To appear in *Quantitative Finance*, Vol. 00, No. 00, Month 20XX, 1–21

Hidden Markov Model for Financial Time Series and Its Application to S&P 500 Index

Stephen H.-T. Lihn†

(*v1.1, July 2017*)

The R package `ldhmm` is developed for the study of financial time series using Hidden Markov Model (HMM) with the lambda distribution framework. In particular, S&P 500 index is studied in depth due to its importance in finance and its long history. Major features in the index, such as regime identification, volatility clustering, and anti-correlation between return and volatility, can be extracted from HMM. Univariate symmetric lambda distribution is essentially a location-scale family of power-exponential distribution. Such distribution is suitable for describing highly leptokurtic time series in the financial market. It provides a theoretically solid foundation to explore such data where the normal distribution may not be adequate. The index is analyzed from two states to six states, then ten states. The five-state HMM and above can capture large amount of auto-correlation, matching what's observed in the data. This is a major validation for the HMM. Although the stock market can be broadly classified into the normal regime and the crash regime, The progression of HMM states allows to go beyond the two-regime paradigm. The index history can be decomposed to a spectrum of volatility states. And the trend of the mean and volatility in HMM states confirms the recognized fact that the stock market tends to rise when the volatility is low, while tends to fall when the volatility is high. The pivotal volatility is calculated. Specifically, we compare the expected volatility from the ten-state HMM to both the VIX index with an adjustment factor and the realized volatility from Oxford-Man Realized Library. They match quite well. This indicates high-state HMM can serve as a tool for volatility forecasting.

Keywords: Hidden Markov Model; lambda distribution; volatility forecasting

JEL Classification: G17; C32; C53

1. The Hidden Markov Model

1.1. *Notation*

The Hidden Markov Model (HMM) implemented in the `ldhmm` package is the **homogeneous first-order HMM**, where the state at time t is only dependent on the states of previous time $t-1$. The mixing distribution is univariate. The parameter space of the HMM is comprised of $\boldsymbol{\pi} = \{\boldsymbol{\theta}, \boldsymbol{\Gamma}, \boldsymbol{\delta}\}$, where $\boldsymbol{\theta}$ is a matrix containing parameters of the mixing distributions, $\boldsymbol{\Gamma}$ is the transition probability matrix, and $\boldsymbol{\delta}$ is the initial state probability vector. In the case of a stationary solution, $\boldsymbol{\delta}$ is also the stationary state distribution vector.

The notations are defined as following:

- The latent states are indexed by an integer, $i = 1, 2, \ldots, m$, where m is number of states.
- The time series is the log-returns of a financial instrument, indexed by an integer, $t = 1, 2, \ldots, T$. The time period can be daily, weekly, monthly, etc.

†Corresponding author. Email: stevelihn@gmail.com; LinkedIn: https://www.linkedin.com/pub/stephen-horng-twu-lihn/0/71a/65.

- The observation at time t in the time series is the log-return between time $t-1$ and t, which is denoted as x_t. And x_t is unbounded. The vector of all observations can be written as $\boldsymbol{x}^{(T)}$.
- The latent state at time t is denoted as C_t, which can have an integer value from $i = 1, 2, \ldots, m$.
- The transition probability matrix is $\{\gamma_{ij}\}$, or simply $\boldsymbol{\Gamma}$ in matrix form. γ_{ij} is the probability to transition from state i to state j. It is independent of t.
- The initial state probability vector is $\{\delta_i\}$, or simply $\boldsymbol{\delta}$ in vector form. $\boldsymbol{\delta}$ also represents **the stationary state distribution** in which $\boldsymbol{\delta\Gamma} = \boldsymbol{\delta}$. Notice that $\boldsymbol{\delta\Gamma}$ is also defined as $\boldsymbol{\alpha}_0$.
- The mixing probability of observations is represented as $P_i(x)$, which is the probability density function (PDF) for x when it is in state i. Or simply $\boldsymbol{P}(x)$ in matrix form, where $\boldsymbol{P}_{ii}(x) = P_i(x)$. That is, $\boldsymbol{P}(x)$ is a diagonal matrix. It is independent of t.

In the financial time series study, we are mostly interested in the stationary solution. The HMM notations outlined in this paper follow closely the book of Zucchini, MacDonald, and Langrock (2016). Note that $\boldsymbol{\delta}$ and $\boldsymbol{\Gamma}$ are subject to unity constraints: $\sum_i \delta_i = 1$ and $\sum_j \gamma_{ij} = 1$.

1.2. *Forward, Backward, and Likelihood*

The forward probability vector $\boldsymbol{\alpha}_t$ is defined as

$$\boldsymbol{\alpha}_t = \begin{cases} \boldsymbol{\delta P}(x_1), & \text{when } t = 1 \\ \boldsymbol{\alpha}_{t-1}\boldsymbol{\Gamma P}(x_t), & \text{when } t = 2, 3, \ldots, T. \end{cases} \tag{1.1}$$

The backward probability vector $\boldsymbol{\beta}_t$ is defined as

$$\boldsymbol{\beta}_t = \begin{cases} \mathbf{1}, & \text{when } t = T \\ \boldsymbol{\Gamma P}(x_{t+1})\,\boldsymbol{\beta}'_{t+1}, & \text{when } t = 1, 2, \ldots, T-1. \end{cases} \tag{1.2}$$

The likelihood expectation at time T is $L_T = \boldsymbol{\alpha}_T\,\mathbf{1'}$, which is the quantity to be maximized in order to find the solution for HMM. And $\mathbf{1}$ is the identity vector, in which every element is 1.

The state probability $Q_i\left(t; \boldsymbol{x}^{(T)}\right)$ is the conditional probability of $C_t = i$ given the observations $\boldsymbol{x}^{(T)}$. It is calculated as

$$Q_i\left(t; \boldsymbol{x}^{(T)}\right) = \frac{\boldsymbol{\alpha}_t(i)\,\boldsymbol{\beta}_t(i)}{L_T}. \tag{1.3}$$

1.3. *Matrix Convention*

Assume $\boldsymbol{V}$ is a vector and $\boldsymbol{M}$ is a matrix. The multiplication between vector and matrix is defined as $\boldsymbol{VM} = \sum_i V_i M_{ij}$ and $\boldsymbol{MV'} = \sum_j M_{ij} V_j$. Summing a vector is represented as $\boldsymbol{V}\,\mathbf{1'} = \sum_i V_i$.

Notice that the PDF matrix $\boldsymbol{P}$ only has non-zero values in diagonal cells. Thus it carries some special properties:

(i) $\boldsymbol{VP}(x)$ results in a vector with element $[\boldsymbol{VP}(x)]_i = V_i P_{ij}(x) = V_i P_i(x)$, that re-weighs each state of $\boldsymbol{V}$ by its mixing PDF.
(ii) $\boldsymbol{MP}(x)$ results in a matrix with element $[\boldsymbol{MP}(x)]_{ij} = M_{ij} P_{jk}(x) = M_{ij} P_j(x)$, which transitions each state i by re-weighing all the states j's by their mixing PDF's, $P_j(x)$.
(iii) The forward algorithm involves $\boldsymbol{V\Gamma P}(x) = V_i \Gamma_{ij} P_j(x)$ which is a vector indexed by j.
(iv) The backward algorithm involves $\boldsymbol{\Gamma P}(x)\,\boldsymbol{V'} = \Gamma_{ij} P_j(x)\, V_j$, which is a vector indexed by i.

In order to prevent over-float and/or under-float in floating point calculation, $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ are implemented recursively by their logarithms in the R package. For the same reason, the maximum likelihood expectation (MLE) optimization minimizes the minus log-likelihood (MLLK), $-\log(L_T)$.

2. Symmetric Lambda Distribution

The `ldhmm` package focuses on one special case of HMM where the mixing distribution of the observations is a symmetric λ distribution, which is essentially an exponential-power distribution. It is briefly introduced as following: The λ distribution is modeled after the state probability function in statistical physics, where the PDF has an exponential form $P(x) \propto e^{y(|z|)}$, e.g. y can be thought of as minus energy levels normalized by temperature. Next, $y(z)$ is assumed to be the solution of a polynomial of the λ-th order, $(\pm y)^{\lambda} + \ldots + c = z^2$, where c is a constant, with the constraint $y \to -\infty$ as $|z| \to \infty$. The skewness can be added by a depressed polynomial term, $\beta z y$, say for the y^3 case (Lihn (2015)). Finally, in order to conform to a location-scale family, we let $z = (x-\mu)/\sigma$. This completes the contruction process.

This broad framework can be refined to focus just on the tail behavior. We set both $\ldots$ and c to zero, and we reach the simplest form of a symmetric λ distribution, $(\pm y)^{\lambda} = z^2$. Thus the PDF used in this package is a two-sided stretched exponential function, defined by the parameter tuple $\theta = (\mu, \sigma, \lambda)$, (See Section 2.2 of Lihn (2015b))

$$P(x;\mu,\sigma,\lambda) = \frac{1}{\sigma\lambda\,\Gamma\left(\frac{\lambda}{2}\right)}\, e^{-\left|\frac{x-\mu}{\sigma}\right|^{\frac{2}{\lambda}}}. \tag{2.1}$$

The shape parameter λ is called the "order" of the distribution (Lihn (2017)). When $\lambda = 1$, it converges to a normal distribution. This form of distribution has a rich history with several different ways of parametrization and construction[1]. Here the PDF is standardized as Eq. (2.1) for our purpose of analyzing financial log-return time series. One major difference is to use $2/\lambda$ instead of β as the power of $|x-\mu|$ since we prefer λ being in the order of a positive low single-digit integer, instead of β being a fraction when $\lambda > 2$.

For each state $i = 1, 2, \ldots, m$, we have $P_i(x) = P(x; \mu_i, \sigma_i, \lambda_i)$. There are three parameters for each state, $\theta_i = (\mu_i, \sigma_i, \lambda_i)$. It is written in matrix form as $\boldsymbol{\theta} = \{\theta_i, i = 1, 2, \ldots, m\}$. Note that μ is an unconstrained real number, while σ and λ must be positive. And λ is expected to be in the range of 1 and 4.

The standard deviation $\Sigma(\sigma,\lambda)$ and kurtosis $K(\lambda)$ of $P(x;\mu,\sigma,\lambda)$ are

$$\Sigma(\sigma,\lambda) = \sigma \left[\frac{\Gamma\left(\frac{3\lambda}{2}\right)}{\Gamma\left(\frac{\lambda}{2}\right)}\right]^{\frac{1}{2}}, \tag{2.2}$$

$$K(\lambda) = \frac{\Gamma\left(\frac{\lambda}{2}\right)\Gamma\left(\frac{5\lambda}{2}\right)}{\Gamma\left(\frac{3\lambda}{2}\right)^2}. \tag{2.3}$$

The kurtosis increases with λ, so does $\Sigma(\sigma,\lambda)/\sigma$. Notice that the scale parameter σ is not the standard deviation, even in the case of $\lambda = 1$. ($\Sigma(\sigma, \lambda = 1) = \sigma/\sqrt{2}$.)

What differentiates this package from other HMM packages is the additional degree of freedom from λ that can accommodate any level of kurtosis in the HMM states. One doesn't have to make assumption that the mixture model has to be made out of Gaussian distributions. If some states

[1] The following are the well known documentations on the Internet:

(i) The `normalp` package in R: https://www.jstatsoft.org/article/view/v012i04/v12i04.pdf

(ii) NIST Dataplot: http://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/pexpdf.htm

(iii) Wolfram: https://reference.wolfram.com/language/ref/ExponentialPowerDistribution.html

(iv) GNU GSL: https://www.gnu.org/software/gsl/manual/html_node/The-Exponential-Power-Distribution.html

(v) Wikipedia: https://en.wikipedia.org/wiki/Generalized_normal_distribution

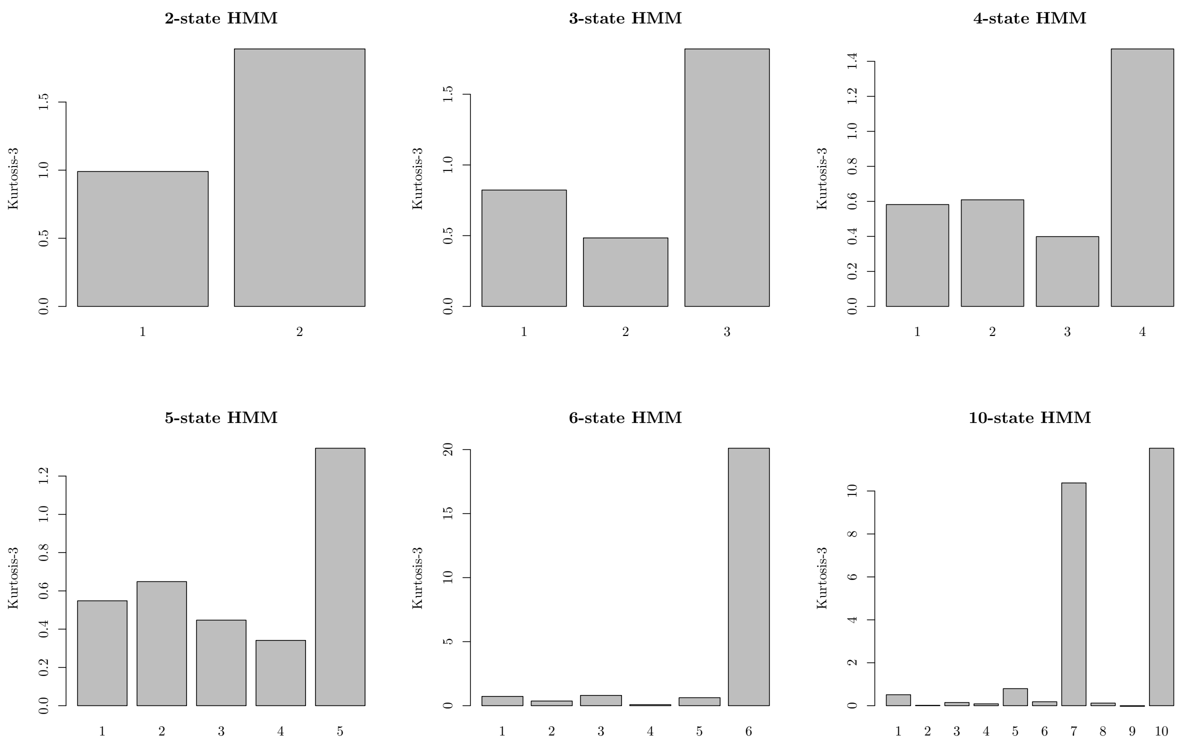

Figure 2.1. Excess kurtosis of the states in six HMM fits for SPX index. In each fit, states are sorted by their standard deviations in ascending order. Large outliers tend to get pushed to the high-volatility state(s). For other states with lower volatilities, excess kurtosis decreases as number of states increases.

turn out to be very close to Gaussian, one should be very delighted. However, we see it as the conclusion drawn from the model, not an assumption going into the model.

In general, kurtosis in each state decreases as number of states increases. Large outliers tend to get pushed to the high-volatility states. This is illustrated in Figure 2.1 with six HMM fits for SPX index, which will be elaborated in following sections. However, excess kurtosis never really "goes away". Thus it is inappropriate to make full assumption of a Gaussian mixture model. It is inconsistent with the data.

The symmetric λ distribution family has been shown to possess some beautiful mathematical properties. For instance, it has a closed form solution for local volatility function and it is related to an elegant mean-reverting stochastic process (Lihn (2017)). It solves the option pricing model for both SPX at $\lambda = 4$ (Lihn (2016)) and VIX at $\lambda \in [2, 4]$ (Lihn (2017b)). Its skew elliptic subfamily ($\lambda = 3$) can fit log-return distributions very well without resorting to mixture models (Lihn (2015)).

2.1. *Expected Volatility of the Mixture*

The log-return and volatility of state i are annualized as

$$\begin{aligned} R_i &= 252 \times \mu_i \\ V_i &= 100 \times \sqrt{252} \times \Sigma\left(\sigma_i, \lambda_i\right), \end{aligned} \tag{2.4}$$

where $\Sigma(\sigma_i, \lambda_i)$ is the standard deviation of state i. It is assumed there are 252 trading days in a year (Shephard and Shephard (2009)). We follow the same convention as in the Oxford-Man Realized Library (Realized (2009)) in order that we can compare our model volatility to its realized volatility, which is published daily.

Let $R(t) = \sum_{i=1}^{m} R_i \, Q_i\left(t; \boldsymbol{x}^{(T)}\right)$ be the expected log-return of the mixture at time t. Then the expected volatility $V(t)$ of the mixture at time t is defined as

$$\left(\frac{V(t)}{100}\right)^2 = \sum_{i=1}^{m} \left[(R_i - R(t))^2 + \left(\frac{V_i}{100}\right)^2 \right] Q_i\left(t; \boldsymbol{x}^{(T)}\right). \tag{2.5}$$

$V(t)$ is one of the most important quantities developed in this paper. $V(t)$ of a well-calibrated HMM can predict the volatility of the stock market. For SPX, $V(t)$ matches the realized volatility calculated from the high-frequency intraday tick data in Oxford-Man realized data library. This will be elaborated in Section 6.

3. Analysis of S&P 500 Index

3.1. *Setting the Stage*

The `ldhmm` package is specifically tailored for the analysis of SPX index. The package comes with the daily closing prices from 01/01/1950 to 12/31/2015. The data can be loaded from

```
> ts = ldhmm.ts_log_rtn("spx", on="days")
```

where `ts` is a list with three components: `ts$d` contains the vector of dates, `ts$x` contains the vector of log-returns, and `ts$p` contains the vector of prices. Hence `ts$x` corresponds to the observations $\{x_t\}$. This function is also capable of appending live daily closing prices from Federal Reserve (FRED) with the option of `fred.data=TRUE`. This is important for daily forecasting purpose.

There are two distinct features in the daily returns of SPX. First, its kurtosis is very large, around 30. Even after 10 largest outliers are dropped, the kurtosis is still as large as 8.5. This is computed as

```
> sapply(0:10, function(drop) kurtosis(ldhmm.drop_outliers(ts$x, drop)))
[1] 30.279540 12.479094 11.631622 10.983566 10.505473 10.041342
     9.595356  9.252240  8.903898  8.664750  8.518157
```

Second, the auto-correlation of the absolute of the returns $|x_t|$ is more than 20%. This feature is called "volatility clustering". The market data's ACF can be calculated via the `ldhmm.ts_abs_acf` function:

```
> ldhmm.ts_abs_acf(ts$x, drop=0, lag.max=6)
[1] 0.2444656 0.2737506 0.2502788 0.2434303 0.2809230 0.2389925
> ldhmm.ts_abs_acf(ts$x, drop=10, lag.max=6)
[1] 0.2271080 0.2461340 0.2290556 0.2375244 0.2520413 0.2273132
```

We see that SPX's absolute ACF hovers around 23-28% for the first 6 lags. Dropping 10 largest outliers doesn't help reducing the absolute ACF much. The `ldhmm` package will have to address these two fundamental features in order to be a suitable tool to model SPX daily data.

3.2. *Two States by Normal Distribution*

The analysis starts with two states. As shown in Bae, Kim, and Mulvey (2014) and Mulvey and Liu (2016), the stock market can be classified to two regimes: **the normal regime** where the market is calm and rising, and **the crash regime** where the market is panic and plunges down. The former is also called "the bull market" while the later "the bear market". Empirical evidence shows that the market spent "most" of time in the normal regime.

We will use the two-state analysis to introduce the usage of the main functions in the package, and show why the λ distribution is needed. Many functions operate on the internal slots of the input `ldhmm` object and produce an enriched output `ldhmm` object.As the first experiment, we shall set up the analysis using the normal distribution, which has two parameters: μ and σ. To train the HMM, one needs to initialize an `ldhmm` object called `h` with $\boldsymbol{\pi} = \{\boldsymbol{\theta}, \boldsymbol{\Gamma}, \boldsymbol{\delta}\}$, where $\boldsymbol{\theta}$ is the `param` parameter for the mixing distributions $\boldsymbol{P}(x)$, and the `gamma` parameter for the transition probability matrix $\boldsymbol{\Gamma}$. If you don't know how to specify $\boldsymbol{\Gamma}$, you can use the `ldhmm.gamma_init` function to set up a very naive initial matrix. We always assume `stationary=TRUE`, therefore the initial state probability vector `delta` needs not to be provided. We use `m=2` to denote number of states, the `ldhmm` constructor is called as following,

```
> m = 2
> param0 = matrix(c(
      mu_1, sigma_1,
      mu_2, sigma_2), m, 2, byrow=TRUE)
> gamma0 = ldhmm.gamma_init(m)
> h <- ldhmm(m, param0, gamma0, stationary=TRUE)
```

If this is your first time analyzing a time series, you will have to guess what μ_i and σ_i should be. In this case, (μ_1, σ_1) is approximately $(0.0006, 0.01)$ and (μ_2, σ_2) is approximately $(-0.0007, 0.02)$. Later in Section 6.3, we will propose a universal law on the relation between μ and $\Sigma(\sigma, \lambda)$, which can also be used to bootstrap the `param` parameter.

Then we invoke the maximum likelihood expectation (MLE) optimizer with the object `h` and the log-return vector `ts$x` as the training observations:

```
> hd <- ldhmm.mle(h, ts$x, decode=TRUE, print.level=2)
```

by default, it uses `nlm` to minimize MLLK. You can also switch to other supported optimizers by changing the slot `mle.optimizer`. The optimization result $\boldsymbol{\pi} = \{\boldsymbol{\theta}, \boldsymbol{\Gamma}, \boldsymbol{\delta}\}$ is stored in the returned object `hd`. When the flag `decode=TRUE` is set, the result object is further enriched with the decoded state information based on `ts$x`. The decoding algorithm is in the `ldhmm.decoding` function.

We now inspect the content of the object `hd` for our stationary solution. The mixing distribution parameters are

```
> hd@param
                mu       sigma
[1,]  0.0005841565 0.008990645
[2,] -0.0006888613 0.023348432
```

The transition probability matrix $\boldsymbol{\Gamma}$ is

```
> hd@gamma
           [,1]       [,2]
[1,] 0.98835236 0.01164764
[2,] 0.03858714 0.96141286
```

And the stationary state probability vector $\boldsymbol{\delta}$ is

```
> hd@delta
[1] 0.768136 0.231864
```

These results can be accomplished by other HMM softwares. E.g. one can easily generate the same result via `depmixS4` as a way of validation.

However there is an issue if we look deeper into the data. The kurtosis isn't right. The theoretical statistics of each state is

```
> ldhmm.ld_stats(hd)
              mean          sd kurtosis
[1,]  0.0005841565 0.006357346        3
[2,] -0.0006888613 0.016509834        3
```

Compare this to the statistics of the data classified in each state, as shown below:

```
> hd@states.local.stats
              mean          sd  kurtosis    skewness length
[1,]  0.0006007835 0.006333934  3.565389 -0.02491462  12868
[2,] -0.0007847466 0.016749016 14.828955 -0.71360850   3737
```

We see that the kurtoses of the data are not 3. In fact, the kurtosis of the second state is very high (14.8), and it has a pronounced skewness (-0.7). Thus, although we obtain some result, the statistics of theoretical model is inconsistent with what's observed in the data. This is the main reason for using λ distribution, which can properly account for the behavior of the tails and produce consistent statistics.

3.3. *Two States by λ Distribution*

The symmetric λ distribution has three parameters, $\theta_i = (\mu_i, \sigma_i, \lambda_i)$. To run the analysis, one needs to initialize the `param` parameter and construct the input object `h` as below:

```
> m = 2
> param0 = matrix(c(
      mu_1, sigma_1, lambda_1,
      mu_2, sigma_2, lambda_2), m, 3, byrow=TRUE)
> gamma0 = ldhmm.gamma_init(m)
> h <- ldhmm(m, param0, gamma0, stationary=TRUE)
```

You can simply use $\lambda = 1$ if you are not sure how to guess λ, and reference μ and σ from your previous fits. The MLE optimizer is pretty good at converging to the proper λ. Now you can follow the same procedure to invoke the MLE optimizer and obtain the trained object `hd`.

```
> hd <- ldhmm.mle(h, ts$x, decode=TRUE, print.level=2)
```

We can take a look at the content of the result. The mixing distribution parameters are

```
> hd@param
                mu       sigma   lambda
[1,]  0.0006179039 0.006958493 1.418630
[2,] -0.0003604997 0.013167718 1.710065
```

We see that both states have λ larger than 1, indicating both states are leptokurtic.

The transition probability matrix $\boldsymbol{\Gamma}$ is

```
> hd@gamma
           [,1]        [,2]
[1,] 0.99339098 0.006609019
[2,] 0.01631407 0.983685929
```

And the stationary state probability vector $\boldsymbol{\delta}$ is

```
> hd@delta
[1] 0.7116872 0.2883128
```

The statistics of the data classified in each state can be shown by

```
> hd@states.local.stats
              mean          sd  kurtosis    skewness length
[1,]  0.0005747770 0.006296764  3.821433 -0.05581264  11977
[2,] -0.0004506958 0.015358800 17.010688 -0.79468256   4628
```

On the other hand, the theoretical statistics of each state is

```
> ldhmm.ld_stats(hd)
              mean          sd kurtosis
[1,]  0.0006179039 0.006326634 3.990282
[2,] -0.0003604997 0.014766363 4.890138
```

The kurtosis of the first state matches (3.82 vs 3.99), but the kurtosis of the second state still differ quite a bit. What is going on? This is because there are very large outliers in the data that can not be captured by the two-state model. The package provides the following utility to drop those outliers then calculate statistics. It turns out that, after 11 largest moves are dropped (out of 4600 observations), the kurtosis of the second state can be reconciled:

```
> ldhmm.calc_stats_from_obs(hd, drop=11)
              mean          sd kurtosis    skewness length
[1,]  0.0005778304 0.006250343 3.657341 -0.04427838  11966
[2,] -0.0003396821 0.014416549 4.847647 -0.04041189   4617
# notice the reduced kurtosis and skewness of [2,]
```

Not only the kurtosis is rectified, the large skewness of the second state is also remediated. The concept of dropping the few largest outliers is called "asymptotic statistics", which was discussed in Section 8.1 of Lihn (2015).

3.4. *AIC and BIC*

In addition to better capturing kurtosis, we also observe that λ distribution is superior than normal distribution by comparing the MLLK, AIC, and BIC, as shown in the following table:

	MLLK	AIC	BIC
Normal	-56086	-112160	-112114
Lambda	-56473	-112929	-112867

According to both AIC and BIC, the choice of λ distribution is more appropriate. We will use AIC and BIC as our guide for model selection, as we gradually increase the number of states in the following sections.

3.5. *Interpretation of Two States - Regime Identification*

Now we are in a good shape in terms of modeling, we shall make some interpretations based on the two-state HMM result:

(i) The first state represents **the normal state**, where the mean is positive. The second state represents **the crash state**, where the mean is negative.
(ii) The standard deviation of the crash state is more than twice of the normal state, depicting the panic and volatility of the bear markets. The higher kurtosis in combination with a negative return also indicates more frequent violent moves during sell-off.
(iii) According to the stationary state probability vector `delta`, the market spent 70% of its time in normal state.

The result from HMM answers the empirical observations with solid numbers.

The regime identification is to classify the stock market into a few regimes that human can understand intuitively. Two-state HMM are ideal for such task due to its simplicity. In this package, the best way to visualize the states is to compare the expected volatility $V(t)$ from Eq. (2.5) to the realized volatility calculated from Oxford-Man realized variance data. This can be accomplished by the following function:

```
> ldhmm.oxford_man_plot_obs(hd)
```

The graph is shown in Figure 3.1. $V(t)$ is derived from the 2-state HMM trained by the SPX history from 1950 to 2015, then decoded locally via the log-returns of SPX2.r time series from 2000 till now. The main finding is that $V(t)$ mirrors the trend of the realized volatility calculated from SPX2.rv. In terms of regime identification, we observe that the low volatility state corresponds to the bull markets, e.g. very cleanly from mid-2003 to mid-2007, and from 2013 to mid-2015, and extremely low volatility period from mid-2016 to 2017. The high volatility state corresponds to the

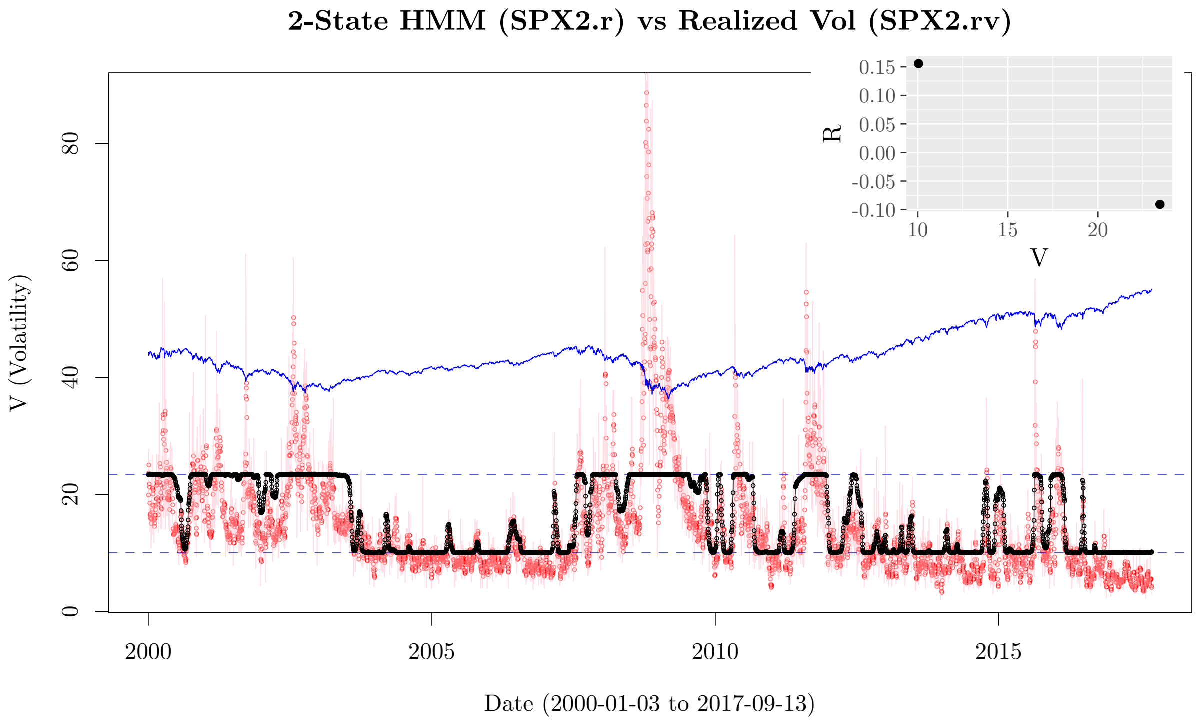

Figure 3.1. Comparison of the expected volatility $V(t)$ from two-state HMM (black) vs the realized volatility (red) from Oxford-Man realized variance data set. Two-state HMM is the simplest model to illustrate the normal and crash regimes. The red line is the daily realized volatility, and the red dots are the 5-day moving average. The dash blue lines are the volatilities of the HMM states. The solid blue line is the rescaled SPX price index. The black dots in the insert show the (R_i, V_i) values of each state.

bear markets, e.g. the dot-com bubble collapse from 2000 to mid-2003, and the financial crisis from mid-2007 to 2009, the mid-2010 and mid-2011 corrections, and the correction from H2 of 2015 to Q1 of 2016.

3.6. *Kurtosis and ACF*

Now we can examine whether two-state HMM helps to understand the issues of kurtosis and ACF. Simulated observations can be generated as following

```
> hs <- ldhmm.simulate_state_transition(hd, init=100000)
> kurtosis(hs@observations)
[1] 8.542578
```

The model kurtosis matches SPX data after 10 largest outliers are dropped. So the model is in a good range. The reader should be warned that kurtosis is an unstable statistics, you may get somewhat different numbers each time you run.

The compute-intensive simulation that generates the model's absolute ACF is packaged into the `ldhmm.simulate_abs_acf` function:

```
> ldhmm.simulate_abs_acf(hd, n=100000, lag.max=1)
[1] 0.1389113
```

The ACF number is too low, it is nowhere near 23% that we are expecting. Thus the two-state HMM has some degree of volatility clustering but can not fully reflect the feature for SPX. We must increase number of states. The issue of ACF will be examined again in Section 5.

4. The More States The Better

We will explore the HMM with more states, from 3, 4, 5, 6, then 10. By carefully crafting the `param` parameter with different ranges of standard deviation, the MLE optimizer can converge reasonably well. The following are the results for 3, 4, 5, 6 states. The 10-state result is presented in Section 5.1. The main takeaway is that the MLE is optimized at five states based on AIC and BIC. And ten-state HMM brings us very close to the State Space Model (SSM) where the states are almost continuous in a certain range.

4.1. *Three States*

The three-state HMM produces better AIC and BIC scores than the two-state HMM. The MLLK is -56799, AIC is -113569, and BIC is -113453. Its result is shown below:

```
> hd@param
                 mu       sigma   lambda
[1,]  0.0007057892 0.005933416 1.356518
[2,]  0.0003024532 0.011299302 1.221545 # the intermediate state
[3,] -0.0011976005 0.018191048 1.689612
> hd@gamma
             [,1]       [,2]         [,3]
[1,] 9.858362e-01 0.01366418 0.0004996217
[2,] 1.299432e-02 0.98201140 0.0049942767
[3,] 9.997684e-05 0.02298345 0.9769165696
> hd@delta
[1] 0.4264269 0.4639618 0.1096113
> ldhmm.ld_stats(hd)
               mean          sd kurtosis
[1,]  0.0007057892 0.005176394 3.822568
[2,]  0.0003024532 0.009052779 3.484238
[3,] -0.0011976005 0.020085198 4.820435
> hd@states.local.stats
               mean          sd  kurtosis length
[1,]  0.0006740378 0.005097534  3.754966   7206
[2,]  0.0003038977 0.009099652  3.445072   7719
[3,] -0.0014313206 0.020958050 12.971801   1680
> ldhmm.calc_stats_from_obs(hd, drop=3)[3,] # remove 3 days of outliers!
               mean          sd kurtosis length
[3,] -0.0014237631 0.019883053 4.772711   1677
> hs <- ldhmm.simulate_state_transition(hd, init=100000)
> kurtosis(hs@observations)
[1] 10.85158
```

We find that

(i) An intermediate state (2) emerges between the normal state (1) and the crash state (3).
(ii) The crash state now has a much larger negative return and volatility.
(iii) According to `gamma`, The market spent only 11% of time in state 3 (vs 43% in state 1, and 46% in state 2). The "true" crash state is meaningfully isolated.
(iv) From `gamma`, we see that there is only 0.55% chance getting into state 3, but has 2.3% chance getting out of it.
(v) We only need to remove 3 days of outliers to make the kurtosis of the third state consistent.
(vi) An important pattern begins to develop in the output of `ldhmm.ld_stats`: the states align with the increasing volatilities. The larger the volatility the smaller the return.
(vii) The simulated model kurtosis of the mixture is increased to nearly 11. As more states are added with different means and variances, the kurtosis will increase steadily.

Figure 4.1 shows the comparison of the expected volatility $V(t)$ from three-state HMM vs the

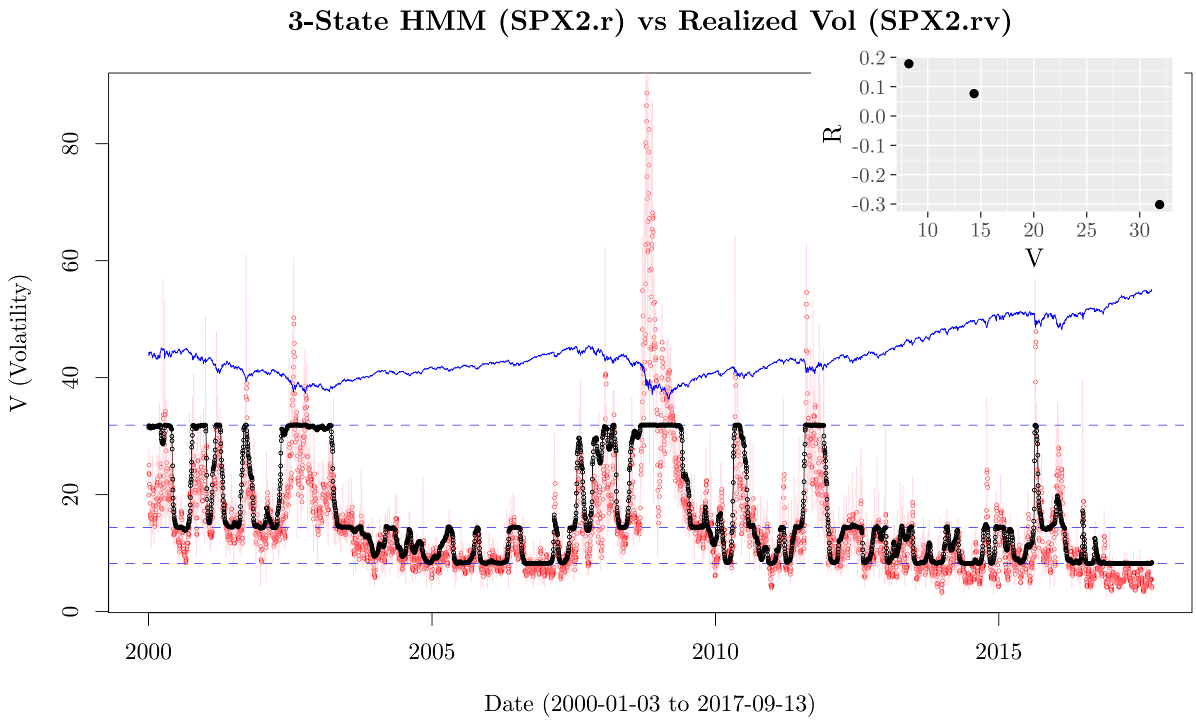

Figure 4.1. Comparison of the expected volatility $V(t)$ from three-state HMM (black) vs the realized volatility (red) from Oxford-Man realized variance data set. Three-state HMM reveals the intermediate state or the so-called transition state. The red line is the daily realized volatility, and the red dots are the 5-day moving average. The dash blue lines are the volatilities of the HMM states. The solid blue line is the rescaled SPX price index. The black dots in the insert show the (R_i, V_i) values of each state.

volatility from Oxford-Man realized variance data set. The worldview isn't binary anymore. We observe that the bull market is a condition of staying in state 1 and 2 persistently. On the other hand, the bear market was formed by oscillations between state 2 and 3. Maybe the definitive start of a bull market can be defined as the first touch of state 1, while the definitive start of a bear market as the first touch of state 3 (which ends previous bull market).

4.2. *Four States*

The AIC and BIC scores continue to improve for four states. The MLLK is -56913, AIC is -113779, and BIC is -113594. In this fit, we only need to remove one day of outlier in state 4 to make the kurtosis match. Obviously the outlier is Black Monday of 1987. Its HMM result is shown below:

```
> hd@param
                 mu       sigma   lambda
[1,]  9.196106e-04 0.005229458 1.261847
[2,]  3.757789e-04 0.008665867 1.272882
[3,] -6.420789e-05 0.014844040 1.185018
[4,] -1.498255e-03 0.026035525 1.582056
> floor(hd@gamma*10000)/100
      [,1]  [,2]  [,3]  [,4]
[1,] 97.77  2.06  0.16  0.00
[2,]  0.91 98.31  0.77  0.00
[3,]  0.00  1.69 97.59  0.70
[4,]  0.23  0.27  4.00 95.48
> hd@delta
```

```
[1] 0.21250126 0.50825276 0.24145344 0.03779254
> ldhmm.ld_stats(hd)
              mean          sd kurtosis
[1,]  9.196106e-04 0.004294759 3.581640
[2,]  3.757789e-04 0.007166091 3.608837
[3,] -6.420789e-05 0.011634897 3.398529
[4,] -1.498255e-03 0.026543720 4.470772
> hd@states.local.stats
              mean          sd kurtosis length
[1,]  0.0009779361 0.004145592 3.338066   3534
[2,]  0.0003718359 0.007141628 3.633649   8462
[3,] -0.0001573969 0.011778537 3.252953   3943
[4,] -0.0017771802 0.028106106 9.779386    666
> ldhmm.calc_stats_from_obs(hd, 1)[4,] # remove 1 day!
              mean          sd kurtosis length
[4,] -0.0014354959 0.026707143 4.111900    665
> hs <- ldhmm.simulate_state_transition(hd, init=100000)
> kurtosis(hs@observations)
[1] 14.06909
```

The volatility of states is refined from (0.5%, 0.9%, 2%) to (0.4%, 0.7%, 1.2%, 2.7%). The third state has 1.2% volatility and almost zero return, which fits the concept of the transition state. If it transitions back to state 1 or 2, then the bull market resumes. If it transitions to state 4, the crash state, then the market suffers violent downward moves. And notice that the market only spent 4% of time in the devastating crash state (However, it was not so infrequent since 2000).

4.3. *Five States*

The AIC and BIC scores improves slightly. The MLLK is -56971.67, AIC is -113873, and BIC is -113603. In this fit, we only need to remove one day of outlier in state 5 to lower the kurtosis. Its HMM result is shown below:

```
> hd@param
                 mu       sigma   lambda
[1,]  0.0009930523 0.005075172 1.248172
[2,]  0.0002569163 0.008111946 1.288760
[3,]  0.0005442314 0.012434221 1.205838
[4,] -0.0010363320 0.021245575 1.160083
[5,] -0.0045300495 0.040342119 1.541332
> floor(hd@gamma*10000)/100
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,] 97.40  2.57  0.01  0.00  0.00
[2,]  1.07 98.10  0.62  0.19  0.00
[3,]  0.13  1.14 98.17  0.44  0.10
[4,]  0.00  0.01  2.48 97.08  0.41
[5,]  0.00  0.00  0.05  6.16 93.78
> hd@delta
[1] 0.19101121 0.42689342 0.27665911 0.09439302 0.01104324
> ldhmm.ld_stats(hd)
              mean          sd kurtosis
[1,]  0.0009930523 0.004132910 3.548254
[2,]  0.0002569163 0.006775294 3.648374
[3,]  0.0005442314 0.009867994 3.447088
[4,] -0.0010363320 0.016409895 3.341393
[5,] -0.0045300495 0.039941736 4.345499
> hd@states.local.stats
              mean          sd kurtosis length
[1,]  0.0010151286 0.004003505 3.348072   3034
[2,]  0.0002467643 0.006683865 3.622080   7306
[3,]  0.0005733122 0.009970374 3.313509   4578
```

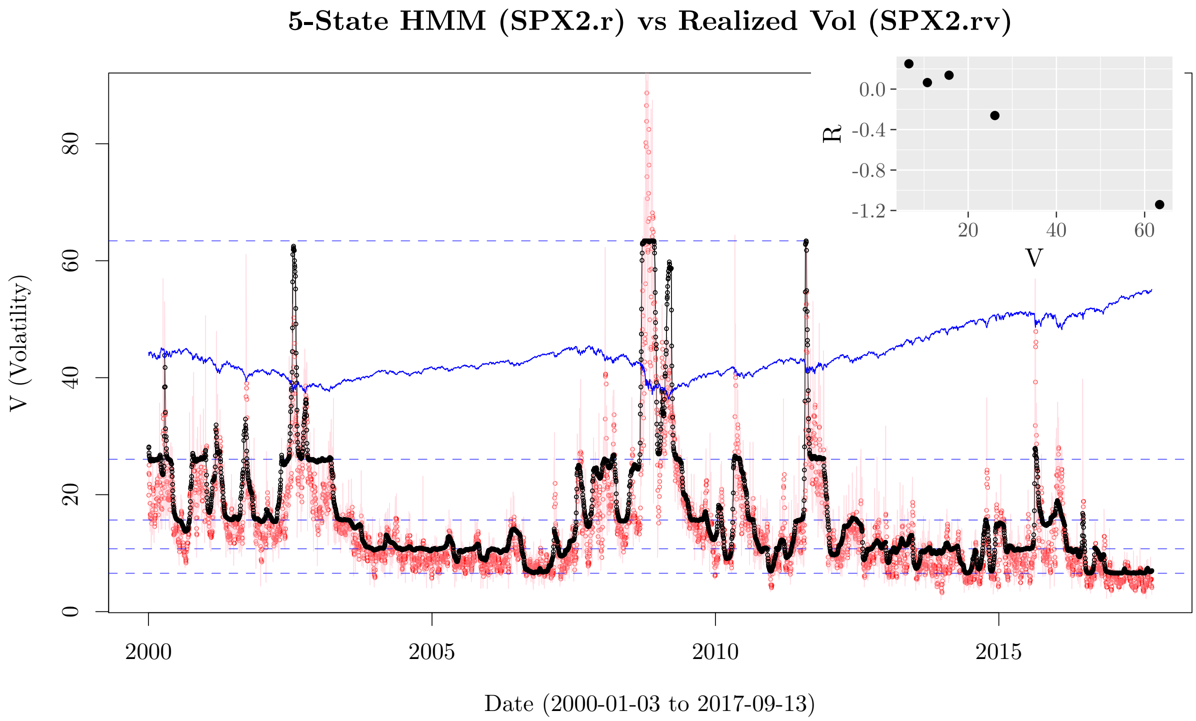

Figure 4.2. Comparison of the expected volatility $V(t)$ from five-state HMM (black) vs the realized volatility (red) from Oxford-Man realized variance data set. Five-state is considered the optimal HMM according to AIC and BIC. The red line is the daily realized volatility, and the red dots are the 5-day moving average. The dash blue lines are the volatilities of the HMM states.

```
[4,] -0.0011935638 0.016867423 3.385824    1520
[5,] -0.0053585776 0.042302368 6.829236     167
> ldhmm.calc_stats_from_obs(hd, 1)[5,] # remove 1 day!
              mean           sd kurtosis length
[5,] -0.0040113568 0.038670313 2.993876     166
> hs <- ldhmm.simulate_state_transition(hd, init=100000)
> kurtosis(hs@observations)
[1] 18.30003
> ldhmm.simulate_abs_acf(hd, n=200000, lag.max=6)
[1] 0.2458919 0.2467940 0.2386975 0.2339076 0.2219662 0.2147192
```

The five-state HMM has several attractive features:

(i) From AIC and BIC's perspective, five-state is the most optimized HMM in model selection.
(ii) The excess kurtosis of states, as shown in Figure 2.1, is most evenly distributed.
(iii) The complexity of states are not too much and not too less. There are two low-volatility states (1, 2, bull market), one transition state (3), and two high-volatility states (4, 5, bear market).
(iv) The simulated kurtosis of 18.3 is reasonably high to reflect what's observed in the data.
(v) The first few lags of the absolute ACF (24%) are very close to what's observed in the data.

Figure 4.2 shows the comparison of the expected volatility $V(t)$ from five-state HMM vs the volatility from Oxford-Man realized variance data set. We observe that the bull market is a condition to stay in the two low-volatility states, state 1 and 2, persistently. On the other hand, the bear market was formed by oscillations between state 3 and 4. And the fifth state is very destructive.

The diagonal cells of the transition probability matrix (`hd@gamma`) are above 97% for the lower 4 states. This means these HHM states are distinct states (think of them as quantum states), and at any given time, the market state is pushed to almost a single state. Only the fifth state has a lower 94% diagonally, which allows it to transition to state 3 and 4 with significant probability. In next section, we will present the ten-state HMM, whose structure of transition probability matrix is very different. The probabilities of diagonal cells are much lower, many are in the range of 90%.

4.4. *Six States*

The AIC and BIC scores no longer improves. The MLLK is -56962, AIC is -113828, and BIC is -113457. What is interesting is that the sixth state can pick up those days of extremely high volatility, making its kurtosis at 23. But other than this feature, it doesn't appear to add value to other aspects of the HMM. From the six states and above, we begin to observe the fine structure in the data, where the volatilities of two neighboring states (state 2 and 3) are almost the same (degenerate) but they have very different μ's.

Here we only show the `param` result and the statistics of the mixing distributions:

```
> hd@param:
                mu       sigma   lambda
[1,]  0.0010213403 0.004999996 1.319998
[2,] -0.0002915723 0.008400014 1.170002
[3,]  0.0013732134 0.008900007 1.350000
[4,] -0.0003144338 0.015699970 1.040001
[5,] -0.0007789220 0.023000009 1.279999
[6,] -0.0053364110 0.007000002 3.880000
> ldhmm.ld_stats(hd)
              mean          sd  kurtosis
[1,]  0.0010213403 0.004259999  3.727569
[2,] -0.0002915723 0.006525897  3.363990
[3,]  0.0013732134 0.007731493  3.805421
[4,] -0.0003144338 0.011337999  3.081299
[5,] -0.0007789220 0.019104501  3.626500
[6,] -0.0053364110 0.066665445 23.107859 # large kurtosis
```

5. Auto-Correlation and Ten-State HMM

One of the most important validations for the HMM is whether it can model the auto-correlation in the absolute of log-returns. After the six-state HMM, we venture to the ten-state HMM, which possesses some interesting properties not seen in the lower-state HMM's.

5.1. *Ten-State HMM*

The statistics of the mixing distributions is shown below:

```
> s <- data.frame(ldhmm.ld_stats(hd))
> s[order(s$sd),]
            mean          sd  kurtosis
1   0.0010218351 0.003896462  3.510961
3   0.0017494924 0.005476218  3.017826
2  -0.0009153308 0.006529737  3.146678
6   0.0011360615 0.008168016  3.087349
4  -0.0003438331 0.009366239  3.794715
7   0.0006138718 0.009694191  3.184742
5  -0.0004305358 0.010531909 13.376046 # large kurtosis
8  -0.0008658711 0.012999651  3.119932
9  -0.0000020415 0.018809720  2.969632
10 -0.0038099143 0.036805330 14.992067 # large kurtosis
```

There are two states (5 and 10) that have large kurtosis. There are two states (2 and 3) around stdev = 0.005 with two large opposite returns. These are the fine structures of the market data. The largest moves don't necessarily have to be bucketed into the most volatile state.

The transition probability matrix $\boldsymbol{\Gamma}$ is:

```
> floor(hd@gamma*10000)/100
       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
[1,]  95.57  3.58  0.84  0.00  0.00  0.00  0.00  0.00  0.00  0.00
[2,]   3.49 91.94  3.64  0.91  0.00  0.00  0.00  0.00  0.00  0.00
[3,]   1.03  4.67 90.62  2.95  0.71  0.00  0.00  0.00  0.00  0.00
[4,]   0.00  1.04  5.66 90.65  1.64  0.98  0.00  0.00  0.00  0.00
[5,]   0.00  0.00  1.08  5.60 87.28  4.98  1.03  0.00  0.00  0.00
[6,]   0.00  0.00  0.00  1.06  2.04 92.02  3.91  0.94  0.00  0.00
[7,]   0.00  0.00  0.00  0.00  0.83  4.06 90.96  3.25  0.87  0.00
[8,]   0.00  0.00  0.00  0.00  0.00  1.01  4.53 90.48  3.09  0.86
[9,]   0.00  0.00  0.00  0.00  0.00  0.00  1.03  4.53 91.37  3.05
[10,]  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1.00  4.34 94.64
```

It shows a feature that other lower-state HMM's don't have. That is, each state has about 90% probability to stay in the same state, and has large enough probability to move to two states above and two states below. Thus at any given time, the state probability is spread across several states. (Think of it as superposition of quantum states.) Thus we have a spectrum of almost continuous state distribution. Such flexibility produces much better long-range dependency and more accurate volatility prediction. Ten-state HMM has an impressive ACF up to 200 lags, which is discussed next.

5.2. *Absolute ACF of 5-State and 10-State*

The market data's ACF can be calculated via the `ldhmm.ts_abs_acf` function:

```
> ldhmm.ts_abs_acf(ts$x, lag.max=6)
[1] 0.2444656 0.2737506 0.2502788 0.2434303 0.2809230 0.2389925
```

We see that the the market's ACF hovers around 24-28% for the first 6 lags.

The model's absolute ACF can be simulated via the `ldhmm.simulate_abs_acf` function. The simulation takes place in a similar fashion to how $\boldsymbol{\alpha}_t$ is calculated. The initial sampling simulates $\{(C_1, x_1)\}$ such that their distributions reflect $\boldsymbol{\delta P}(x_1)$. Each subsequent leg $\{(C_t, x_t)\}$ is obtained by applying $\boldsymbol{\Gamma P}(x_t)$ to $\{(C_{t-1}, x_{t-1})\}$. And the absolute auto-correlation is calculated between $|x_{t-1}|$ and $|x_t|$. We use a large sample size, `n=200,000`, so that tail probability can be properly accounted for. The following shows the ACF comparison from two-state to ten-state:

```
> ldhmm.simulate_abs_acf(hd, n=200000, lag.max=6)
# 2-state
[1] 0.1498026 0.1469878 0.1405985 0.1409026 0.1309578 0.1337567
# 3-state
[1] 0.2019167 0.1965990 0.1948321 0.1898597 0.1828851 0.1853974
# 4-state
[1] 0.2243816 0.2222023 0.2096111 0.2046574 0.2008934 0.1967752
# 5-state
[1] 0.2458919 0.2467940 0.2386975 0.2339076 0.2219662 0.2147192
# 6-state
[1] 0.2148965 0.2187334 0.1952582 0.1872222 0.1857958 0.1709845
# 10-state
[1] 0.2275259 0.2208111 0.2113389 0.2072190 0.2014405 0.2071603
```

The ACF is too low for two-state and three-state HMM. We see the five-state HMM can capture the highest level of ACF - above 24% in the first lag, which is very close to the market data. Thus we postulate that five-state HMM can forecast a decent amount of correlation for the first few lags. We also observe that six-state HMM can not improve ACF in its first few lags.

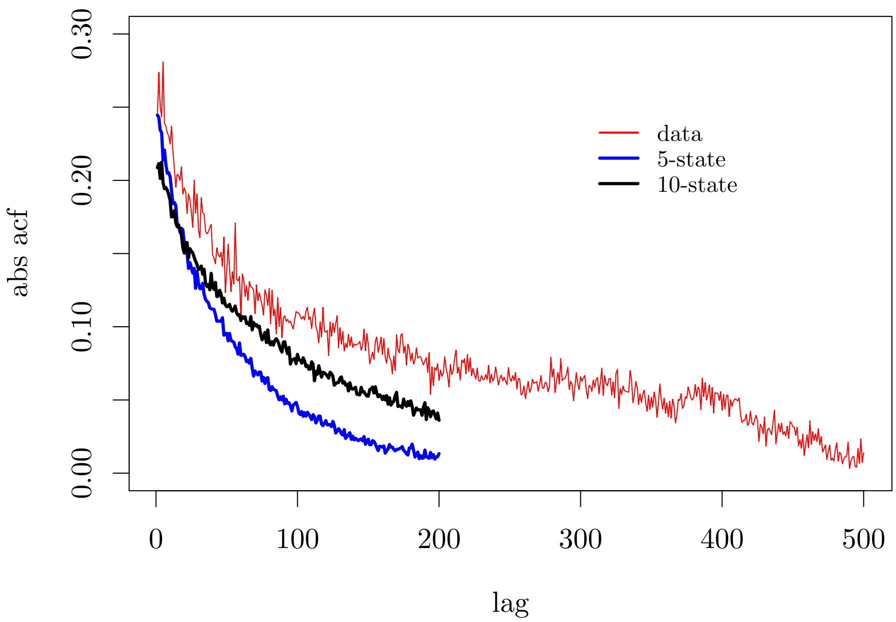

Figure 5.1. Absolute ACF of the five-state (blue) and ten-state (black) HMM up to 200 lags. Red line is the absolute ACF of the SPX index.

However, if we compare the long-range dependency up to 200 lags, we observe that five-state ACF dies out quickly, but ten-state ACF stays relatively high. In Figure 5.1, we examine the 200-lag ACF figures of five-state (blue) and ten-state (black) HHMs, compared to the SPX data (red). Overall, the ACF curve of ten-state performs much better than five-state, except the five-state is superior for the first few lags. Both five-state and ten-state have merits. But since the ten-state HMM produces what we call a "volatility spectrum", it is more attractive and is the focus of exploration in the next section.

The reason that ten-state HMM has very good long-range dependency is probably related to its more dispersed transition probability matrix $\boldsymbol{\Gamma}$ (see Section 5.1). The distances between adjacent states are not large, so at any given time, the state probability can be spread across several states. This allows the HMM to better associate past observations with new observations, and produce continually adjusted expected volatility.

6. Volatility Prediction

From our experience of fitting multi-state HMM, we begin to see that the HMM states are the volatility states. As number of states increases, the concept of a volatility spectrum emerges. Thus we would like to have as many states as we possibly can as long as we can still maintain statistical significance. (When there are very few observations in each state, one can not interpret the meaning of the states anymore. This is the danger of overfitting.) We find that ten-state HMM is a stable model that can represent the concept of volatility spectrum very well.

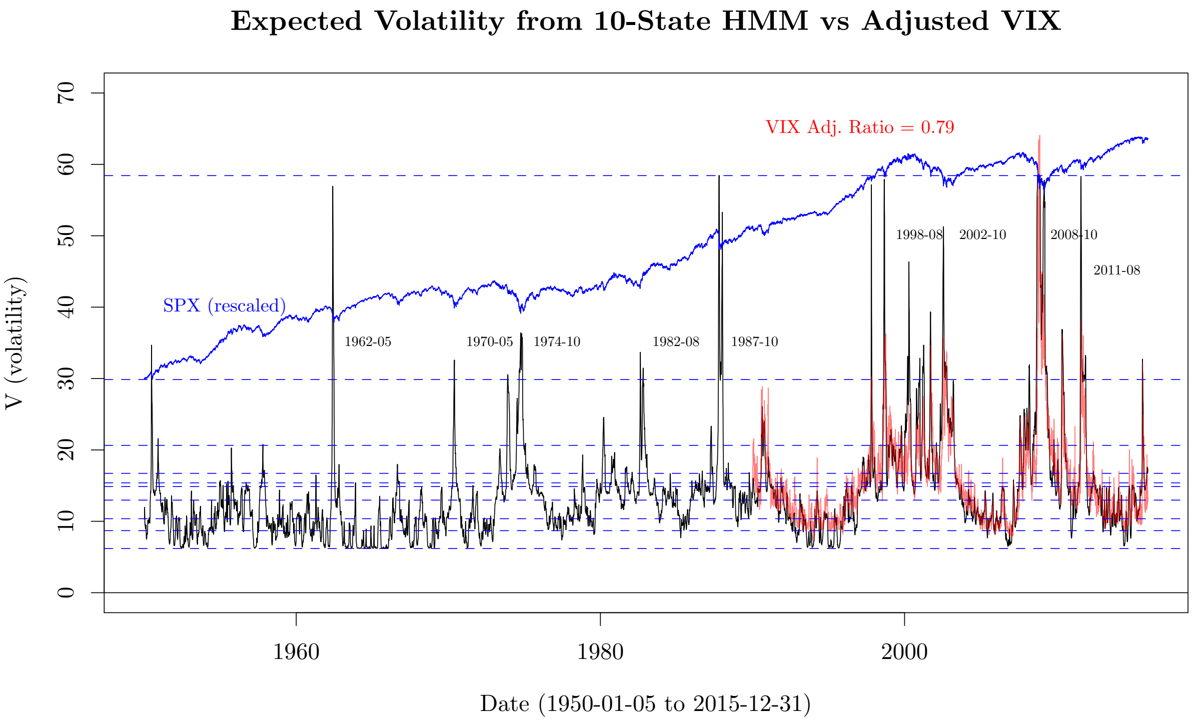

Figure 6.1. The expected volatility $V(t)$ of ten-state HMM (black) vs adjusted VIX index (red). The fit is based on all SPX data from 1950 to 2015. VIX is adjusted by the long-term average of the daily ratios between $V(t)$ and VIX during the overlapping period. This ratio is 0.79. The dash blue lines are the volatility of the HMM states. The SPX log-price level is drawn in blue line as a reference so that each spike can be identified with well-known bear markets in history.

6.1. *Historical Volatility from Ten-State HMM*

We use the ten-state HHM to demonstrate its capability of volatility forecasting. The package comes with the trained `ldhmm` object so you don't have to redo the time-consuming fitting. We load this object and calculate the historical volatility as below:

```
> hs <- ldhmm.read_sample_object()
> spx <- ldhmm.ts_log_rtn(on="days")
> hss <- ldhmm.decoding(hs, spx$x)
> V <- ldhmm.decode_stats_history(hss, annualize=TRUE)[,"V"]
> plot(spx$d, V, type="l")
```

Figure 6.1 shows the expected volatility $V(t)$ of the ten-state HMM result from 1950 to 2015 (black). The first attempt to understand what $V(t)$ means is to compare it to the VIX index (available since 1990). It is found that VIX is consistently higher than $V(t)$. We calculate the long-term average of daily ratios between $V(t)$ and VIX during the overlapping period. This ratio is 0.79, which is used to adjust VIX down as shown in the red line. The reader can observe that $V(t)$ and the adjusted VIX matches quite well. Thus $V(t)$ is indeed a quantity that has real-world meaning.

The second attempt is to compare $V(t)$ to the realized volatility from the Oxford-Man Realized Library (available since 2000). To plot the volatility with Oxford-Man data,

```
> hs <- ldhmm.read_sample_object()
> ldhmm.oxford_man_plot_obs(hs)
```

The most recent data file will be downloaded automatically from the website of Oxford-Man Institute. Figure 6.2 shows the expected volatility $V(t)$ from the ten-state HMM decoded by SPX2.r

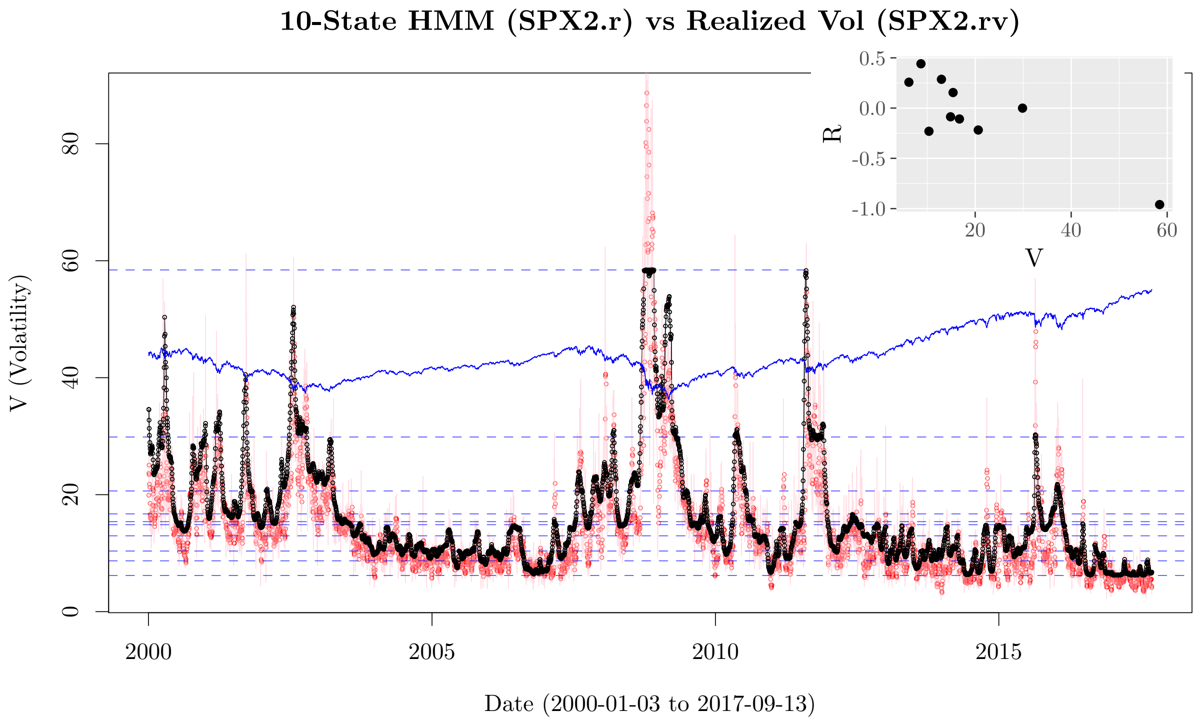

Figure 6.2. Comparison of the expected volatility $V(t)$ from ten-state HMM (black) vs the volatility from Oxford-Man realized variance data set (red). The red line is the daily realized volatility, and the red dots are the 5-day moving average. The ten states are able to label the realized volatility almost continuously, which is close to a State Space Model (SSM). And $V(t)$ matches the moving average of Oxford data reasonably well. The dash blue lines are the volatilities of the HMM states.

time series (black)[1]. It is compared to the realized volatility annualized from SPX2.rv daily variance time series (red). The red line is the daily data and the red dots are the 5-day moving average. The reason for moving average is that the daily realized volatility is much more volatile. We can clearly see that $V(t)$ is a predictive analytics for realized volatility. It matches the moving average of Oxford-Man data reasonably well. Thus we find a very useful application for high-state HHM - It has the capability to figure out the realized volatility of a financial time series without having to calculate it from intra-day high-frequency data.

One can also narrow down the date range of the plot by specifying the `start.date` and `end.date` to a shorter, more recent date range. If `vix.adj.ratio` is specified (it is default to `NULL`), the VIX index will be plotted with the adjustment. One can study how the three volatility gauges change on a daily basis. An example is shown in Figure 6.3. We observe that HMM moves between the trends of the realized volatility and VIX, but it has its opinion. For instance, in late 2015, it moves above the other two gauges. But in late 2016, it is sided with the extremely low realized volatility in Oxford data.

6.2. *Volatility Forecasting*

A simple method to forecast volatility for next time period is proposed here. We use the historical observations $\boldsymbol{x}^{(T_0)}$ to fit HMM and obtain a stationary result $\boldsymbol{\pi} = \{\boldsymbol{\theta}, \boldsymbol{\Gamma}, \boldsymbol{\delta}\}$. Assume the historical

[1]Note that one can use one set of observations $\boldsymbol{x}^{(T)}$ to train HMM and obtain a stationary result $\boldsymbol{\pi} = \{\boldsymbol{\theta}, \boldsymbol{\Gamma}, \boldsymbol{\delta}\}$. Then use $\boldsymbol{\pi}$ to decode another time series $\boldsymbol{x}'^{(T')}$.

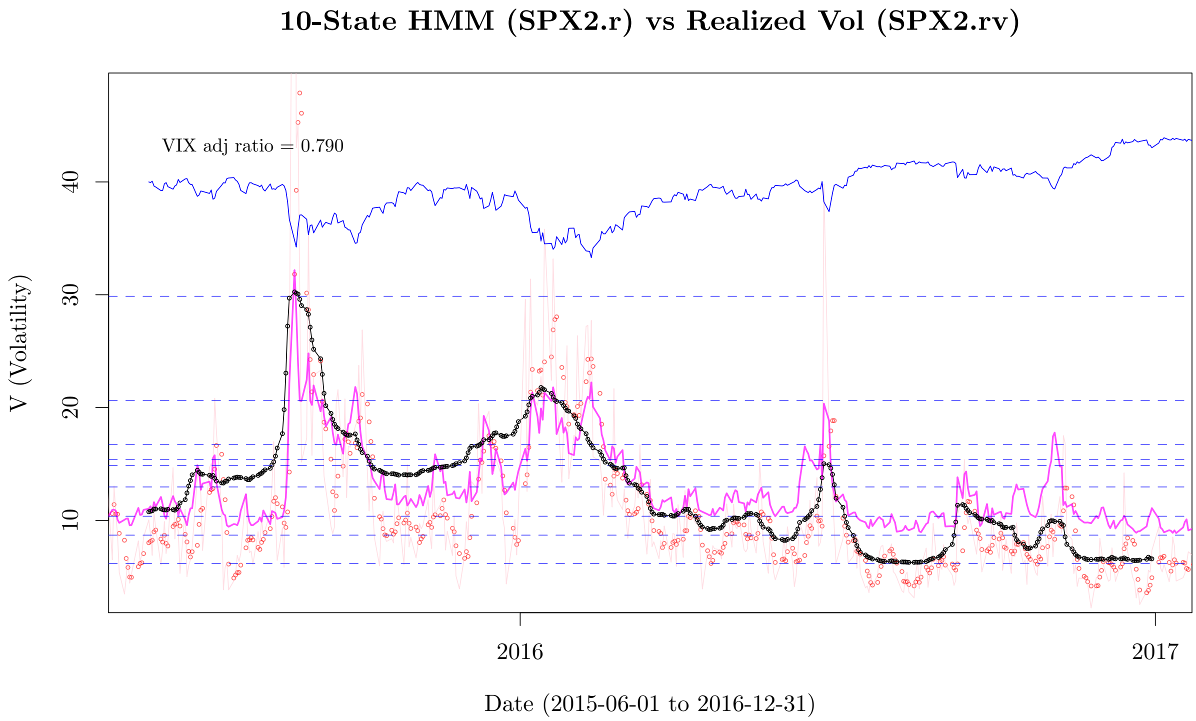

Figure 6.3. The 18-month range plot of the expected volatility $V(t)$ from ten-state HMM (black) vs the volatility from Oxford-Man realized variance data set (red dots) and the adjusted VIX index (magenta line). This plot illustrates the usage of `start.date` and `end.date` to specify a shorter, more recent date range in order to study how the three volatility gauges change on a daily basis. The VIX adjustment ratio is 0.79.

observations up to the most recent closes is $\boldsymbol{x}^{(T)}$, and next-day's observation will be x_{t+1}. We construct a new observation set $\boldsymbol{x}'^{(T+1)} = \left\{\boldsymbol{x}^{(T)}, x_{t+1}\right\}$. Then use $\boldsymbol{\pi}$ to decode $\boldsymbol{x}'^{(T+1)}$ and calculate the expected volatility $V\left(t+1; x_{t+1}\right)$ which is conditional on the future observation x_{t+1}. By providing different inputs of x_{t+1}, e.g. log-returns from -2% to 2% by 0.5%, one can forecast the volatility for a range of next-day moves. This method is implemented in `ldhmm.forecast_volatility` function:

```
> spx2 <- ldhmm.oxford_man_ts("SPX2.r", log=TRUE)
> xf <- seq(-0.02, 0.02, by=0.005) # to be forecasted
> ldhmm.forecast_volatility(hd, spx2$x, xf)
         [,1]     [,2]      [,3]      [,4]     [,5]     [,6]
[1,] -0.02000 -0.01500 -0.010000 -0.005000 0.000000 0.005000
[2,] 12.07079 10.72948  9.128792  7.500931 6.808746 6.916728
        [,13]    [,14]    [,15]
[1,] 0.010000 0.015000  0.02000
[2,] 7.762019 9.496151 12.07038
```

The reader should be noted that this is different from the typical HMM forecast, where the forecast involves predicting the most likely distributions of states and observations for next few periods. That is forecasting the market's direction. But here we are forecasting the most likely volatility based on a hypothetical market movement scenario.

6.3. *The Inverse Relation between Volatility and Return*

It is commonly known that the return of S&P 500 is inversely correlated to the volatility. This can be explained by the high anti-correlation of the daily changes between S&P 500 and the VIX index (See Figure 2 of Papanicolaou and Sircar (2014)). Here we attempt to use our HMM results to formulate this empirical law from a different angle.

The reader may have observed from the inserts of 2, 3, 5, 10 states that the relations between V_i and R_i are quite linear. The slopes are negative, that is, lower volatility is associated with positive return, while higher volatility with negative expected return. We can formulate this linear relation as

$$R_i = Y\,(V_0 - V_i)\,,\ \forall\, i, \tag{6.1}$$

where Y is called the "yield" of volatility differential, and V_0 is called the "pivotal realized volatility". By plotting the results from 2-state, 3-state, 4-state, 5-state, and 10-state together, as shown in Figure 6.4, we can calculate the linear regression and obtain

$$\begin{aligned} V_0 &\approx 17.4, \\ V_0(VIX) &\approx 17.4/0.79 = 22.0, \\ Y &\approx 2.2\%. \end{aligned} \tag{6.2}$$

$V_0(VIX)$ is the pivotal volatility adjusted to the VIX scale (ref Figure 6.1). The meaning of V_0 is that when the realized volatility is below V_0 , it is very likely S&P 500 will go up and continue in its bull market. However, if the realized volatility is above V_0 persistently, the market will go down and enter into a bear market. The level of VIX=22 as the turning point between bull markets and bear markets coincides very well with the common understanding. In fact, this level is commonly believed to be the long-term equilibrium in the VIX term structure[1].

The meaning of Y is that, for every point of volatility increase, S&P 500 will lose about 2.2% of annual return on average. For example, in the long run, if we expect S&P 500 to increase about 10% per year, its volatility must be below $V_0 - 10/2.2 \approx 12.9$. If we expect S&P 500 to increase about 20% per year, its volatility must be below $V_0 - 20/2.2 \approx 8.3$, which is very low. Thus Y and V_0 are handy numbers to calculation market expectation.

7. Acknowledgement

I would like to thank Professor John Mulvey for his guidance and discussions that lead to this work.

References

Geum Il Bae, Woo Chang Kim, and John M. Mulvey (2014). *Dynamic asset allocation for varied financial markets under regime switching framework.* European Journal of Operational Research, Vol. 234, No. 2, pp. 450-458.

Lihn, Stephen H.-T. (2015). *On the Elliptic Distribution and Its Application to Leptokurtic Financial Data.* SSRN: 2697911.

Lihn, Stephen H.-T. (2015). *The Special Elliptic Option Pricing Model and Volatility Smile.* SSRN: 2707810.

Lihn, Stephen H.-T. (2016). *Closed Form Solution and Term Structure for SPX Options.* SSRN: 2805769.

Lihn, Stephen H.-T. (2017). *From Volatility Smile to Risk Neutral Probability and Closed Form Solution of Local Volatility Function.* SSRN: 2906522.

Lihn, Stephen H.-T. (2017). *Parsimonious Pricing Model for VIX Options and Fixed Point ATM Volatility Hypothesis.* Not published yet.

[1]This number can be calculated as following. Take all the daily settlement prices of VIX futures from 03/2004 to 3/2017. For each trading day, record the price of the longest maturity contract. Then take the average, the result is 22.3.

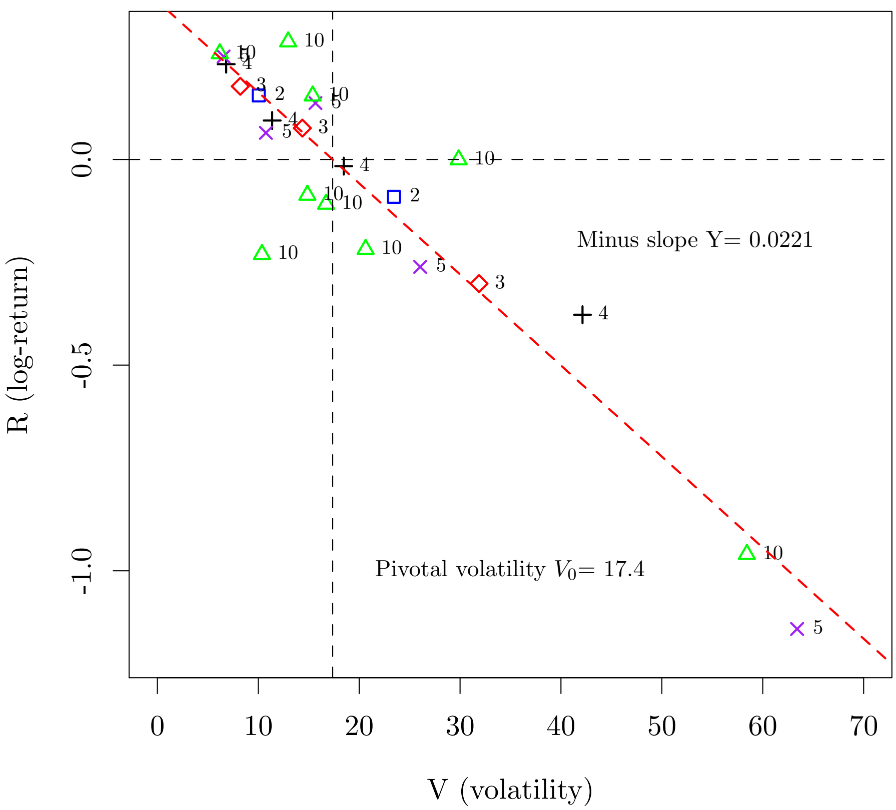

Figure 6.4. The inverse relation between the volatility and return in HMM. The pivotal volatility, $V_0 \approx 17.4$, is the dividing line between the bull market and the bear market. The minus slope, $Y \approx 2.2\%$, is called the "yield" of volatility differential. The fit is obtained by linear regression on all the results from 2-state, 3-state, 4-state, 5-state, and 10-state.

John M. Mulvey and Han Liu (2016). *Identifying Economic Regimes: Reducing Downside Risks for University Endowments and Foundations.* The Journal of Portfolio Management, Fall 2016, Vol. 43, No. 1: pp. 100-108.

Realized Library, Oxford-Man Institute of Quantitative Finance. This Library contains a selection of daily non-parametric estimates of volatility of financial assets. URL: http://realized.oxford-man.ox.ac.uk/

Neil Shephard and Kevin Shephard (2009). *Realising The Future: Forecasting With High Frequency Based Volatility (HEAVY) Models.* Department of Economics, Oxford University. Discussion Paper Series, 438.

Andrew Papanicolaou and Ronnie Sircar (2014). *A Regime-Switching Heston Model for VIX and S&P 500 Implied Volatilities* Quantitative Finance, Volume 14, Issue 10, pp. 1811-1827. SSRN: 2164500.

Walter Zucchini, Iain L. MacDonald, and Roland Langrock, *Hidden Markov Models for Time Series, An Introduction Using R.* Second Edition. CRC Press. ISBN: 9781482253832.