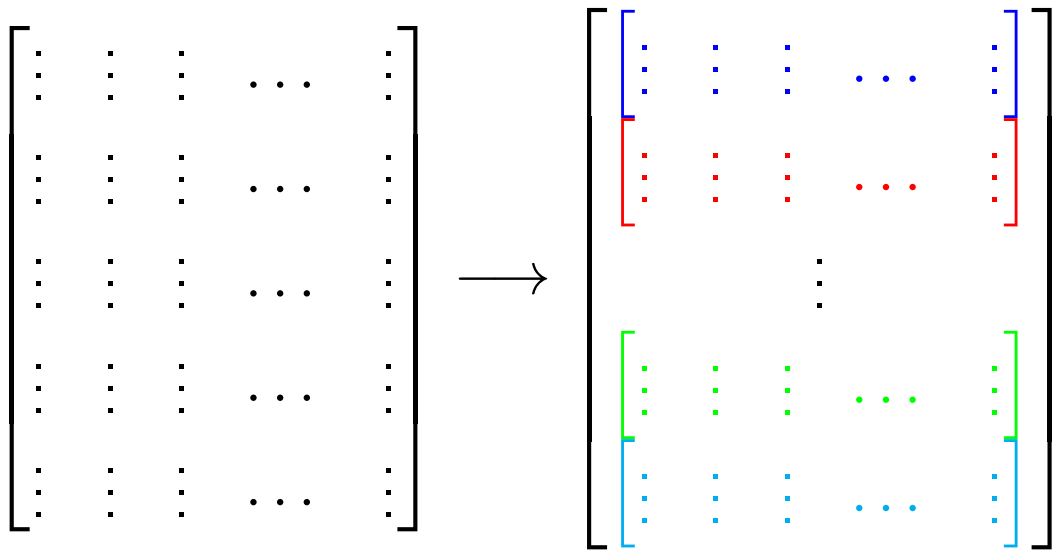
where $X = [W_X W_Y]$, $Z = [Z_X Z_Y]$, $c = T_X + T_Y - 1$, and α depend on P and the degrees of freedom d of the matrix-normal inverse. Wishart prior can be a tuneable parameter. This is the Gram formulation

This one-step VAR kernel has many tuneable parameters, including the lag parameter L, making it difficult for the kernel to perform well in practice without much domain knowledge by the user.

Temporal Data Blocks

Data compression process

- We design a data compression process that redefines the original dataset into temporal data blocks so as to analyze temporal information within each block



- This approach allows to extract additional information and not only a single vector of prices for each interval

Temporal data block

- A data block is the central unit of our analysis, i.e., a multivariate time series (MVT) of P time series sampled at the same time intervals

$$\boldsymbol{x}_j = \left\{ \begin{bmatrix} x_1(1) \\ x_2(1) \\ \vdots \\ x_P(1) \end{bmatrix} \cdots \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_P(t) \end{bmatrix} \cdots \begin{bmatrix} x_1(T_j) \\ x_2(T_j) \\ \vdots \\ x_P(T_j) \end{bmatrix} \right\}$$

- A MVT can be represented as a P-by-T_j matrix: $\boldsymbol{x}_j \in \mathbb{R}^{P \times T_j}$
- Therefore, the compressed database of MVTs is

$$\mathcal{D} = \left\{ \boldsymbol{x}_j : \boldsymbol{x}_j \in \mathbb{R}^{P \times T_j} \right\}_{j=1}^{N}$$

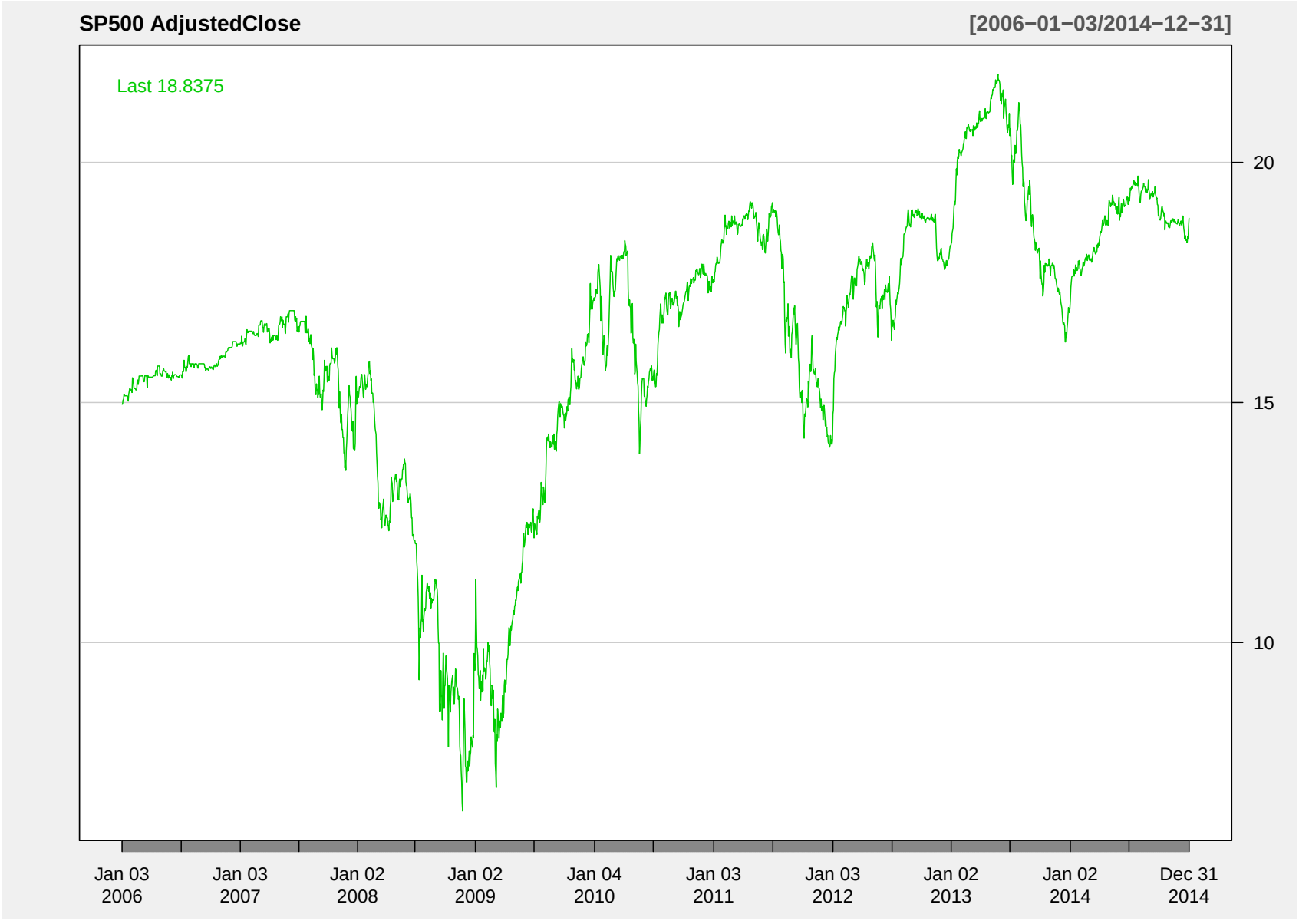
Forecasting Performance Evaluation

Mixed-frequency forecasting challenge

- One output variable
  - Standard & Poor's 500 Index (S&P500)
- Three exogenous variables
  - Volatility index (VIX)
  - U.S. 10-year treasury bond (US10Yr)
  - Cooper 3-month future price (LME3m)

Challenge: **predict** next month performance of S&P500 given its own history and the exogenous variables measured on a daily basis (2006–2014).

S&P 500 daily Adjusted price, 2006-2014



Data pre-processing

- $y = R_{t+1}$ (next month log return)
- Input features are constructed on a daily basis to capture temporal patterns of different scale on S&P500, VIX, US10yr and LME3m using the pre-processing function

$$ROC_{t,n} = \ln\left(\frac{x_t}{x_{t-n}}\right)$$

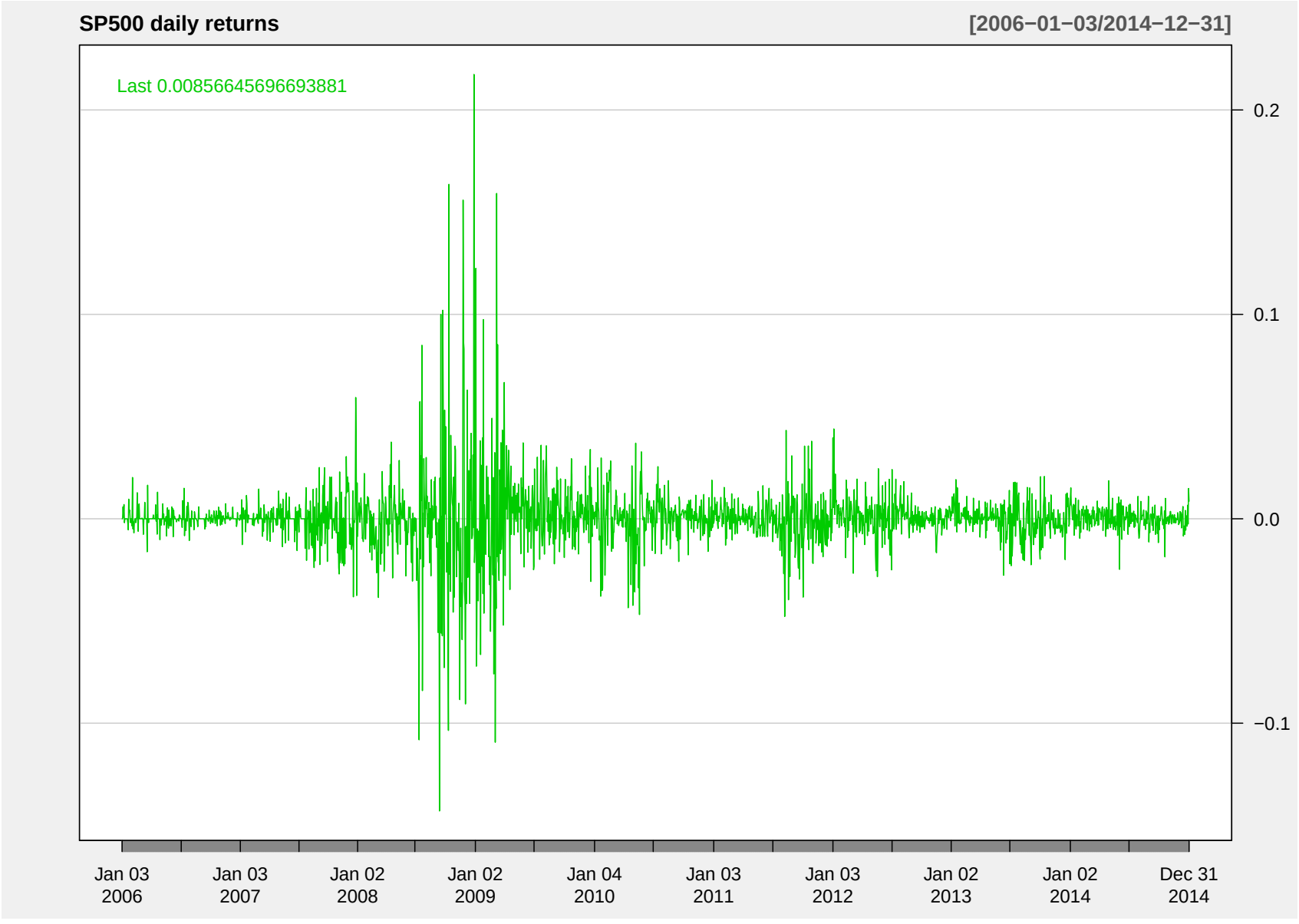
- For each ith time series ($i = 1, \ldots, 4$), we derive a vector of several rates of changes on day t

$$x_t^i = [ROC_{t,20}^i, ROC_{t,40}^i, ROC_{t,60}^i, ROC_{t,100}^i, ROC_{t,140}^i]$$

- Then, the input features for day t takes the form of

$$x_t = [x_t^1, x_t^2, x_t^3, x_t^4]$$

S&P 500 daily returns, 2006-2014



Performance metrics

- We use the mean absolute scaled error (MASE) which scales the forecasted error e_t using a naive forecast

$$\text{MASE} = \text{mean}\left|\frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|}\right|$$

- We also include the accuracy or hit rate (HITS) $\rightarrow$ proportion of correctly predicted trends in the period

$$\text{HITS} = \frac{|\{F_i \mid (Y_i - Y_{i-1}) \cdot (F_i - F_{i-1}) > 0,\ i = 1, \ldots, n\}|}{n}$$

Empirical results

Table 1: Mean Absolute Scaled Error of Multivariate Dynamic Kernels

	MASE				
	Naive	k_{GA}	k_{VAR}	k_{MDED}	VAR
2006 – 2008	1.0000	0.7835	0.7954	0.7688	1.1507
2009 – 2011	1.0000	0.8959	0.8496	0.8464	1.1305
2012 – 2014	1.0000	0.7284	0.7123	0.7332	1.5677
Total	1.0000	0.8195	0.7981	0.7940	1.2455

Empirical results

Table 2: Forecasting Accuracy of Multivariate Dynamic Kernels

	HITS				
	Naive	k_{GA}	k_{VAR}	k_{MDED}	VAR
2006 – 2008	0.6571	0.7778	0.7500	0.7222	0.6389
2009 – 2011	0.500	0.5833	0.5278	0.5556	0.6111
2012 – 2014	0.5833	0.7222	0.7222	0.7222	0.4167
Total	0.5794	0.6944	0.6667	0.6667	0.5556

Industrial Experimental Design:
Trading Perspective

SVR timing model

- We design a timing rotation investment strategy in which a positive prediction $\hat{f}_{t+1}$ will result in going long S&P500 while a negative prediction in going short
- The log-return, $\hat{R}_{t+1}$, of our strategy is computed as

$$\hat{R}_{t+1} = \begin{cases} |R_{t+1}| & \text{if } R_{t+1} \cdot \hat{f}_{t+1} \geq 0; \\ -|R_{t+1}| & \text{otherwise.} \end{cases}$$

- We include different levels of transaction costs

Objective: **evaluate** the performance of the investment strategy based on SVR with multivariate dynamic kernels as compared to the buy-and-hold strategy.

Experimental results timing model

Table 3: Summary performance statistics for SVR timing rotation strategies with transactions costs.

	Period: 2006/01 – 2014/12			
	Benchmark	Model, 0 bp.	Model, 30 bp.	Model, 50 bp.
k_{GA}				
Total cumulative (%)	50.04	145.64	133.62	125.59
Mean (%)	5.56	16.18	14.85	13.96
Standard deviation (%)	15.54	14.90	14.94	14.97
Sharpe ratio	0.36	1.09	0.99	0.93
k_{VAR}				
Total cumulative (%)	50.04	123.78	113.86	107.24
Mean (%)	5.56	13.75	12.65	11.92
Standard deviation (%)	15.54	15.11	15.16	15.21
Sharpe ratio	0.36	0.91	0.83	0.78
k_{MDED}				
Total cumulative (%)	50.04	131.99	122.08	115.45
Mean (%)	5.56	14.67	13.56	12.83
Standard deviation (%)	15.54	15.03	15.10	15.15
Sharpe ratio	0.36	0.98	0.90	0.85
VAR				
Total cumulative (%)	50.04	67.19	45.86	31.60
Mean (%)	5.56	7.47	5.10	3.51
Standard deviation (%)	15.54	15.47	15.57	15.65
Sharpe ratio	0.36	0.48	0.33	0.22

Experimental results timing model

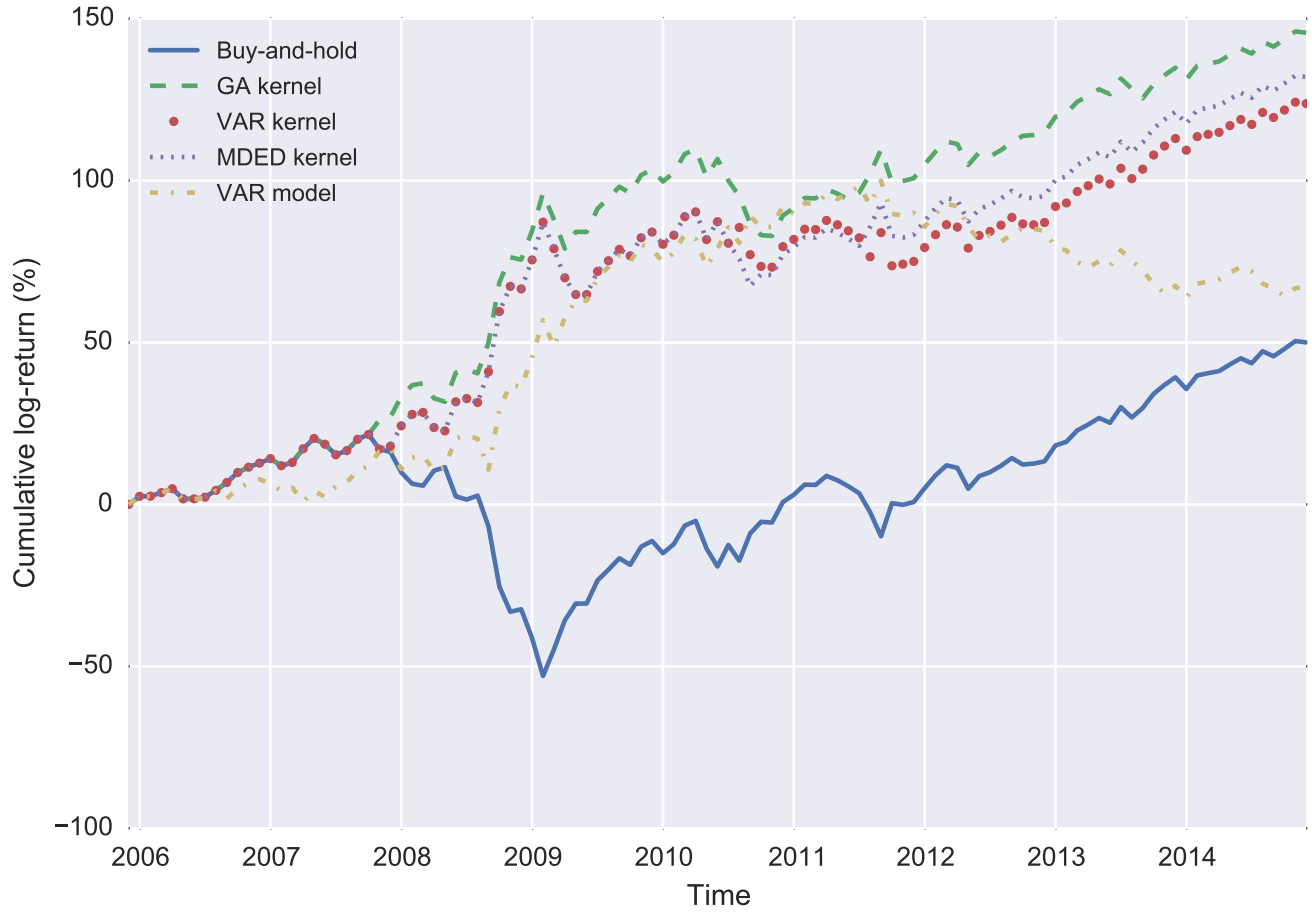


Figure 1: Cumulative log-returns from the SVR timing strategy with multivariate dynamic kernels for the period 2006 – 2014, with no transaction costs.

Experimental results timing model

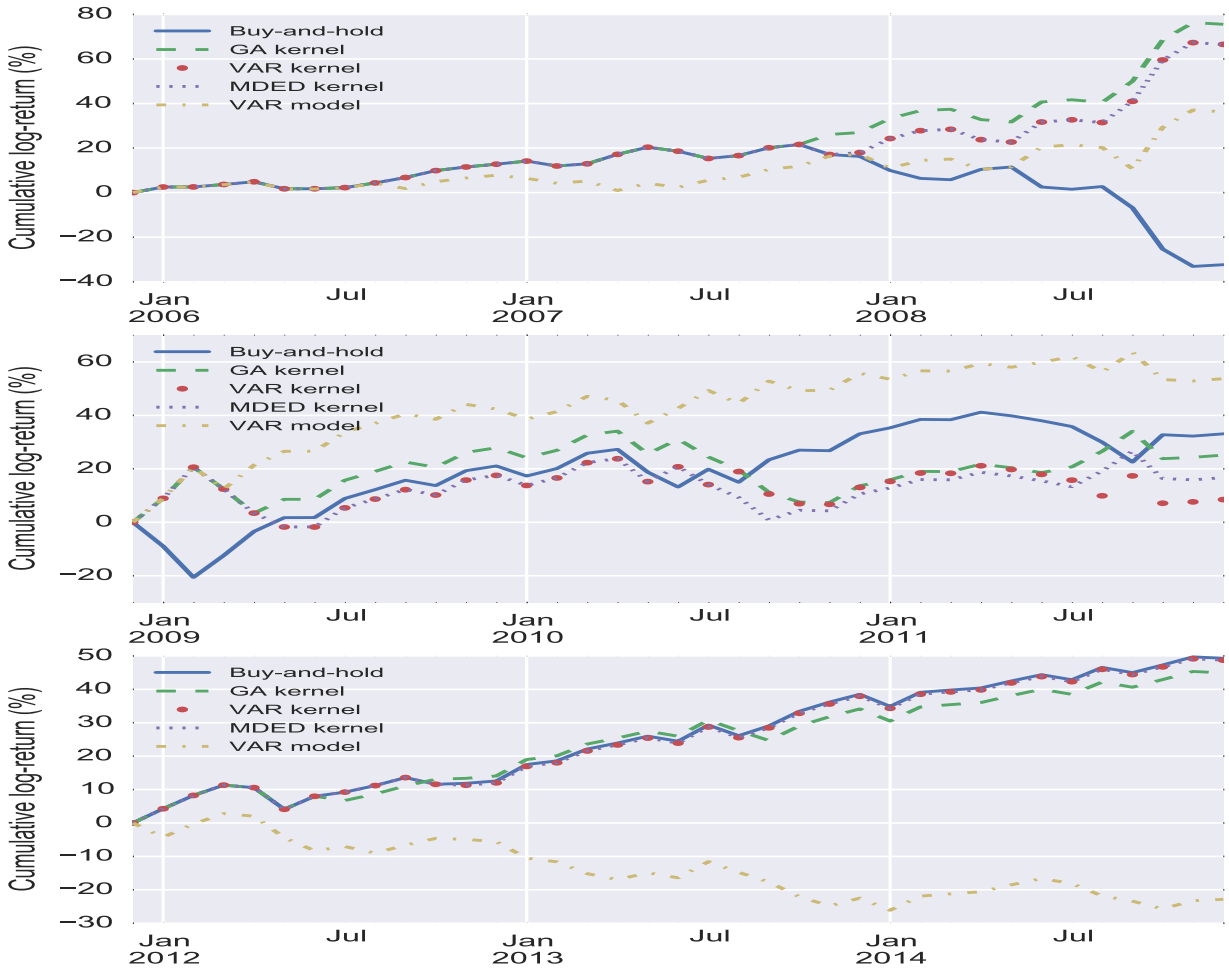


Figure 2: Cumulative returns from SVR timing strategy with multivariate dynamic kernels along different periods (no transaction costs)

Conclusions

Conclusions

- The proposed forecasting method facilitates the integration of data measured at different frequencies and at irregular time intervals in financial markets.
- The method increases the predictions accuracy and performs better than the naive forecast.
- Our VAR and MDED kernels prove to be highly competitive with the gold-standard GA similarity in literature and provides significant gains in computational speed.

Why this (easy) approach works

- Aggregated normality phenomena: the wider the period of observations the closer to normal density.
  Our compressed data blocks amount to observing data at wide periods but no intermediate information is lost.
- The best forecaster for *Gaussian* variable Y_t w.r.to MSE and only knowledge of its past is linear regression.
  For multivariate (Gaussian) time series the VAR gives a good fit.

An apologia for simple models

From David Hand, *Classifier Technology and the Illusion of Progress*, Statistical Science 2006.

> "if insufficient information is known about likely sources of variability in the data, then the principle of parsimony suggests that it is better to stick to simple models".

Multivariate Dynamic Kernels for Financial Time Series Forecasting

ICANN16: 25th Int. Conference on Artificial Neural Networks, Barcelona, Sept. 2016

Argimiro Arratia, Lluís Belanche and Mauricio Peña-Grass

Universitat Politècnica de Catalunya - CS