Exercise 9 - Self Organising Map applied to BTC (in R)

November 15, 2017

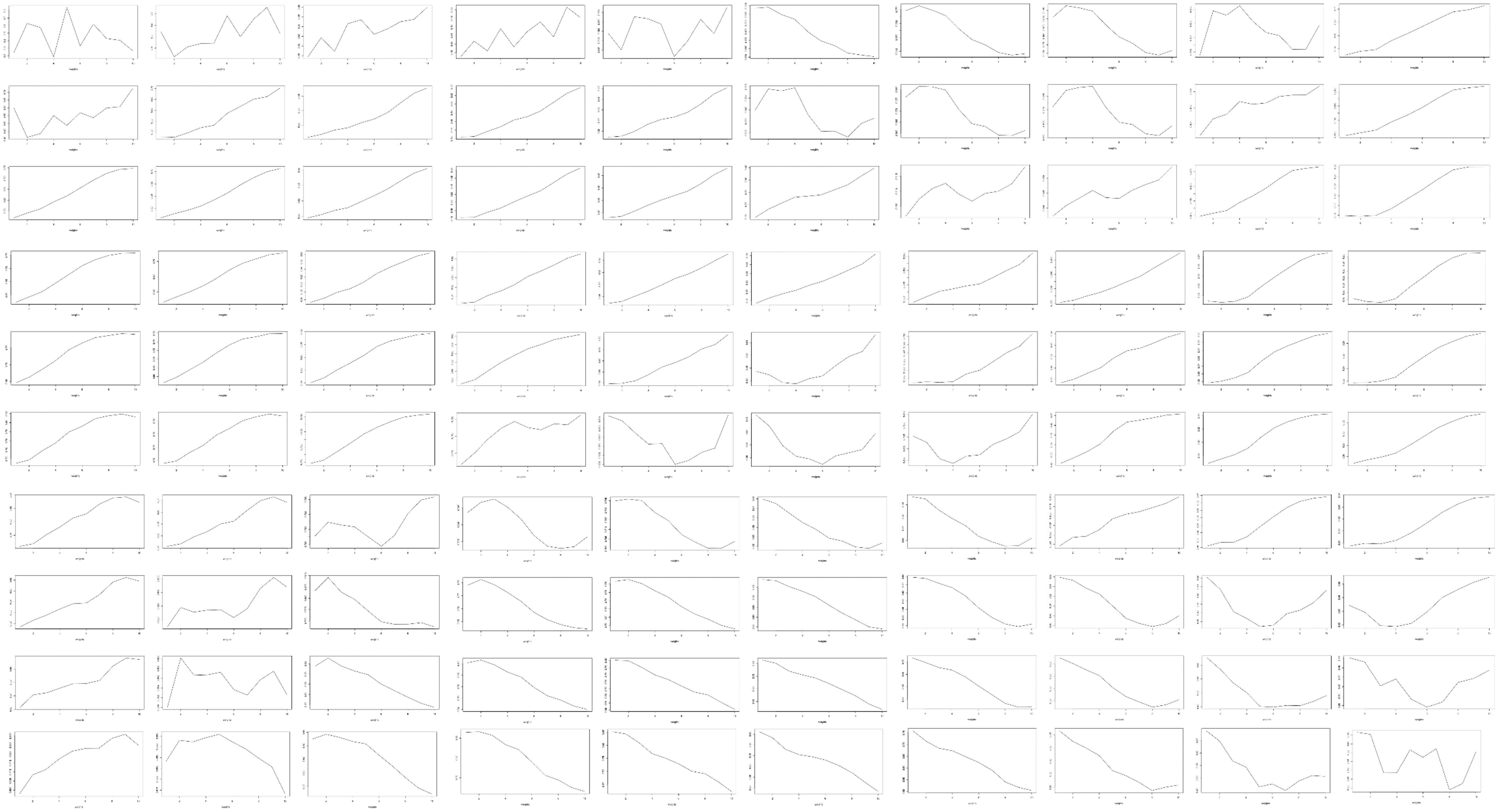

Figure 1 - Output map of weights from SOM applied to BTC

Performance: 0.54

Script:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

data <- read.csv("R/BTC.csv")
data_train <- data[1:300,2]

normalize <- function(x)
{
  return ((x - min(x)) / (max(x) - min(x)))
}

data_train <- normalize(data_train)

data_train_full <- matrix(0, 290, 10)

for(i in 1:290)
{
  for(k in 1:10)
  {
    data_train_full[i,k] <- data_train[i+k-1]
  }
}


t <- 1
d <- matrix(0, 10, 10)
pattern_size <- 10
map <- matrix(list(), 10, 10)
positionx <- integer(290)
positiony <- integer(290)
SOM <- matrix(0,2,290)
rownames(SOM) <- c("X", "Y")

#3. Set random map weights
for (i in 1:10)
{
  for (j in 1:10)
  {
    map[[i,j]] <- runif(10, 0, 1)
  }
}

#4. Pick animal at random
for (t in 1:10000)
{
  r <- round(runif(1, 1, 290))
  x <- data_train_full[r,]
  X <- t(x)

  for (i in 1:10)
  {
    for (j in 1:10)
    {
      w <- map[[i,j]]
      W <- t(w)
      d[i,j] <- sqrt((as.numeric(x)-as.numeric(W))%*%(as.numeric(X)-as.numeric(w)));
    }
  }
  distance <- d[1,1]

  for (i in 1:10)
  {
    for (j in 1:10)
    {
      if (d[i,j] < distance)
      {
        distance <- d[i,j]
        u <- i
        v <- j
      }
    }
  }
  #setting learning parameters
  theta <- 1;
  alpha <- 100/(200+t);
  for (i in 1:10)
  {
    for (j in 1:10)
    {
      eta <- exp(-(((i-u)^2)+(j-v)^2)/2*theta^2)
      map[[i,j]] <- map[[i,j]] + alpha*eta*(X-map[[i,j]])
    }
  }
}


#TESTING PHASE
for (test in 1:290)
{
  y <- data_train_full[test,]
  Y <- t(y)
  for (i in 1:10)
  {
    for (j in 1:10)
    {
      w <- map[[i,j]]
      W <- t(w)
      d[i,j] <- sqrt((as.numeric(y)-as.numeric(W))%*%(as.numeric(Y)-as.numeric(w)))
    }
  }
  distance <- d[1,1]
  for (i in 1:10)
  {
    for (j in 1:10)
    {
      if (d[i,j] < distance)
      {
        distance <- d[i,j]
        positionx[test] <- i
        positiony[test] <- j
      }
    }
  }
}

#Mapping each animal
for(i in 1:290)
{
  SOM[1,i] <- positionx[i]
  SOM[2,i] <- positiony[i]
}


##plot(t(SOM), col = "red")
par(mfrow = c(3,3))
for(row in 1:3)
{
  for(col in 1:3)
  {
    wts <- as.numeric(map[[row,col]])
    plot(wts, type = "l", xlab = "weights", ylab = "")
  }
}
```