Choleski Stochastic Volatility
R/Finance 2012

Robert McCulloch

Booth School of Business
University of Chicago

May 11, 2012

Joint with
Hedibert Lopes, Ruey Tsay, Carlos Carvalho, Hugh Chipman, Ed George, Matt Pratola, Dave Higdon, Jim Gattiker ....

Outline:

- Go over some of my recent finance orientated research (all Bayesian).
- Some comments on how I am computing stuff (mostly C++, need more R!)

Quick Review, State Space Models:

State space models provide a very general way to think about time series modeling.

There is an underlying state of the system which evolves over time. The evolution of the state is captured by the *state equation*:

$$p(\theta_t \mid \theta_{t-1})$$

where θ_t is the state at time t.

At each time, we get to observe X_t which depends on θ_t, we have the *observation equation*:

$$p(x_t \mid \theta_t).$$

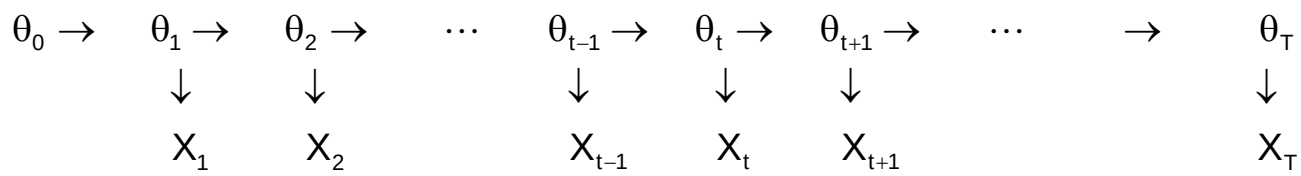
To complete the model we just need a prior on the initial state

$$p(\theta_0).$$

Our joint distribtution is then:

$$p(\theta_0, \theta_1, \ldots, \theta_T, x_1, x_2, \ldots, x_T) = p(\theta_0) \prod p(\theta_t \mid \theta_{t-1})\, p(x_t \mid \theta_t).$$

The general picture:



Each X is a "peek" at the corresponding θ.

If you margin out the θ's get a model in which future X's depend on past X's.

To infer a state we compute:

$$p(\theta_t \mid x_1, \ldots, x_T).$$

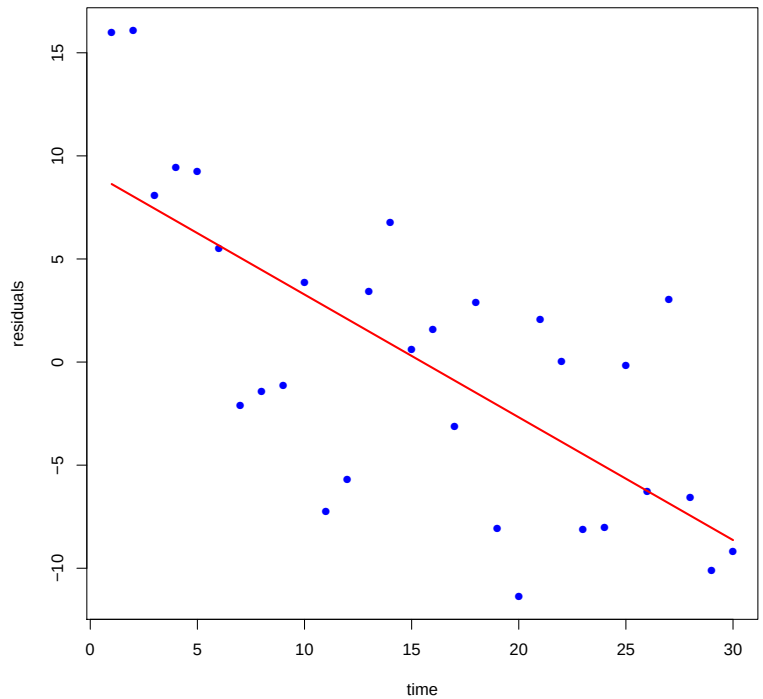
To predict we compute:

$$p(x_{T+1} \mid x_1, \ldots, x_T).$$

"compute" will mean MCMC draw.

Example:

The blue points are a time series in an application I worked on.



Rather than just fit a "trend", I might want to be more flexible.

State space model:

Observation equation:

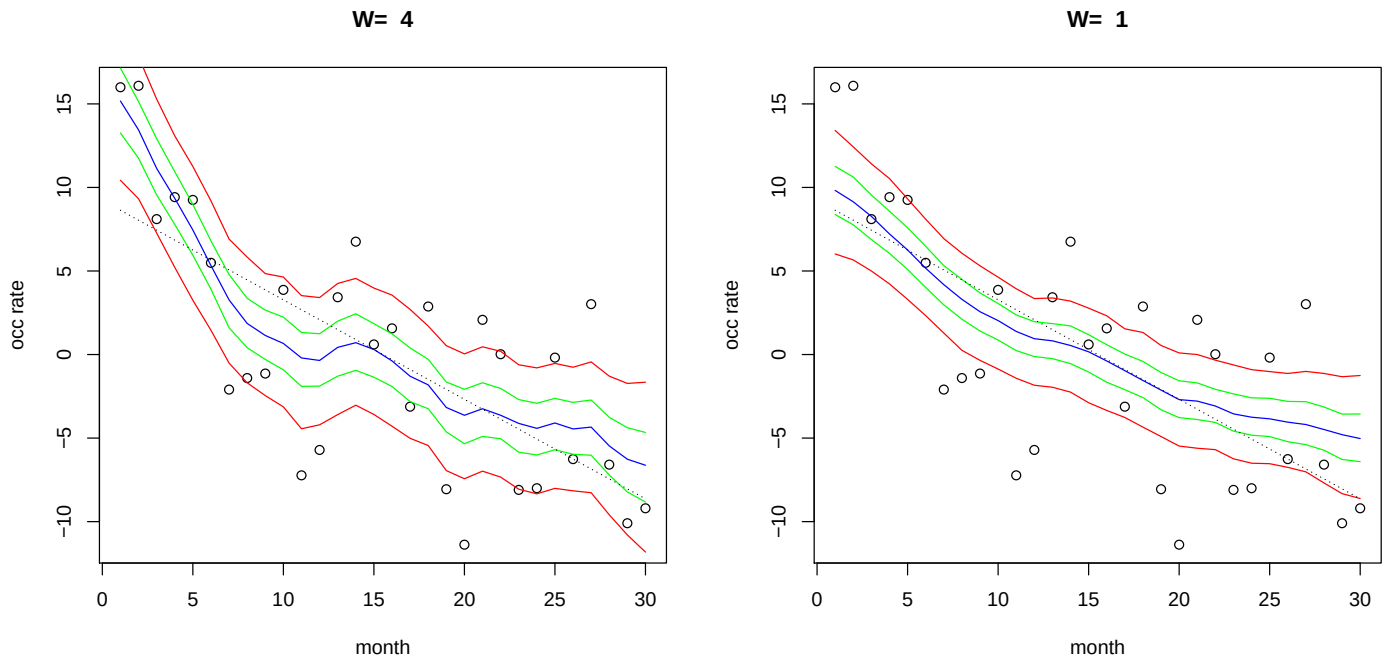
$$X_t = \theta_t + V_t, \quad V_t \sim N(0, V^2).$$

State equation:

$$\theta_t = \theta_{t-1} + W_t, \quad W_t \sim N(0, W^2).$$

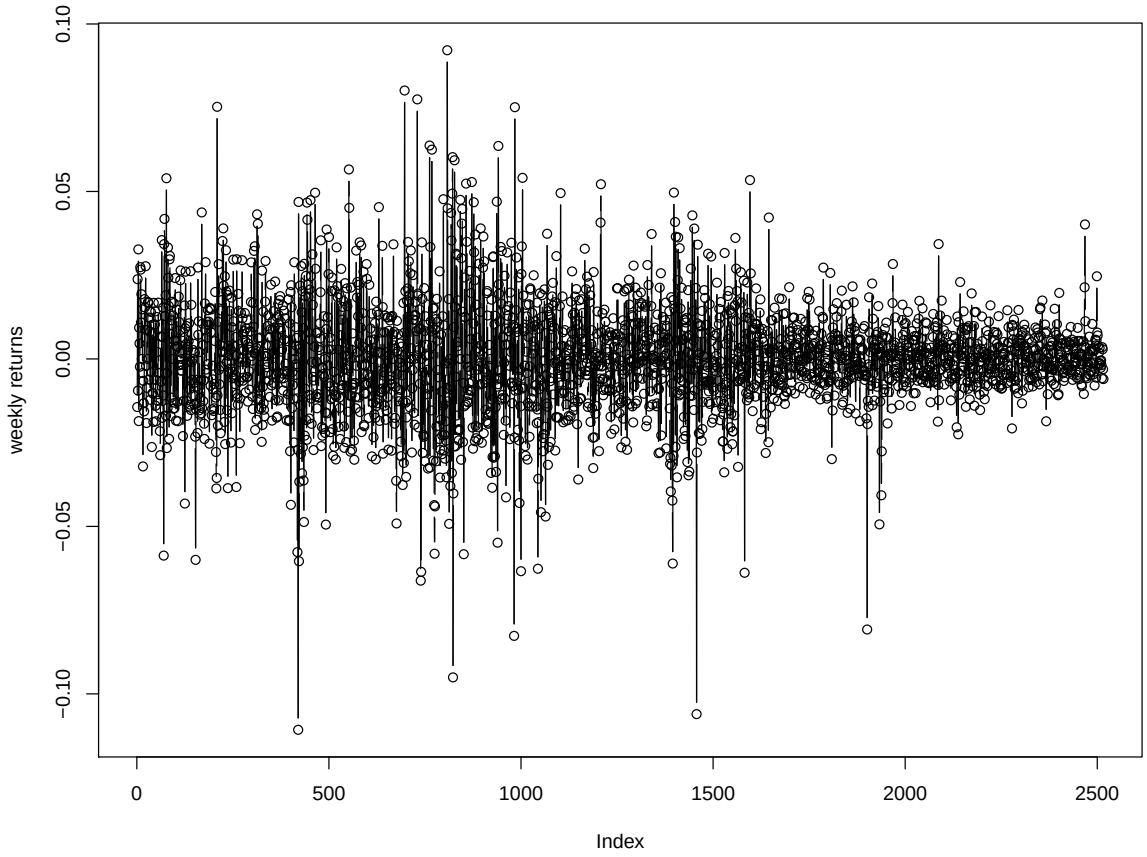
Notice the importance of W !!!!

blue: median of $\{\theta_t\}$ draws.
green: 25% and 75% quantiles of $\{\theta_t\}$ draws.
red: 5% and 95% quantiles of $\{\theta_t\}$ draws.



Example: Univariate Stochastic Volatility

Daily returns on a stock (Coke) in the S&P500.
$T = 2,516$.



We want to model how the variability (volatility) evolves over time.

Y_t is return at time t.

$Y_t \sim N(0, \sigma_t^2)$.

$\sigma_t = e^{(d_t/2)}, \quad d_t = log(\sigma_t^2)$.

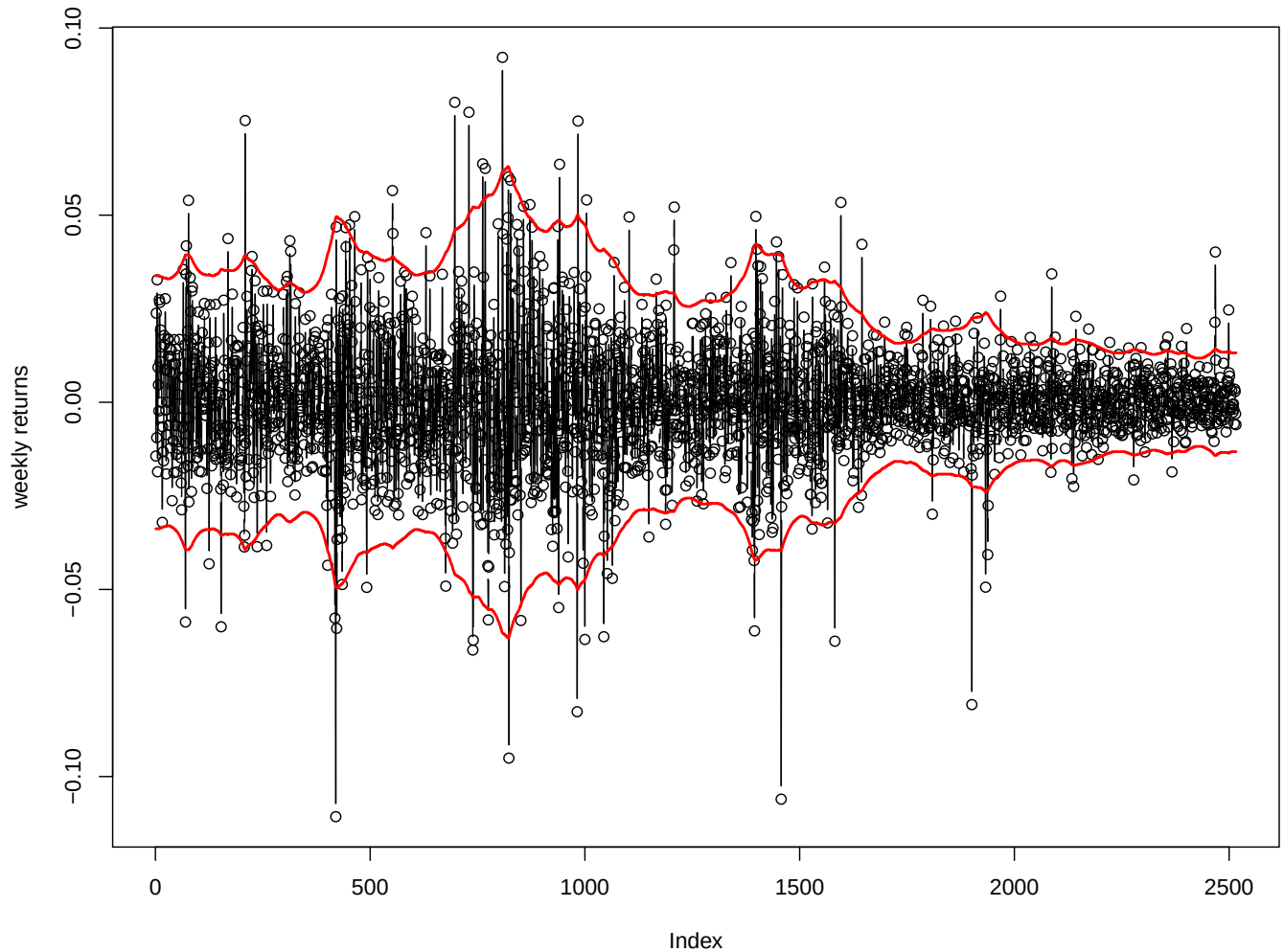
$$\begin{aligned} Y_t &= e^{(d_t/2)} Z_t \\ d_t &= \alpha + \beta d_{t-1} + \tau \epsilon_t \end{aligned}$$

$Z_t, \epsilon_t \sim N(0,1)$, iid.

Observe $\{Y_t\}$, $\{d_t\}$ are the latent states.

This is a non-linear state-space model.

Red is at $\pm 2\,\hat{\sigma}_t$,
$\hat{\sigma}_t$ is posterior mean
(average of draws after burn-in).



MCMC:

Let d_0 be the initial state, $d_0 \sim N(m_0, C_0)$.

$$\begin{array}{rcl} (d_0, \{d_t\}) & | & (\alpha, \beta, \tau), \{Y_t\} \\ (\alpha, \beta, \tau) & | & (d_0, \{d_t\}) \end{array}$$

For every draw of the $\{d_t\}$, compute the $\{\sigma_t\}$, then average to get the posterior mean.

References in paper on my website.
(Chib, Shephard, Carter & Kohn, Fruwirth-Schnatter).

In particular we use the wonderful FFBS (forward-filter, backward-sample) algorithm.

Now let Y_t denote a vector of time-series.

Maybe *dependence structure and volatility* change over time.

We want to do *Multivariate Stochastic Volatility*.

$$Y_t \sim N_p(0, \Sigma_t).$$

Problems:

- Σ_t is a positive definite matrix.
- if p is large, $p(p+1)/2$ is very large.

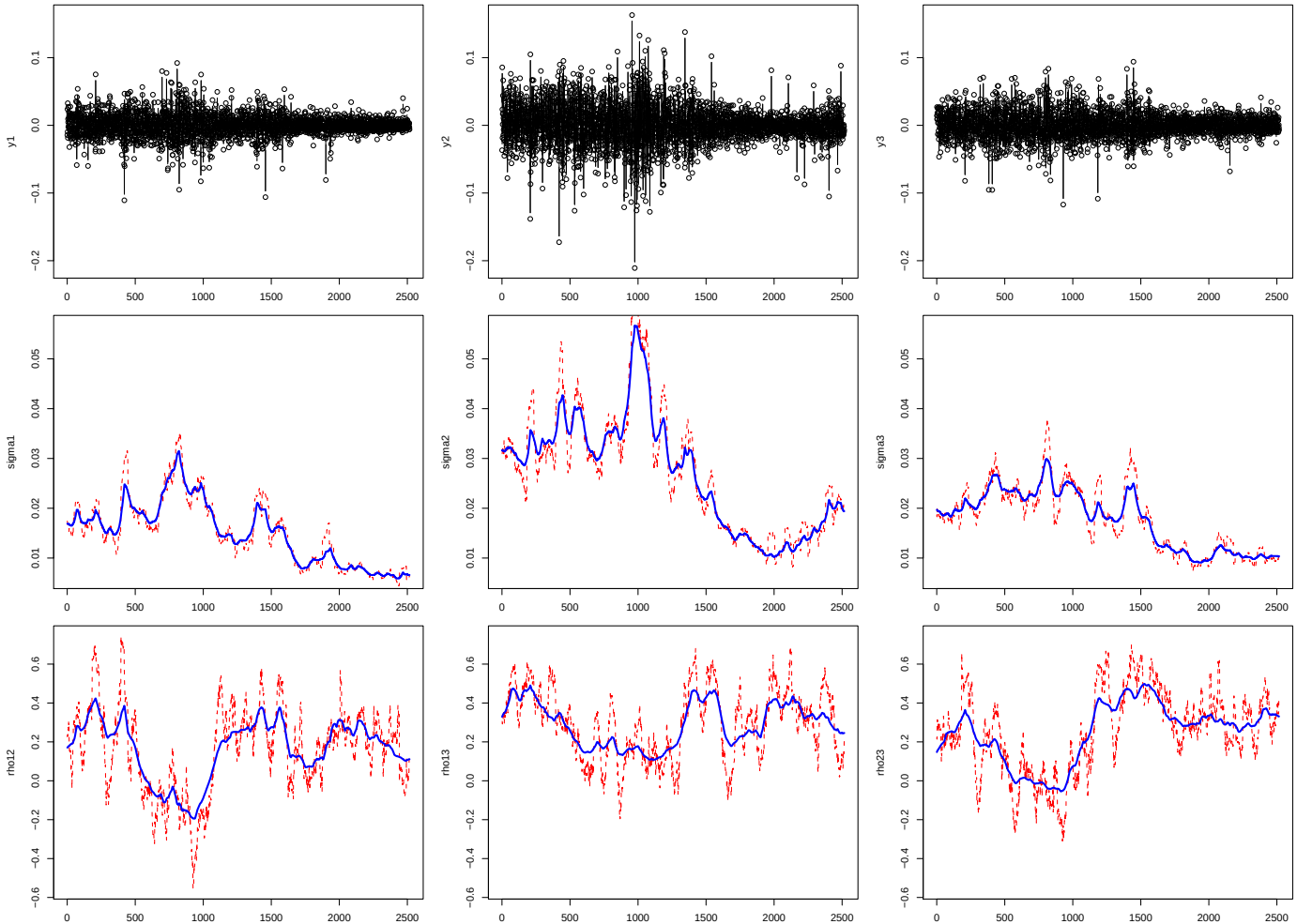
$p = 50 \rightarrow p(p+1)/2 = 1275.$

p=3:

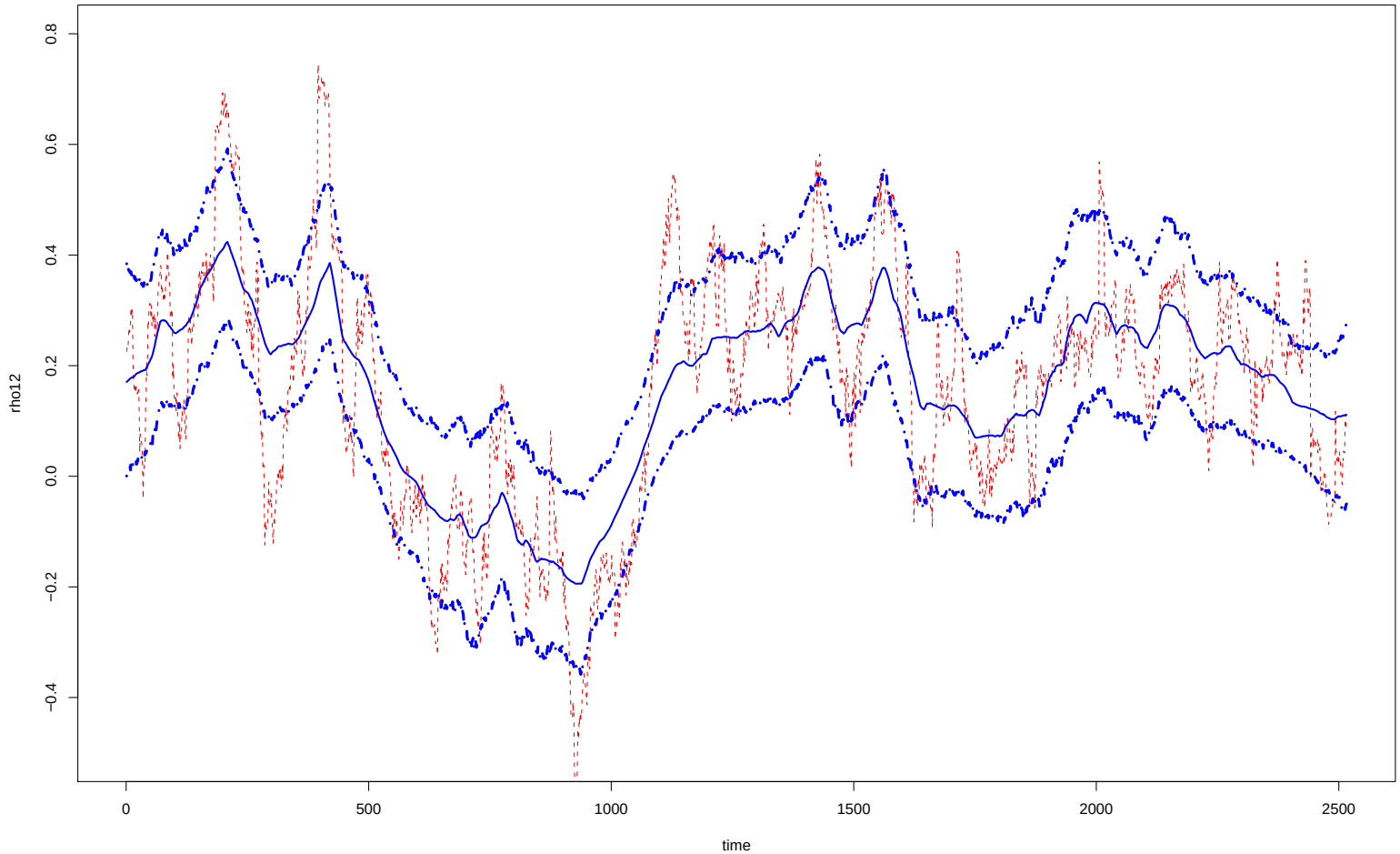
first row: data (Coke, Dell, DuPont)

second row: σ_{it}, third row: ρ_{ijt}.

blue are our stuff, red is sample covariance on a rolling window.



ρ_{12t}: pointwize posterior bands.



We rewrite Σ_t in terms of a series of time-varying regressions.

p=3:

$$\begin{aligned} Y_{1t} &= \exp(d_{t1}/2)\, Z_{1t} && p(Y_{1t}) \\ Y_{2t} &= \phi_{21t}\, Y_{1t} + \exp(d_{2t}/2)\, Z_{2t} && p(Y_{2t} \mid Y_{1t}) \\ Y_{3t} &= \phi_{31t}\, Y_{1t} + \phi_{32t}\, Y_{2t} + \exp(d_{3t}/2)\, Z_{t3} && p(Y_{3t} \mid Y_{2t}, Y_{1t}) \end{aligned}$$

With bigger p, you just keep going!

A high-dimensional non-linear state-space model.

Observe: $\{Y_{it}\}$, $i = 1, 2, \ldots, p$.

States: $\{d_{it}\}$, $i = 1, 2, \ldots, p$, $\{\phi_{ijt}\}$, $j = 1, 2, \ldots, (i-1)$.

At time t, $\Sigma_t \Leftrightarrow (d_{it}, \ldots, d_{pt}, \phi_{12t}, \ldots, \phi_{p(p-1)t})$.

State Equations:

$$d_{it} = \alpha_i + \beta_i \, d_{i(t-1)} + \tau_i \, \epsilon_t.$$

$$\phi_{ijt} = \alpha_{ij} + \beta_{ij} \, \phi_{ij(t-1)} + \tau_{ij} \, \epsilon_t.$$

Priors:

$$p(d_{i0}), \quad p(\phi_{ij0}), \quad p(\alpha_i, \beta_i, \tau_i), \quad p(\alpha_{ij}, \beta_{ij}, \tau_{ij}).$$

MCMC:

Let $\circ$ denote "everything else"
(all other parameters and all the data).

$$
\begin{aligned}
(d_{i0}, \{d_{it}\}_{t=1}^{T}) \mid \circ \\
(\phi_{ij0}, \{\phi_{ijt}\}_{t=1}^{T}) \mid \circ \\
(\alpha_i, \beta_i, \tau_i) \mid \circ \\
(\alpha_{ij}, \beta_{ij}, \tau_{ij}) \mid \circ
\end{aligned}
$$

You just draw each state sequence and the associated AR1 parameters one at a time.

$$
\begin{aligned}
Y_{1t} &= \exp(d_{t1}/2)\, Z_{1t} \\
Y_{2t} &= \phi_{21t}\, Y_{1t} + \exp(d_{2t}/2)\, Z_{2t} \\
Y_{3t} &= \phi_{31t}\, Y_{1t} + \phi_{32t}\, Y_{2t} + \exp(d_{3t}/2)\, Z_{t3}
\end{aligned}
$$

$(d_{i0}, \{d_{it}\}_{t=1}^{T}) \mid \circ :$

$i = p = 3$:

$$
\tilde{Y}_t = Y_{3t} - \phi_{31t}\, Y_{1t} - \phi_{32t}\, Y_{2t} = \exp(d_{3t}/2)\, Z_{t3}
$$

All $(d_{i0}, \{d_{it}\})$ can be drawn as a univariate stochastic volatility.

$$
\begin{aligned}
Y_{1t} &= \exp(d_{t1}/2)\, Z_{1t} \\
Y_{2t} &= \phi_{21t}\, Y_{1t} + \exp(d_{2t}/2)\, Z_{2t} \\
Y_{3t} &= \phi_{31t}\, Y_{1t} + \phi_{32t}\, Y_{2t} + \exp(d_{3t}/2)\, Z_{t3}
\end{aligned}
$$

$(\phi_{ij0}, \{\phi_{ijt}\}_{t=1}^{T}) \mid \circ :$

$i = p = 3, j = 2$:

$$
\tilde{Y}_t = Y_{3t} - \phi_{31t}\, Y_{1t} = \phi_{32t}\, Y_{2t} + \exp(d_{3t}/2)\, Z_{t3}
$$

All $(\phi_{ij0}, \{\phi_{ijt}\})$ can be drawn as in a simple dynamic linear model (DLM).

This draw is done very simply using FFBS.

AR coefficients:

$$(\alpha_i, \beta_i, \tau_i) \mid \circ = (\alpha_i, \beta_i, \tau_i) \mid (d_{i0}, \{d_{it}\})$$

$$(\alpha_{ij}, \beta_{ij}, \tau_{ij}) \mid \circ = (\alpha_{ij}, \beta_{ij}, \tau_{ij}) \mid (\phi_{ij0}, \{\phi_{ijt}\})$$

Now I can tell you what the paper is about!!

Want to do this for "large" p.

$p = 50$: there are $p(p-1)/2 = 1,225$ ϕ sequences to draw.

Two key ideas:

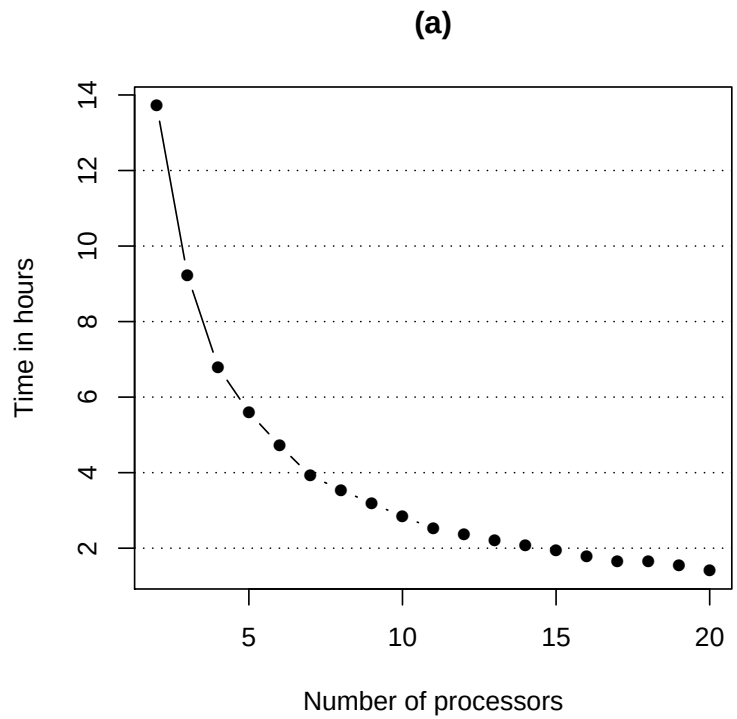
- Draws are done equation by equation.
  Can do blocks of equations separately using
  parallel computing.
- Need very non-standard, highly informative prior
  on the (α, β, τ).

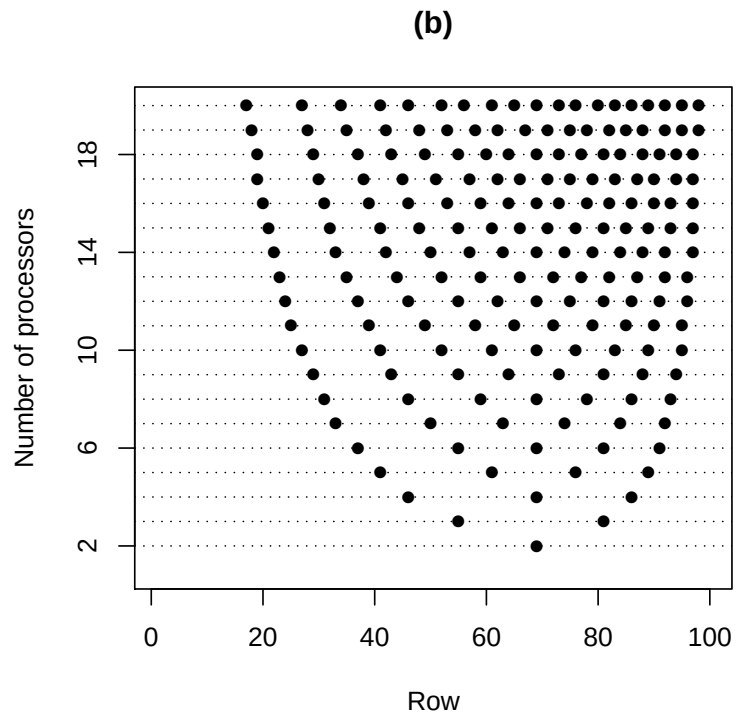
Parallel Computing:

(a): time to compute as a function of the number of processors.
(b): how to allocate equations across processors.

Note: takes 10 times longer to run a univariate SV than a DLM (a d than a ϕ).

$p = 100$.

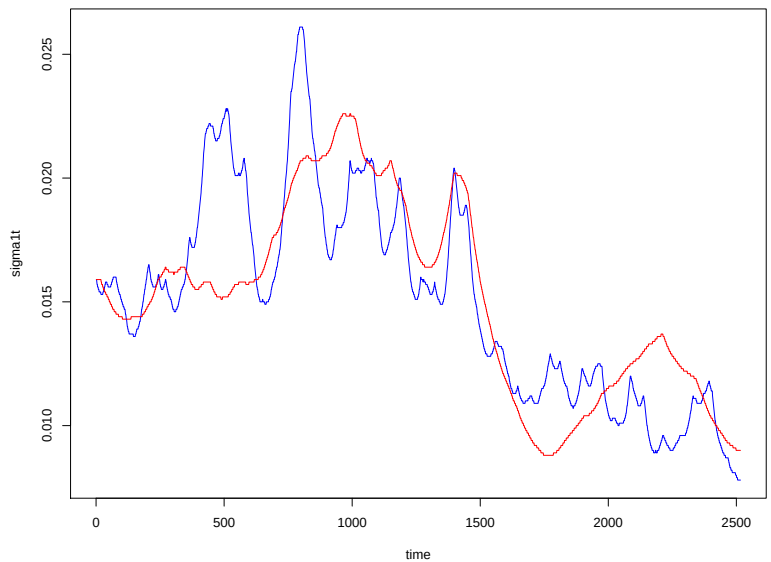




Prior:

With so many things going on, the prior is inevitably influential. It is more like part of the model.

Here is the posterior mean of $\{\sigma_{1t}\}$ from two different priors.



$$d_{it} = \alpha_i + \beta_i \, d_{i(t-1)} + \tau_i \, \epsilon_t.$$

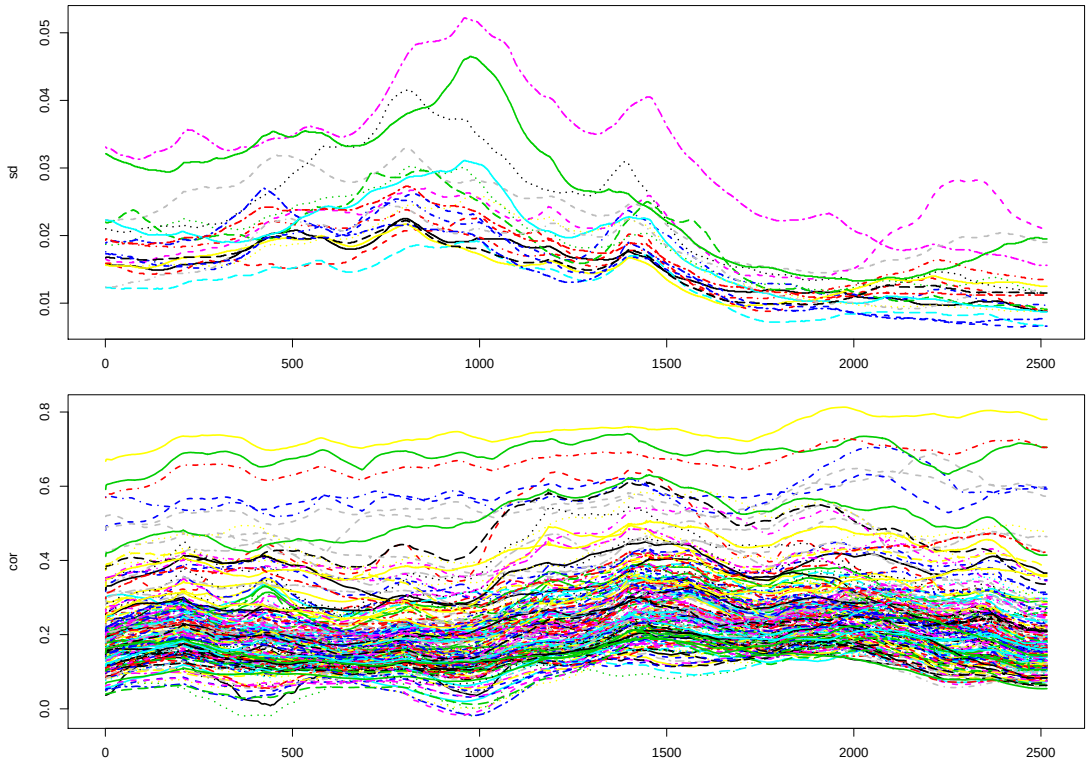
Clearly, prior on τ is a key.

The prior on the (α, β, τ) for the ϕ series is even more important than for the d.

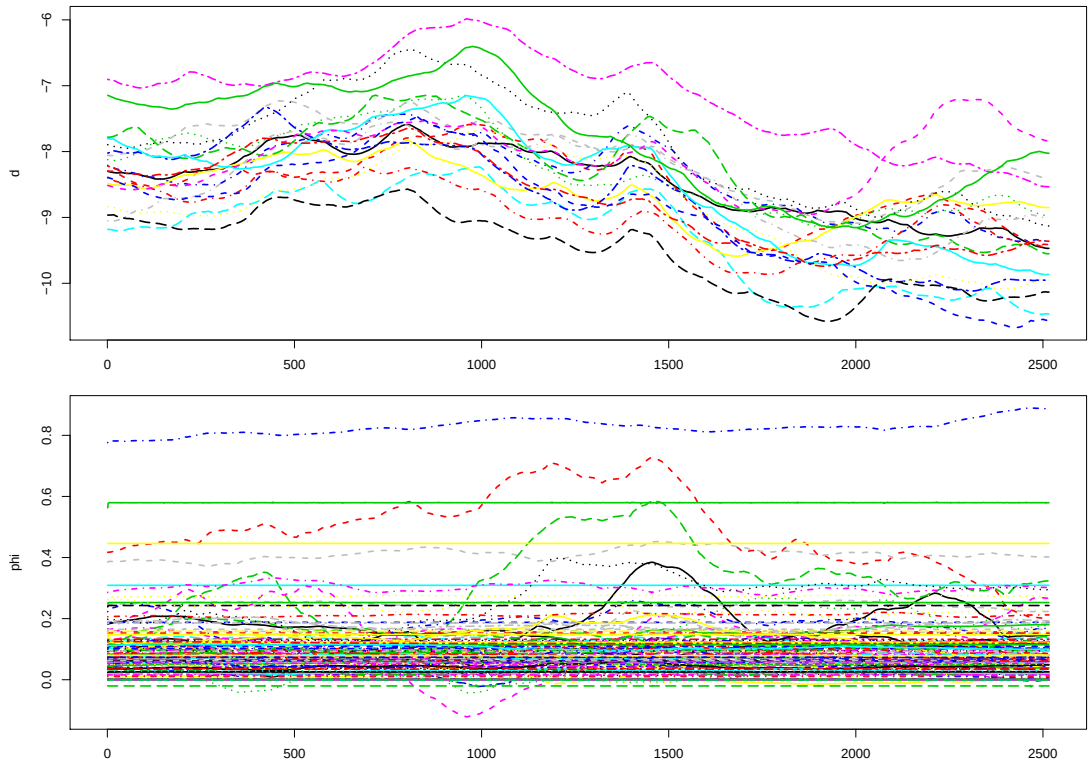
There are many more of them *and* the data is less informative about them.

Example: p=20:

top: all the time varying standard deviations.
bot: all the time varying correlations.



top: all the time varying d.
bot: all the time varying ϕ.



many of the ϕ, flat - line !!!

Key Idea in Bayesian statistics:

> *Prior guides search for parsimony in high dimensional systems !!!!*

Here, "parsimony" means ϕ that don't change much and ϕ that are close to 0.

Prior Specification:

Start by rescaling each series thinking $\Sigma_t \approx I$.

This can be as simple as subtracting off the sample means and dividing by the sample standard deviations.

Have to be able to specify just about any kind of prior we want and draw (α, β, τ) jointly.

But (β, τ) on a bi-variate grid (grid size 100 works).

Then use

$$p(\alpha, \beta, \tau) = p(\beta, \tau)\, p(\alpha \mid \beta, \tau)$$

with $p(\alpha \mid \beta, \tau)$ normal.

$$\phi_t = \alpha + \beta\,\phi_{t-1} + \tau\,\epsilon_t.$$

$$\alpha \mid \beta \sim N(0, \sigma_\alpha^2(1-\beta^2)).$$

When $\beta \approx 0$, the state flat-lines, and α is the level.

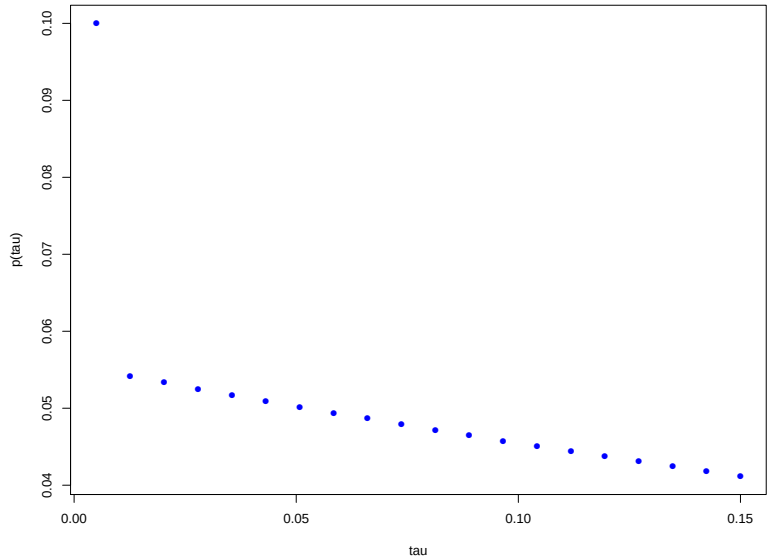
When $\beta \approx 1$ we are close to a random-walk, want $\alpha \approx 0$.

Because we can integrate out α, the draw from the posterior can be done using

$$p(\alpha, \beta, \tau \mid \circ) = p(\beta, \tau \mid \circ)\, p(\alpha \mid \beta, \tau, \circ)$$

where the first draw is on a grid, and the second draw is just a normal.

τ prior:



Pick the probability of the smallest value.

For the other values:

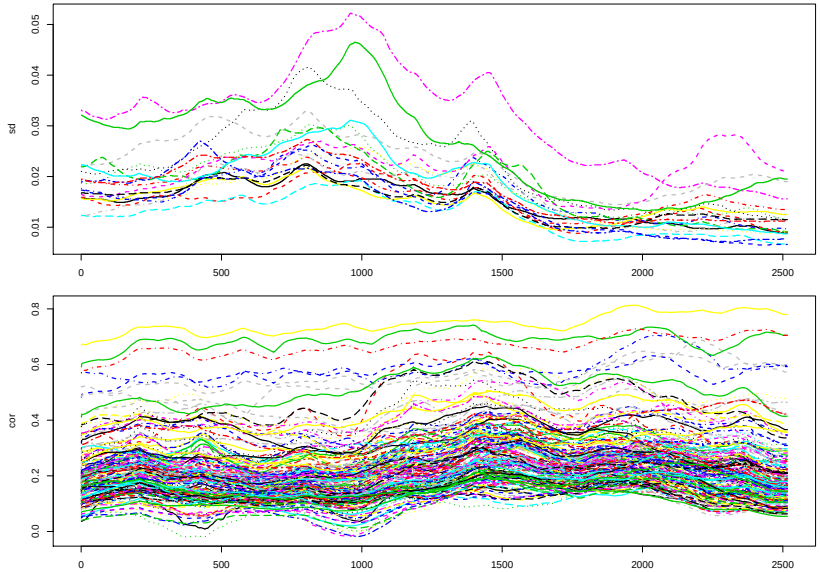
$$p(\tau) \propto \exp(-c|\tau - \tau_{min}|)$$

Push towards small value of τ = smooth state.

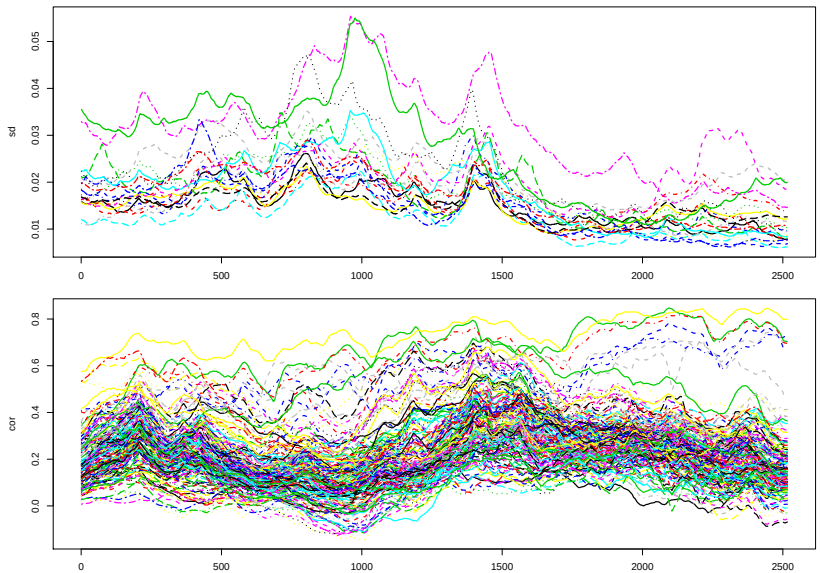
Here, $c = 2$, in practice, $c = 100$.

Prior Parameters: $\tau_{min}, \tau_{max}, p(\tau = \tau_{min}), c$.

$c = 200$.


sd
0.01
0.02
0.03
0.04
0.05
0
500
1000
1500
2000
2500
cor
0.0
0.2
0.4
0.6
0.8
0
500
1000
1500
2000
2500

$c = 100$.


sd
0.01
0.02
0.03
0.04
0.05
0
500
1000
1500
2000
2500
cor
0.0
0.2
0.4
0.6
0.8
0
500
1000
1500
2000
2500

$$\phi_t = \alpha + \beta\,\phi_{t-1} + \tau\,\epsilon_t.$$

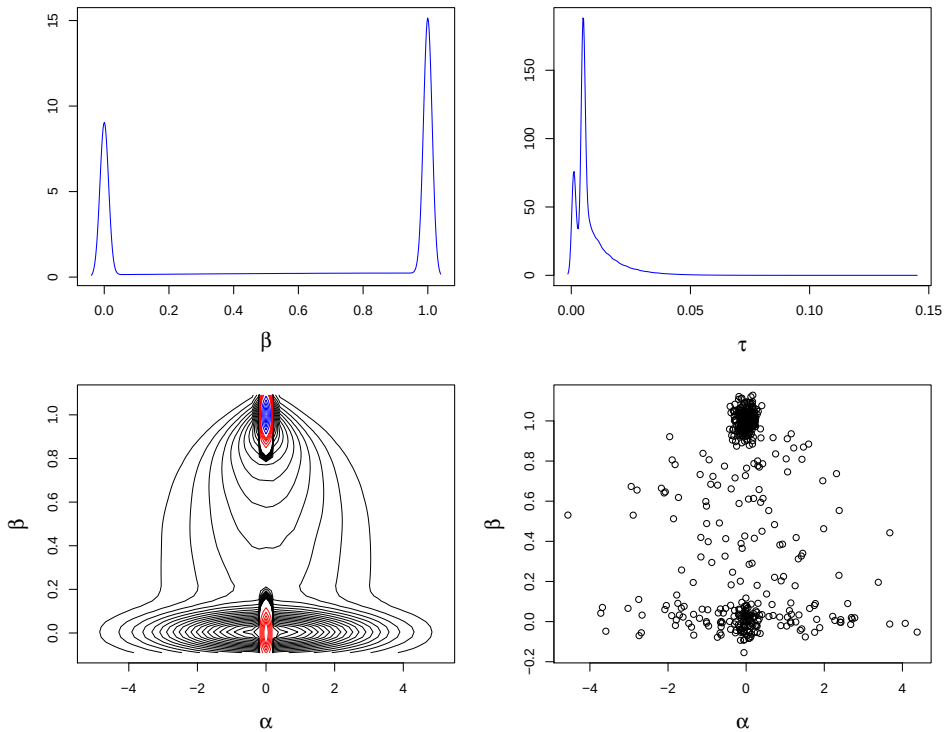
The full prior on (α, β, τ) mixes over configurations of interest.

$$p(\alpha,\beta,\tau) = p_{01}\,\delta_{\{\alpha=0,\beta=1\}}\,p(\tau \mid \beta = 1) + p_{00}\,\delta_{\{\alpha=0,\beta=0\}}\,p(\tau \mid \beta = 0) + p_{u0}\,\delta_{\{\beta=0\}}\,p(\tau \mid \beta = 0)\,p(\alpha \mid \beta = 0) + p_{uu}\,p(\beta)p(\tau \mid \beta \neq 0)\,p(\alpha \mid \beta).$$

- p_{01}: prob of random walk.
- p_{00}: prob of flat-line at 0.
- p_{u0}: prob of flat-line, not at 0.
- p_{uu}: prob $\beta \in (0, 1)$.
- when $\beta = 0$, may want an even smaller τ.

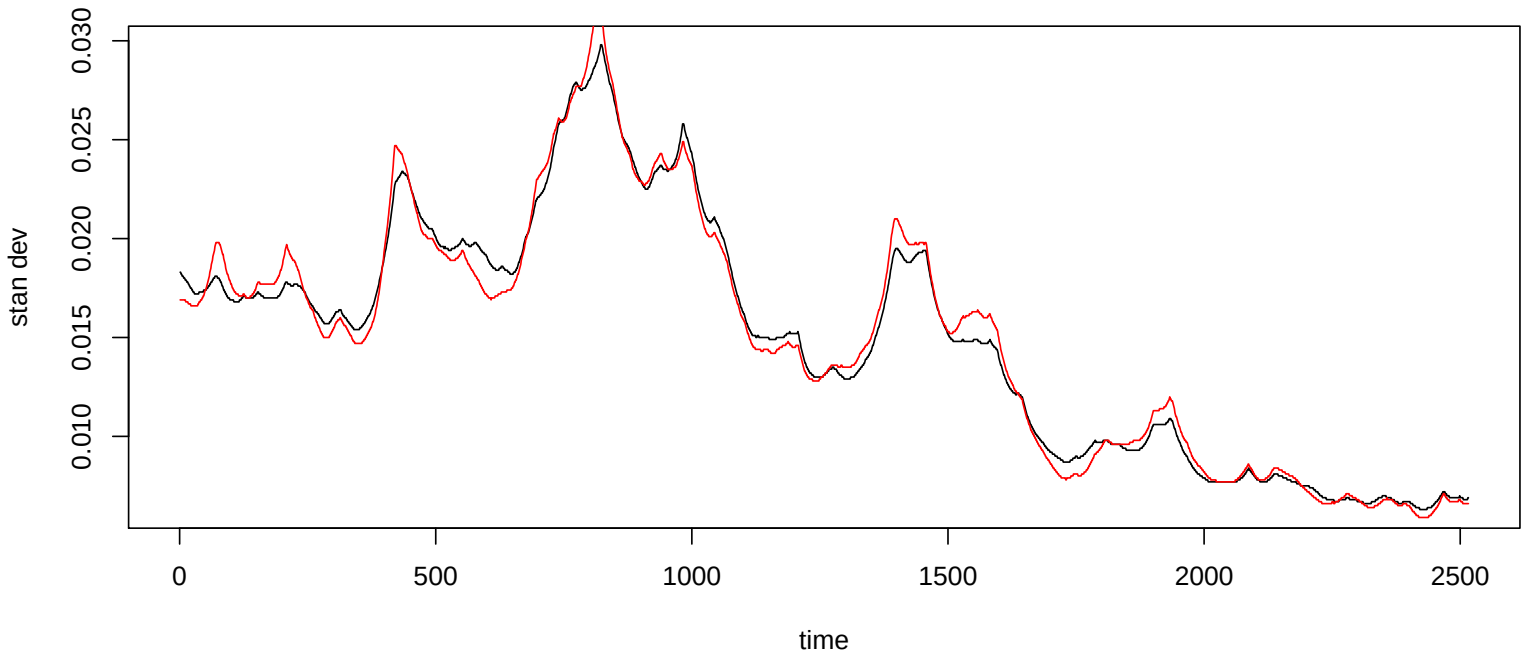
marginals from the prior.

(2,2) is jittered draws, rest are density smooths.



Run $p = 20$ with one order, then reverse the order.

This is the standard deviation of the first (last) series.



Portfolio Weights:

For larger p it is quite difficult to display the fit of the model.

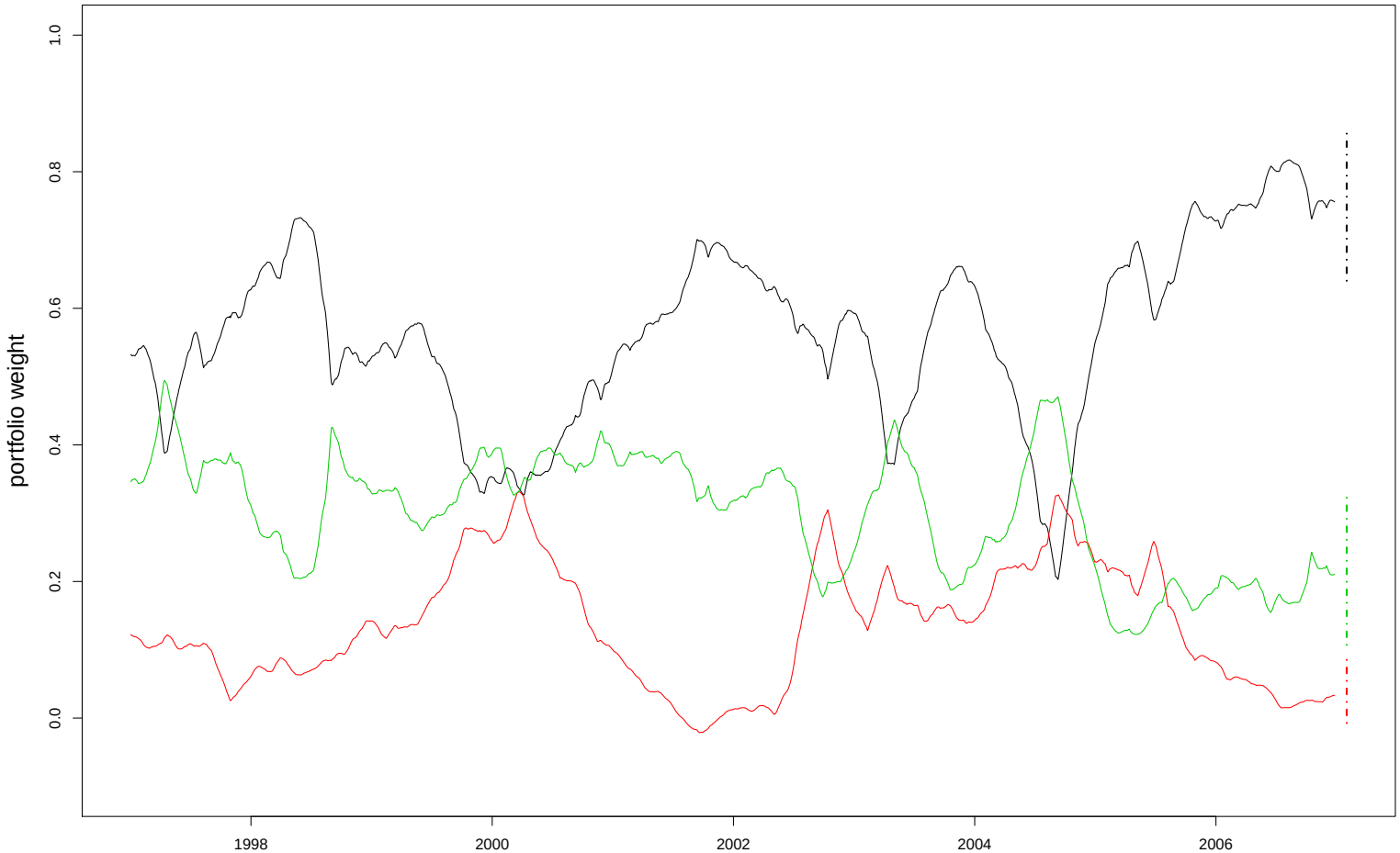
In order to illustrate the fit and use of the model, consider a simple application in the case where all of our series are asset returns.

Let w_t denote the portfolio weights of the *global minimum variance portfolio*. That is, w minimizes $w' \Sigma w$ subject to the constraint that $\sum w_i = 1$.

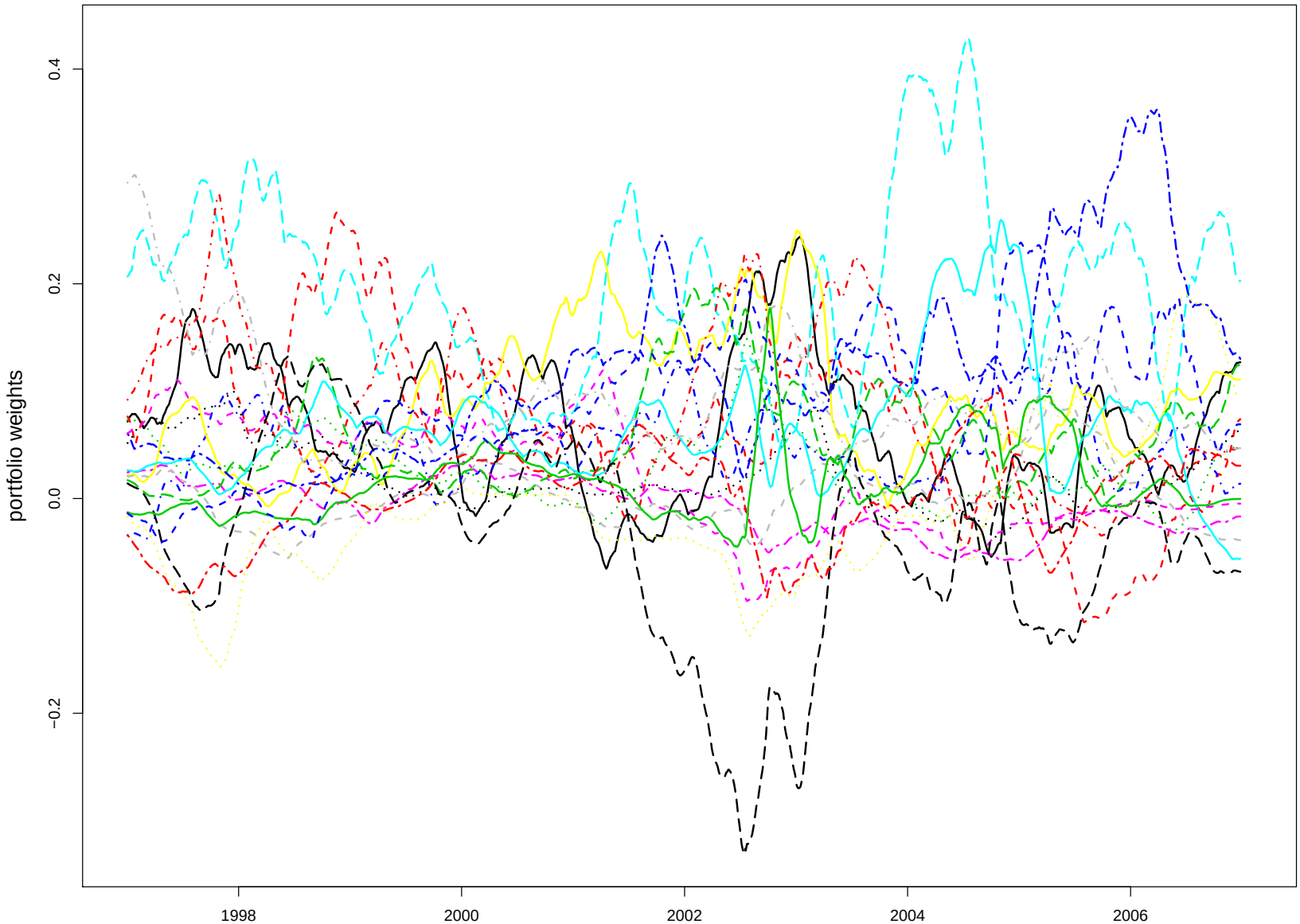
$$w(\Sigma) = \frac{\Sigma^{-1} \iota}{\iota' \Sigma^{-1} \iota}$$

where ι is a vector of ones.

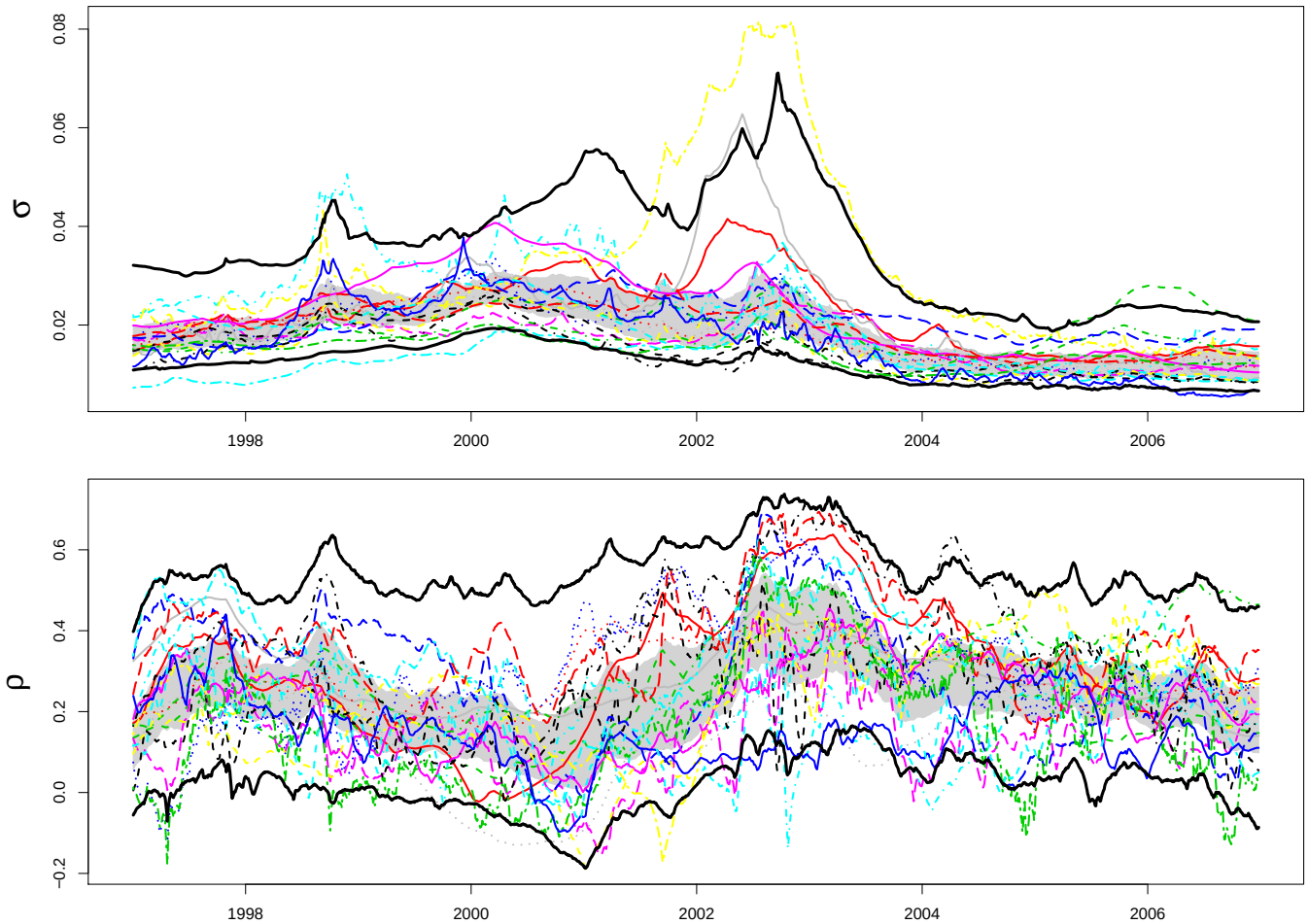
p=3: porfolio weights for the global minimum variance portfolio.



p=20: porfolio weights for the global minimum variance portfolio.



p=94 !!, time varing standard deviations and correlations.



Remarks:

This is a hard problem !!!!

Mixture prior gives you a lot of flexibility.

Can get a constant state with $\beta = 1$ or $\beta = 0$, so the prior identifies which one you get.

Could order the series so that the first so many span the space of factors and then adjust the priors so that it is easier to load on these factors.

Bayesian setup allows us to adjust to outliers, jumps, ...

Example: Discretized Model for Option Pricing

(with Satadru Hore and Hedibert Lopes)

- model parameters: $\lambda = (\beta, \psi, \gamma, \bar{\theta}, \delta, \sigma_\theta, \rho, \sigma_k)$.
- x_i is the strike, i^{th} put option.
- τ is the time till the option expires.
- P_{it}/R_t is the relative option price, i^{th} put option.
- state θ_t is growth rate of economy.

We discretize a continuous time model to obtain a nonlinear state space model for our put option prices.

Observation equation:

$$\frac{P_{it}}{R_t} = f(\theta_t, x_i/R_t, \tau, \lambda) + \sigma\, Z_{it},\ \ i = 1, 2, 3, 4.$$

State equation:

$$\theta_t = \theta_{t-1} + \delta\,(\bar{\theta} - \theta_{t-1})\,\Delta t + \sqrt{\Delta t}\,\sigma_\theta\, Z_t.$$

Example: Stock Market Predictability

(with Carlos Carvalho and Hedibert Lopes)

Our work is an extension of

Pastor, Lubos, and Robert F. Stambaugh, 2009, Predictive systems: Living with imperfect predictors, *Journal of Finance*.

They model our quantities of interest as the system:

$$\begin{aligned} r_{t+1} &= \mu_t + u_{t+1} \\ \mu_{t+1} &= \alpha + \beta\, \mu_t + w_{t+1} \\ x_{t+1} &= A + B\, x_t + v_{t+1} \end{aligned}$$

with

$$Var((u_t, w_t, v_t')') = \Sigma.$$

r: market return
x: predictors (they use 3)
μ: the mean level (the state).

The want to think about μ_t as the anticipated part of next periods return.

μ_t is the conditional expected return, conditioned on a set of information available at time t,

$$\mu_t = E(r_{t+1} \mid \Im_t),$$

where $\Im_t$ denotes the "information".

Predictors are imperfect in that we do **not** want to assume that

$$\mu_t = a + b\, x_t.$$

> *It seem more likely that the predictors are imperfect, in that they are correlated with μ_t, but cannot deliver it perfectly.*

The idea is that there are unobserved variables affecting our players r, μ, and x.

We will never have full information, never observe them all.

They are captured in our model by Σ.

$$\begin{aligned} r_{t+1} &= \mu_t + u_{t+1} \\ \mu_{t+1} &= \alpha + \beta\,\mu_t + w_{t+1} \\ x_{t+1} &= A + B\,x_t + v_{t+1} \end{aligned}$$

with

$$Var((u_t, w_t, v_t')') = \Sigma.$$

$$
\begin{aligned}
r_{t+1} &= \mu_t + u_{t+1} \\
\mu_{t+1} &= \alpha + \beta\,\mu_t + w_{t+1}
\end{aligned}
$$

Key prior information:

> *The predictive system allows us to explore roles for a variety of prior beliefs about the behavior of expected returns, chief among which is the belief that unexpected returns (u_{t+1}) are negatively correlated with innovations in expected return (w_{t+1}).*

Pastor and Stambaugh use economic arguments to motivate the strong belief that

$$\rho_{u,w} < 0.$$

The predictive system model assumes that

$$(u_t, w_t, v_t')' \sim N(0, \Sigma), \quad \text{iid.}$$

A great deal of empirical evidence (and associated methodological development) tells us the assumption of a constant Σ is a bad one (ask Nick Polson !!).

Our goal is to use the predictive systems approach, but include multivariate stochastic volatility.

Instead of just Σ, we want Σ_t, *and* we want to easily incorporate the prior belief that

$$\rho_t = corr(u_t, w_t) < 0, \text{for all } t$$

and possibly other prior beliefs as well.

Pastor and Stambaugh report results strongly questioning the idea that stocks are "good for the long run".

Our preliminary result is that their conclusions may be sensitive to model and prior specification!!!

Note: while the predictive systems are not obviously a state space model many of the ideas (eg. FFBS to draw $\{\mu_t\}$) apply.

Computing

Over the years I've played with various stuff, most notably Python and Java (even Jython).

Long ago, I published a paper on dynamic graphics written in X11.

I still use some of the applets I wrote while learning Java for teaching.

But, after a while I decided I needed to focus on C.

From using Java and Python, I had become enamored with object orientated thinking.

A pure virtual base class in C++ is an interface in Java.

```
//state space model
class spmod
{
public:
   virtual double ptheta0(double theta) = 0; //prior on initial state
   virtual double py(double *y, double theta, int t) = 0; //observation equatio
   virtual double ptheta(double theta, double thetam1) = 0; //state equation
   virtual ~spmod() {}
};
```

Once you get used to the STL and choose a decent matrix class, C++ is pretty nice, I like it.

The sequential algorithms used for state space models cannot be "vectorized".

I coded FFBS for a simple linear DLM in R and and C++ and C++ was 100 times faster.

Another area I work on is Bayesian methods for tree based models. Again, I love C++.

```
class tree {
public:
.....
   //------------------------------
   //tree constructors, destructors
   tree();
   tree(const tree&);
   tree(double);
   ~tree() {tonull();}
   //------------------------------
   //operators
   tree& operator=(const tree&);
.....
   //------------------------------
   //tree functions
   size_t treesize() const; //number of nodes in tree
.....
private:
   //------------------------------
   //parameter for node
......
   //------------------------------
   //rule: left if x[v] < xinfo[v][c]
   size_t v;
   size_t c;
   //------------------------------
   //tree structure
   tree_p p; //parent
   tree_p l; //left child
   tree_p r; //right child
};
```

But, *you have to use R!!*.

Having a open source environment where we can all contribute packages is just fantastic.

It keeps us honest.
Stuff is complicated now, the only way we can know if it works is to have a lot of people try it!

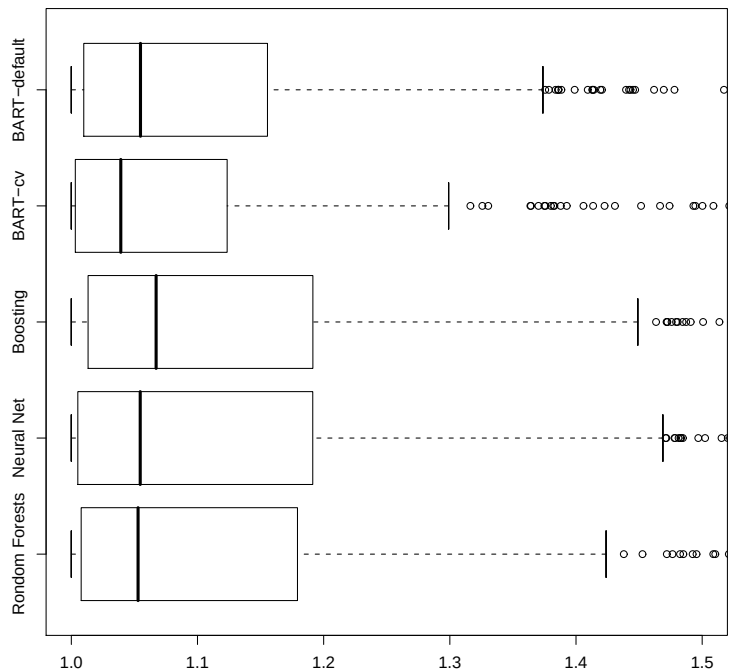
We have a R-package called BayesTree which implements BART: Bayesian Additive Regression Trees.

It is a fantastic package!!

Did out of sample predictive comparisons on 42 data sets.

- p=3 − 65, $n = 100 - 7,000$.
- for each data set 20 random splits into 5/6 train and 1/6 test
- use 5-fold cross-validation on train to pick hyperparameters (except BART-default!)
- gives 20*42 = 840 **out-of-sample predictions**, for each prediction, divide rmse of different methods by the smallest

+ each boxplots represents 840 predictions for a method
+ 1.2 means you are 20% worse than the best
+ BART-cv best
+ BART-default (use default prior) does amazingly well!!



Beat boosting, random forests, and neural-nets!!.

To some extent, my research is driven by what users of BayesTree ask for!!

This seems very healthy to me.

It is a terrible package.

The C++ and the R are horrible!!
(I think the Statistics might be ok).

I think I have improved the C++ and we have an mpi version.

Times for new code, new code with mpi (7 cores), BayesTree package.

	num obs	new-parallel	new-serial	old
1	1000	7	9	43
2	2000	8	18	95
3	3000	9	28	149
4	4000	10	36	204
5	5000	12	45	262
6	10000	18	90	547
7	50000	70	439	NA
8	100000	138	902	NA
9	500000	904	6410	NA

$6410/904 = 7.$

Currently working on:

(i) Fix R part of BayesTree (with George and Chipman)
(ii) A Bayes Finance Package (with Carvalho and Lopes)

Fortunately for me, there is Rccp!
Is rmpi the answer??