

# Stochastic gradient decent

Yinan Zhu

September 27, 2017

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.4.1
```

```
data <- read_csv("~/GitHub/SDS385-course-work/Excercise 2/stochastic gradient decent/wdbc.csv",col_name
```

```
## Parsed with column specification:
```

```
## cols(
```

```
##   .default = col_double(),
```

```
##   X1 = col_integer(),
```

```
##   X2 = col_character()
```

```
## )
```

```
## See spec(...) for full column specifications.
```

```
source('~/.GitHub/SDS385-course-work/Excercise 2/stochastic gradient decent/gradient decent functions.R')
```

```
X=as.matrix(data[3:12])
```

```
X=scale(X)
```

```
X=cbind(X,1)
```

```
y=as.vector(matrix(nrow=nrow(data),ncol=1))
```

```
for(i in 1:nrow(data)){
```

```
  if(data[i,2]=="M")y[i]=1
```

```
  else y[i]=0
```

```
}
```

```
beta0=as.vector(matrix(0,nrow=11))
```

```
trainX=X[1:250,]
```

```
trainy=y[1:250]
```

```
testX=X[251:569,]
```

```
testy=y[251:569]
```

```
ite=1000
```

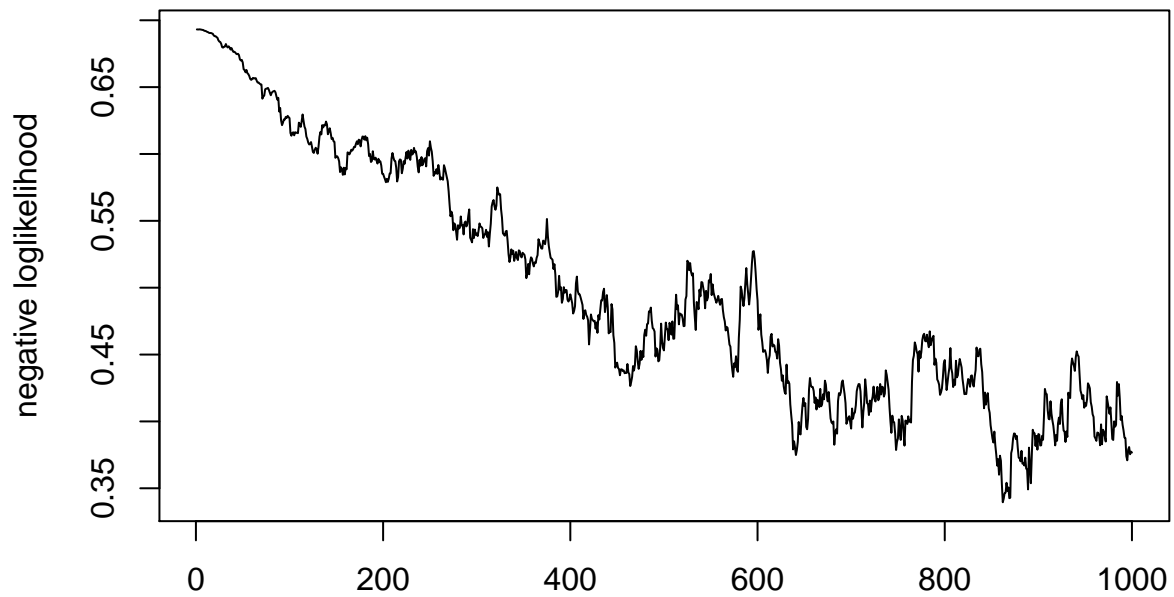
```
alpha=0.05
```

we use moving average with exponential decay to compute “averagenegloglikelihood”:  $\text{averageX}(t) = X(t)\alpha + \text{averageX}(t-1)(1-\alpha)$

```
eps=0.001
```

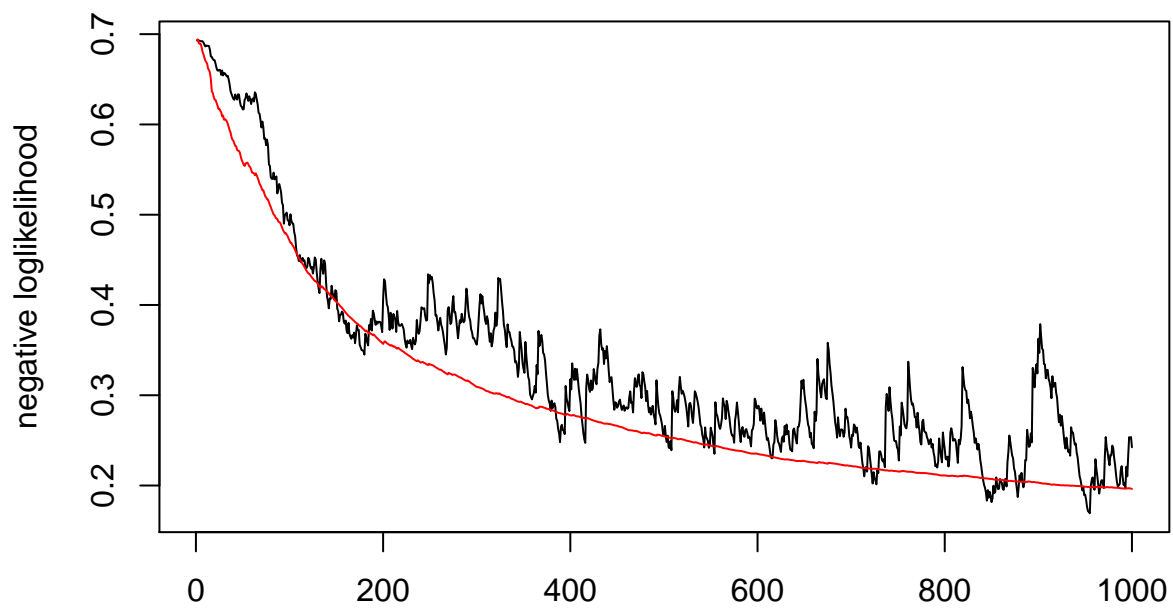
```
result=stochasticgradientdecent(trainX,trainy,testX,testy,beta0,eps,ite,alpha)
```

```
plot(result$averagenegloglikelihood,type='l',xlab='',ylab='negative loglikelihood',sub='eps=0.001')
```



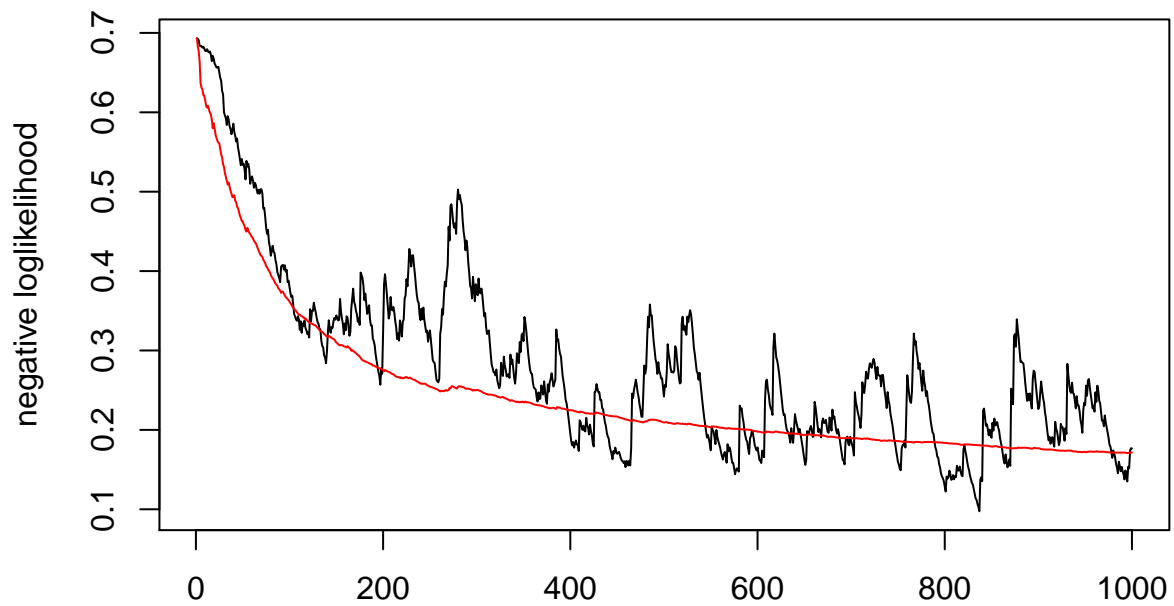
eps=0.001

```
eps=0.005
result=stochasticgradientdecent(trainX,trainy,testX,testy,beta0,eps,ite,alpha)
plot(result$averagenegloglikelihood,type='l',xlab='',ylab='negative loglikelihood',sub='eps=0.005')
lines(result$testnegloglikelihood,col='red')
```



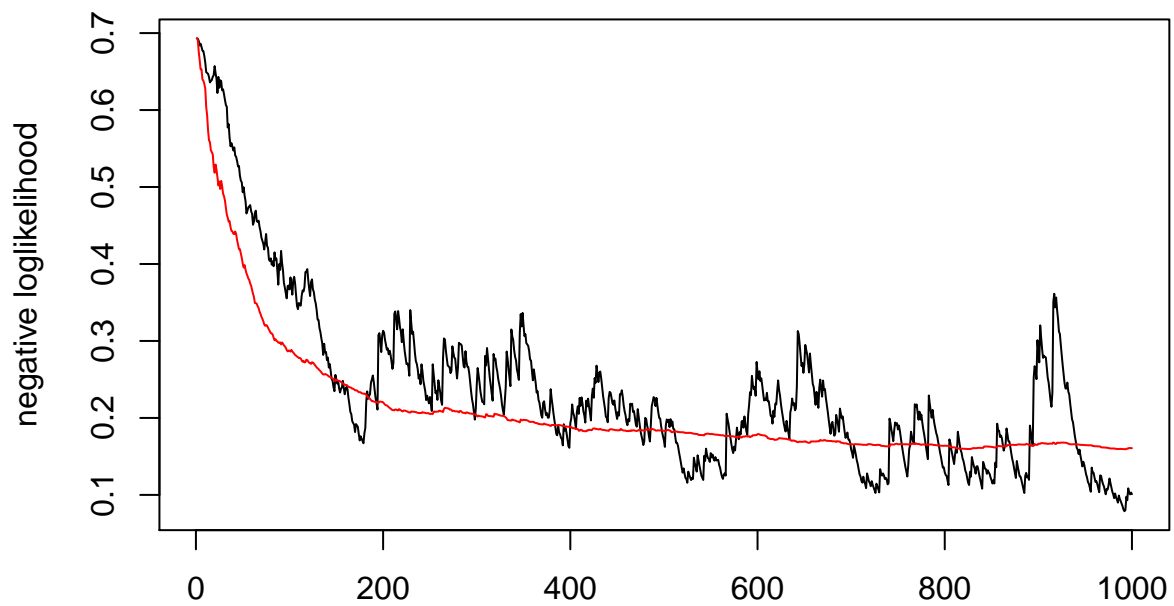
eps=0.005

```
eps=0.01
result=stochasticgradientdecent(trainX,trainy,testX,testy,beta0,eps,ite,alpha)
plot(result$averagenegloglikelihood,type='l',xlab='',ylab='negative loglikelihood',sub='eps=0.01')
lines(result$testnegloglikelihood,col='red')
```



eps=0.01

```
eps=0.02
result=stochasticgradientdecent(trainX,trainy,testX,testy,beta0,eps,ite,alpha)
plot(result$averagenegloglikelihood,type='l',xlab='',ylab='negative loglikelihood',sub='eps=0.02')
lines(result$testnegloglikelihood,col='red')
```



eps=0.02

varying steps, we use the Robbins-Monro rule where (step size at  $t$ ) =  $C/(t+t_0)^{\text{decay}}$

`t0=1`

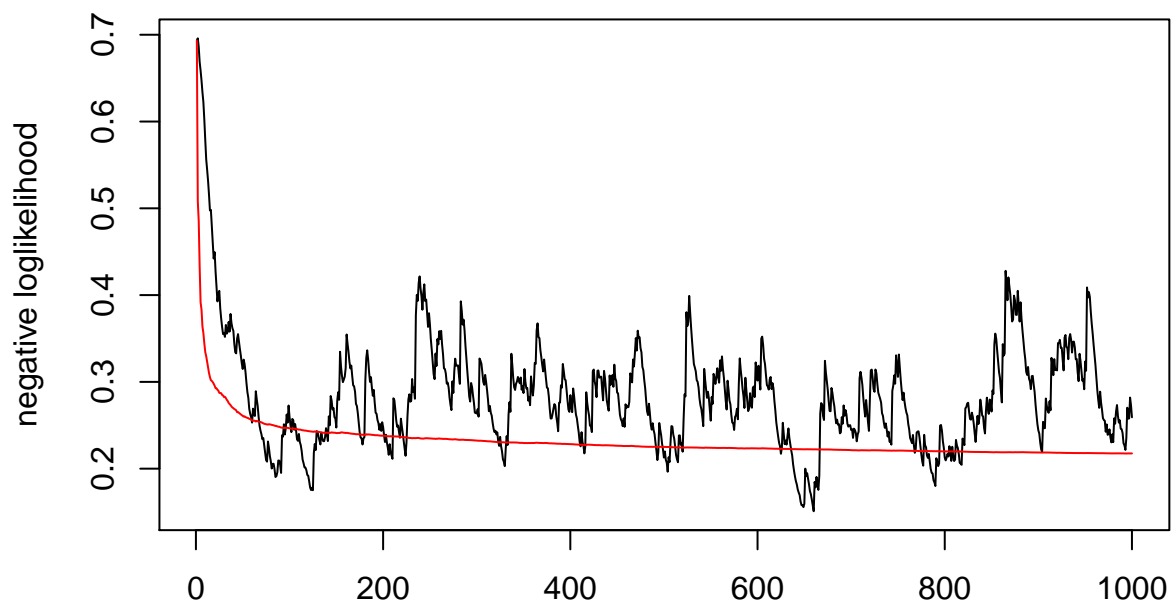
`C=0.5`

`decay=0.95`

`result=varyingstepsgradientdecent(trainX,trainy,testX,testy,beta0,ite,alpha,decay,t0,C)`

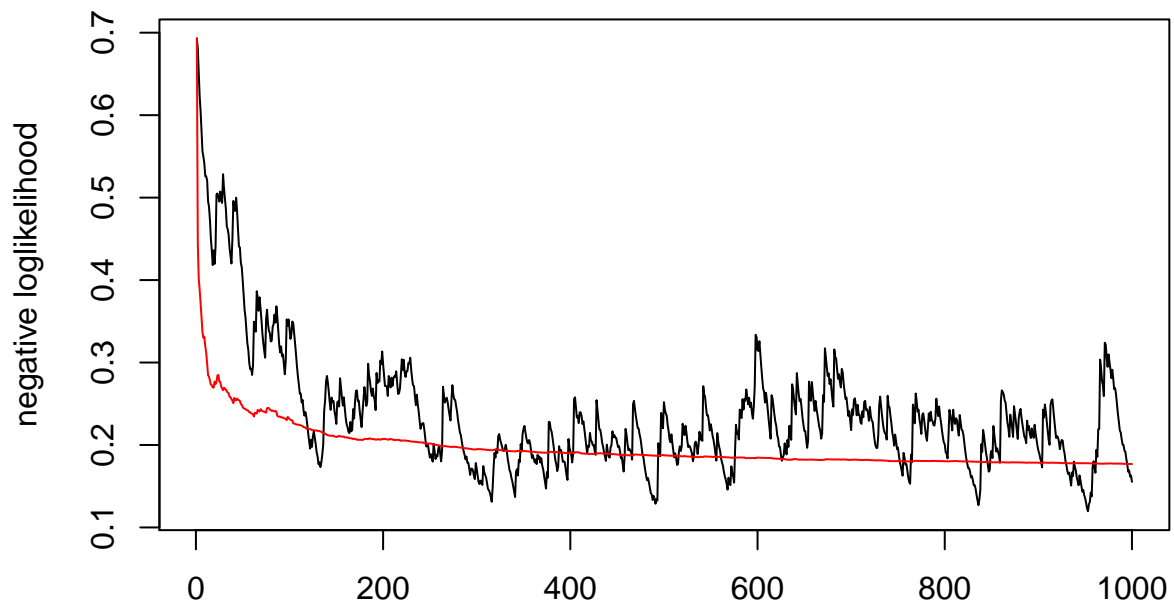
`plot(result$averagenegloglikelihood,type='l',ylab='negative loglikelihood',xlab='',sub='decay=0.95')`

`lines(result$testnegloglikelihood,col='red')`



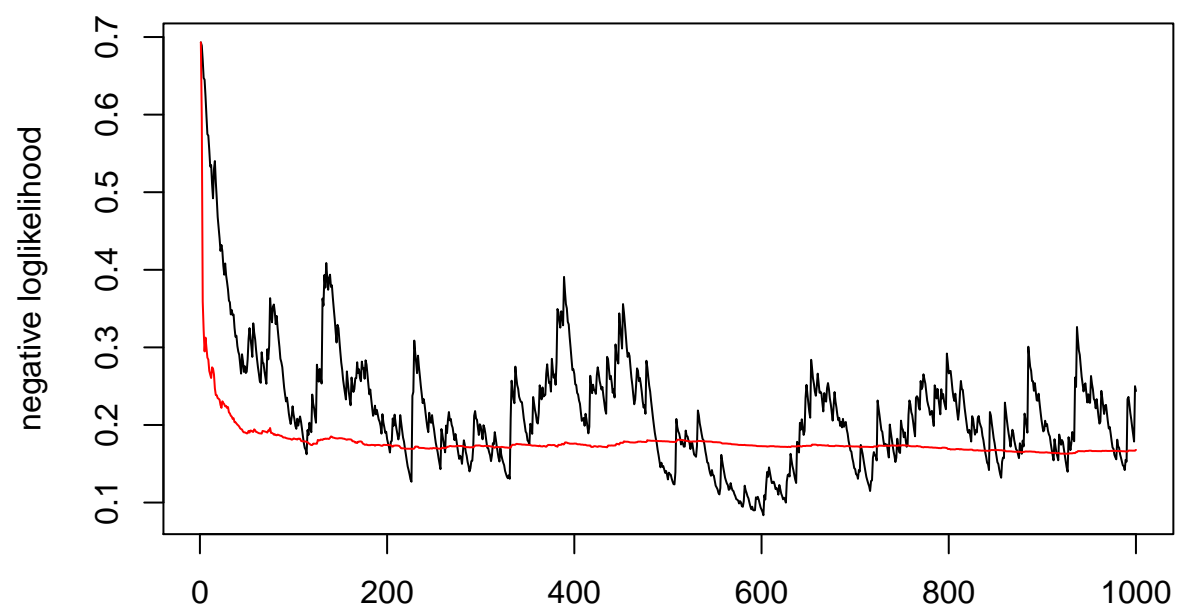
decay=0.95

```
decay=0.75
result=varyingstepsgradientdecent(trainX,trainy,testX,testy,beta0,ite,alpha,decay,t0,C)
plot(result$averagenegloglikelihood,type='l',ylab='negative loglikelihood',xlab='',sub='decay=0.75')
lines(result$testnegloglikelihood,col='red')
```



decay=0.75

```
decay=0.6
result=varyingstepsgradientdecent(trainX,trainy,testX,testy,beta0,ite,alpha,decay,t0,C)
plot(result$averagenegloglikelihood,type='l',ylab='negative loglikelihood',xlab='',sub='decay=0.6')
lines(result$testnegloglikelihood,col='red')
```



decay=0.6