pvar: the calculation and applications of 
\textit{p}-variation

Vygantas Butkus
Vygantas.Butkus@gmail.com
August 6, 2014

Abstract
In this vignette we will give a brief introduction to the \textit{p}-variations. Later on, we will give some illustrative examples of the main functions in \textit{pvar} package.

1 Introduction
Let start with the definitions. Originally \textit{p}-variation is defined for a functions. For a function $f : [0, 1] \rightarrow \mathbb{R}$ and $0 < p < \infty$ \textit{p}-variation is defined as

$$v_p(f) = \sup \left\{ \sum_{i=1}^{m} |f(t_i) - f(t_{i-1})|^p : 0 = t_0 < t_1 < \ldots < t_m = 1, m \geq 1 \right\}.$$

Suppose $X = \{X_i\}_{i=0}^{n}$ is any real-value sequence of numbers. We will call such sequence a \textit{sample}, whereas $n$ will be referred to as a \textit{sample size}. The \textit{p}-variation of the sample could be expressed as

$$v_p(X) = \max \left\{ \sum_{i=1}^{k} |X_{j_i} - X_{j_{i-1}}|^p : 0 = j_0 < \ldots < j_k = n, k = 1, \ldots, n \right\}.$$

The set of index $j_0, \ldots, j_k$ that achieves the maximum is refereed as supreme partition.

Lots and detailed information on the known properties of \textit{p}-variation can be found in the books of Norvaiša and Dudley (see [1] and [2]). There are formulated and proved a lot of the properties of \textit{p}-variation. In addition, the comprehensive bibliography of \textit{p}-variation is include.

The core functions of the package is based on algorithm described in [?]. The calculation of \textit{p}-variation was inspired by paper [3], where authors Norvaiša and Račkauskas presented necessary and sufficient conditions for the convergence in law of partial sum processes in \textit{p}-variation norm. Based on the result of this paper, Račkauskas proposed the method of testing sample of multiple shift in mean. This test is also implemented in this package.
It seems that up till pvar package there were no software for $p$-variation calculus (if I am wrong, please, give me a note). The package pvar is here to fill that hole. The algorithm is written in C++, therefore it is quite fast. It is worth noting, that pvar package could be used to calculate $p$-variation for any piecewise monotone functions and to investigate general properties of $p$-variation.

2 Main function

The main function in pvar package is pvar (that a surprise). It calculates the supreme partition and the value of $p$-variation. The main usage of pvar function goes as follows

```r
N <- 1000
p <- 2
x <- rwiener(N)
pv <- pvar(x, p)
pv
## p-variation
## 5.63
summary(pv)
## The summary of p-variation:
## Value: 5.63, p = 2
## Data: x, n = 1000 (+1)
## Data vector (n=1001): 0, 0.03589, 0.07105, 0.04352, 0.05018, 0.05238, ...
## Partition has 83 points: 1, 3, 20, 26, 35, 60, ...
```

More over, the data and supreme partition could be illustrated with plot function

```r
plot(pv)
```
The points in supreme partition (the red points) are actually used in $p$-variation. All the other points are redundant and could be dropped out without any effect to $p$-variation. This fact is illustrated in the example below.

```r
pv.PP = pvar(x[pv$partition], TimeLabel = time(x)[pv$partition], 2)
plot(pv.PP, main = "The same pvar without redundant points")
```
Moreover, $p$-variation on different intervals could be merged into unified sample. Let look at the example

```matlab
x = rwiener(1000)
pv1 = pvar(x[1:500], 2)
pv2 = pvar(x[500:length(x)], 2)
sum_pv1_pv2 = pv1 + pv2
```

The illustration of this looks like this
A. Račauskas has proposed a method of detection of multiple changes in a sequence of independent observations. This method is based on results presented in [3]. This method is realized in this package under the name of Pvar Break Test (or PBT for short). It is implemented in `PvarBreakTest` function.

Let us consider the example. Let construct the data $x$ with multiple shifts in mean.

```r
c<-
```
Original data, with several shifts in mean

From the original data, it is not obvious if there are any breaks in sample. To test whether there is a multiple break, we can apply `PvarBreakTest` test. The results are remarkably good. Look below.

```r
xtest <- PvarBreakTest(x)
xtest
```

```
# PvarBreakTest
#
# H0: no structural change
# Results: H0 is rejected at the confidence level of 0.05.
# Data: x, n=1000.
# The output of the test:
# Statistics Critical Value alpha ~p.value
# 24.0422  12.7693  0.0500  0.0073
```
Program suggests 3 break points.

References


