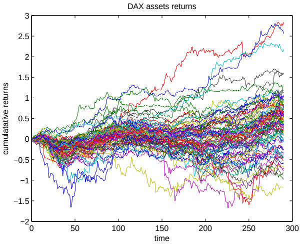
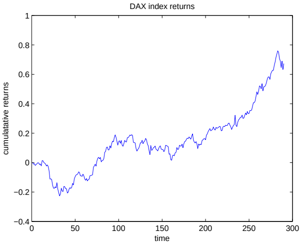
Dynamic Asset Allocation for Bivariate Enhanced Index Tracking using Sparse PLS

Brian McWilliams   Giovanni Montana
Department of Mathematics
Imperial College London

21 July 2009

Index Tracking





- The index, Y, is a weighted sum of its constituent assets, X
- Aim to minimize the variance of the error between the index returns and our portfolio returns
- Find the most predictive subset of X and assign weights, β
- The most predictive subset of X and the weights will change over time
- Ignore costs and constraints for now

The Problem

- The data:
  - Input streams (asset returns): $X \in \mathbb{R}^{n \times p}$
  - Response streams (index returns): $Y \in \mathbb{R}^{n \times q}$

  observations arrive one at a time, sequentially.
- $Y = X\beta + E$
- p is often large with many correlated variables so we want to reduce dimensionality of X to $R < p$
- Objective: Predict Y using a subset of X so that tracking error is minimized
- Do this efficiently *on-line*

Singular Value Decomposition (SVD)

- Decompose X into three matrices:

$$X = UDV^{\mathsf{T}}$$

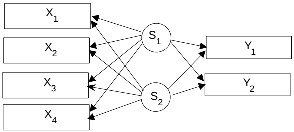
- $U = [u^{(1)}, .., u^{(n)}] \in \mathbb{R}^{n \times n}$ and $V = [v^{(1)}, ..., v^{(p)}] \in \mathbb{R}^{p \times p}$ are orthonormal and eigenvectors of XX^{T} and $X^{\mathsf{T}}X$ respectively
- $D \in \mathbb{R}^{n \times p}$ is diagonal and the square root eigenvalues of $X^{\mathsf{T}}X$
- Low rank approximation property: $\sum_{i=1}^{r} u^{(i)} d^{(i)} v^{(i)}$ is the best rank-r approximation of X

Principal Components Analysis (PCA)

- Find orthogonal basis vectors which maximize covariance of data.
- Compute eigenvectors of $C = X^{\mathsf{T}}X$ (i.e. singular vectors of X).
- Project X into R-dimensional space spanned by $[v^{(1)}, ..., v^{(R)}]$.
- Use in index tracking: First PC captures the *market factor* Alexander and Dimitriu (2005) .
- But - PCA assumes large variances are more important. Doesn't take into account the response.

Partial Least Squares Regression (PLS)

- Dimensionality reduction and regression
- Assume data and response depend on a small number of latent factors, S



$$X = \sum_{r=1}^{R} s^{(r)} {b^{(r)}}^{\mathsf{T}} + E, \qquad Y = \sum_{r=1}^{R} s^{(r)} {w^{(r)}}^{\mathsf{T}} + F$$

- Y=XVW+G
- s=Xv - Find weights, v s.t.

$$\max_{\|v\|=1} [\mathrm{cov}(Xv, Y)]^2$$

PLS

- $v^{(1)}$ is the largest eigenvector of $X^\mathsf{T} Y Y^\mathsf{T} X$
- Many algorithms/variations exist. Mainly in Chemometrics literature.
- Limited by rank(Y). Must perform *R separate SVD computations* to obtain $[v^{(1)}, \ldots, v^{(R)}]$
- Solution: add ridge term to make H full rank:

$$H = \alpha X^\mathsf{T} X + (1-\alpha) X^\mathsf{T} Y Y^\mathsf{T} X \quad (\alpha \sim 10^{-5})$$

- We can now extract up to rank(X) latent factors with one SVD Gidskehaug et al. (2004)

LASSO (Tibshirani, 1996)

- Penalize L_1 norm of regression coefficients

$$\hat{\beta} = \min_{\beta} \|Y - X\beta\|^2 + \gamma \sum_{i=1}^{p} |\beta_i|$$

- Large enough γ results in a sparse solution
- Numerous efficient algorithms: LARS, Coordinate descent
- $\beta_i^{lasso} = \text{sign}(\beta_i)(|\beta_i| - \gamma)_+$ - Soft threshold
- Sparse portfolios Brodie et al. (2008)

Regularized SVD

- Reformulate PCA eigenvector problem as regression Shen and Huang (2008), Witten et al. (2009)

$$\min_{\tilde{u},\tilde{v}} \|X - \tilde{u}\tilde{v}\|_F^2$$

- Solve using SVD of X, $\tilde{u} = u^{(1)}d^{(1)}$, $\tilde{v} = v^{(1)}$
- Now apply penalty to $\tilde{v}$

$$\min_{\tilde{u},\tilde{v}} \|X - \tilde{u}\tilde{v}\|_F^2 + p_\gamma(\tilde{v})$$

- Our choice of penalty is $p_\gamma(\tilde{v}) = \gamma \sum_{i=1}^{p} |\tilde{v}_i|$ but could use a different penalty
- Solve the LASSO by iteratively applying $v_i^{lasso} = \text{sign}(\tilde{v}_i)(|\tilde{v}_i| - \gamma)_+$

Sparse PLS

- We can apply this method to obtain Sparse PLS

$$\min_{\tilde{u},\tilde{v}} \|H - \tilde{u}\tilde{v}\|_F^2 + \gamma \sum_{i=1}^{p} \|\tilde{v}_i\|$$

- Solve using SVD of H, $\tilde{u} = u^{(1)}d^{(1)}$, $\tilde{v} = v^{(1)}$ and apply soft threshold iteratively as before
- Sparse PLS weight vectors lead to sparse PLS regression coefficients

Solving PLS online

- Solving PCA online is easy: RLS techniques (Yang, 1995), SVD updating (Levy and Lindenbaum, 2000)
- For PLS we need to find and incrementally update eigenvectors of

$$H = \alpha X^\mathsf{T} X + (1 - \alpha) X^\mathsf{T} Y Y^\mathsf{T} X$$

- Can't use RLS or most common SVD updating algorithms
- Our solution - use Adaptive SIM: Adaptive generalization of power method for finding eigenvectors (Erlich and Yao, 1994)

Incremental Sparse PLS (iS-PLS)

1. For each data pair $(x_t \ , \ y_t)$
   - update C_t and M_t

   $$C_t = \lambda_t C_{t-1} + x_t^{\mathsf{T}} x_t \quad M_t = \lambda_t M_t + x_t^{\mathsf{T}} y_t$$

   $0 \leq \lambda_t \leq 1$ is an *adaptive* forgetting factor
   - update H_t

   $$H_t = \alpha C_t + (1 - \alpha) M_t M_t^{\mathsf{T}}$$

2. Update eigenvectors of H_t by performing one SIM iteration
3. Soft threshold the updated eigenvectors
4. Recompute sparse PLS parameters (latent factors, y-loading vectors, regression coefficients)

Adaptive Forgetting

- Consider residual errors:
  - *a priori* error: $e_t = y_t - x_t\beta_{t-1}$
  - *a posteriori* error: $\epsilon_t = y_t - x_t\beta_t$
- Update forgetting factor based on difference between the error variances (Paleologu et al., 2008) :

$$\lambda_t = \frac{\sigma_q \sigma_\epsilon}{\sigma_e - \sigma_\epsilon}$$

$q_t = x_t C_t^{-1} x_t^{\mathsf{T}}$

- Slowly changing data - Small difference between prior and posterior error - $\lambda_t \approx 1$
- Quickly changing data - Large difference between prior and posterior error - $\lambda_t << 1$

Simulation Results

- Generate 3 factors as AR(1) processes:

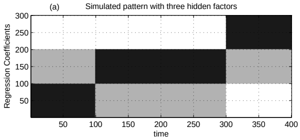
$$F_{t,j} = \delta_j F_{t-1,j} + \epsilon_{t,j} \qquad \text{for } t = 2, \ldots, 400$$

  - $\delta_j = [0.1, 0.4, 0.2]$, $\epsilon_j \sim \mathcal{N}(\mu_j, 3.5^2)$

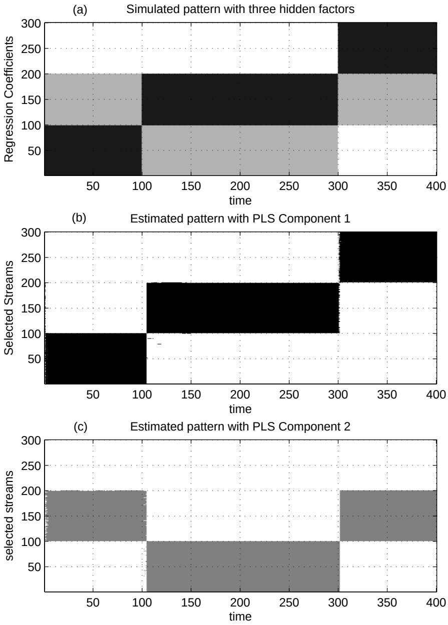
- Each input variable is generated as

$$x_{t,i} = F_{t,j} + \eta_t \qquad \eta_t \sim \mathcal{N}(0, 1)$$

  - Each variable only depends on one factor

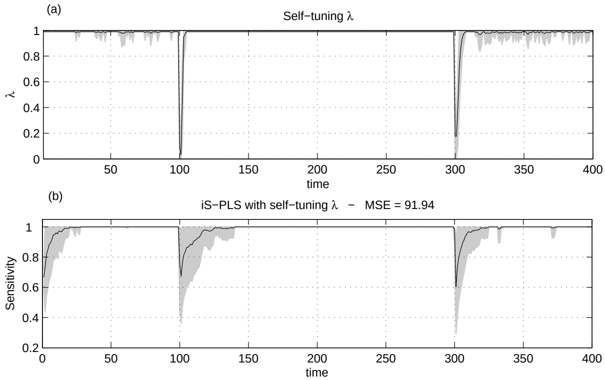


Simulation Results



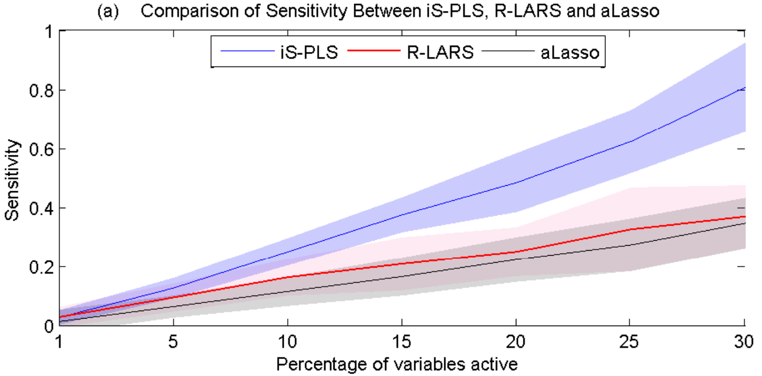
Simulation Results

- Sensitivity averaged over 500 Monte Carlo simulations



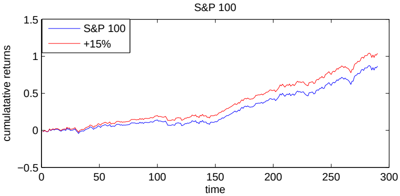
- Forgetting factor changes in response to the active latent factors changing.
- After a change, solution quickly converges to maximum sensitivity.

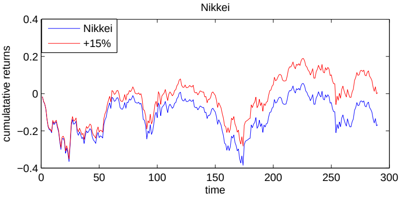
Simulation Results



- Mean sensitivity of the iS-PLS algorithm as a function of number of active variables compared to Recursive LARS Kim et al. (2004) and Adaptive Lasso Anagnostopoulos et al. (2008).

Application to Bivariate Index Tracking

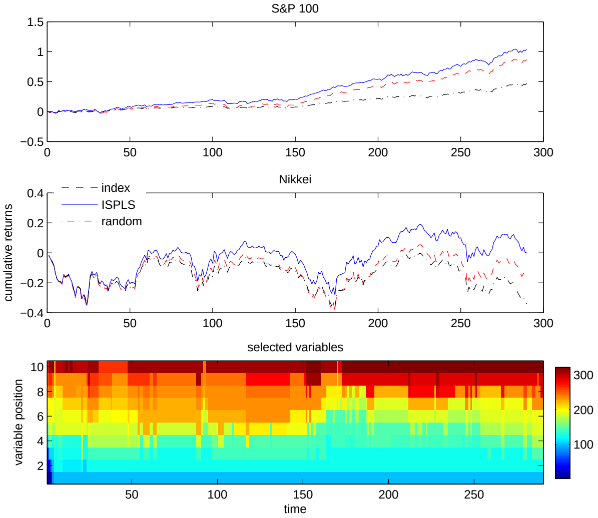




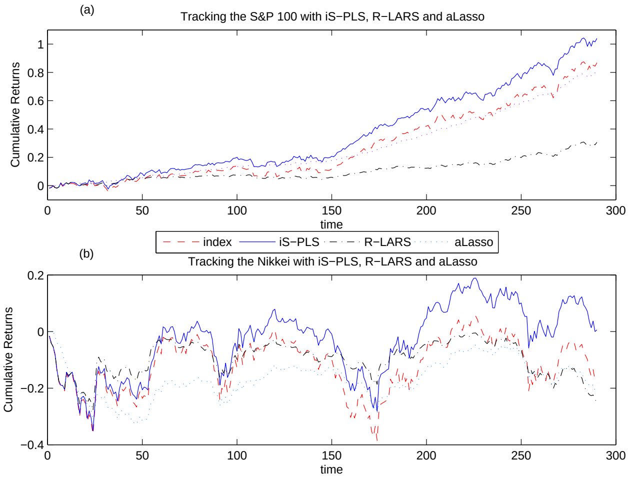
- Try to overperform S&P 100 returns AND Nikkei returns by 15% annually
- 323 available assets - select only 10
- Use only the first PLS component
- Compare with averaged random portfolio updated using RLS

Application to Bivariate Index Tracking

- iS-PLS portfolio overperforms the index returns by ∼ 15%
- Random portfolio underperforms the index returns
- Bottom plot shows the selected variables over time



Application to Bivariate Index Tracking



- iS-PLS compared with Recursive LARS and Adaptive Lasso in the same bivariate enhanced index tracking problem.

Summary

- Very few approaches to on-line variable selection exist:
  - Adaptive Lasso Anagnostopoulos et al. (2008)
  - Recursive LARS Kim et al. (2004)

  Neither takes into account latent factors
- iS-PLS does fast on-line dimensionality reduction and variable selection
- Performs well when data and response depend on latent factors
- But...

Limitations and Future Work

- Limitations
  - At present we are specifying the number of PLS components and variables
  - Index tracking does not take into account constraints or transaction costs
- Future Work
  - Automatic model selection - online cross validation Vijayakumar et al. (2005)
  - Realistic index tracking - nonnegativity and inequality constraints?
  - Minimize transaction costs - coefficients smooth over time?

References

Alexander, C. and Dimitriu, A. (2005). Sources of over-performance in equity markets: mean reversion, common trends and herding. Technical report, ISMA Center, University of Reading, UK.

Anagnostopoulos, C., Tasoulis, D., Hand, D. J., and Adams, N. M. (2008). Online optimisation for variable selection on data streams. *Proceedings of the 18th European Conf. on Artificial Intelligence*, pages 132–136.

Brodie, J., Daubechies, I., Mol, C. D., Giannone, C., and Loris, I. (2008). Sparse and stable Markowitz portfolios. *European Central Bank Working Paper Series*, 936.

Erlich, S. and Yao, K. (1994). Convergences of adaptive block simultaneous iteration method for eigenstructure decomposition. *Signal Processing*, 37:1–13.

Gidskehaug, L., Stdkilde-Jrgensen, H., Martens, M., and Martens, H. (2004). Bridge-PLS regression: two-block bilinear regression without deflation. *Journal of Chemometrics*, 18:208–215.

Kim, S.-P., Rao, Y. N., Erdogmus, D., and Principe, J. C. (2004). Tracking of multivariate time-variant systems based on on-line variable selection. *2004 IEEE Workshop on Machine Learning for Signal Processing*, pages 123–132.

Levy, A. and Lindenbaum, M. (2000). Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9.

Paleologu, C., Benesty, J., and Ciochina, S. (2008). A robust variable forgetting factor recursive least-squares algorithm for system identification. *IEEE Signal Processing Letters*, 15:597–600.

Shen, H. and Huang, J. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99:1015–1034.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288.

Vijayakumar, S., D'Souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

Yang, B. (1995). Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43:95–107.