

# 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 1/gradient decent/wdbc.csv",col_names = FALSE)

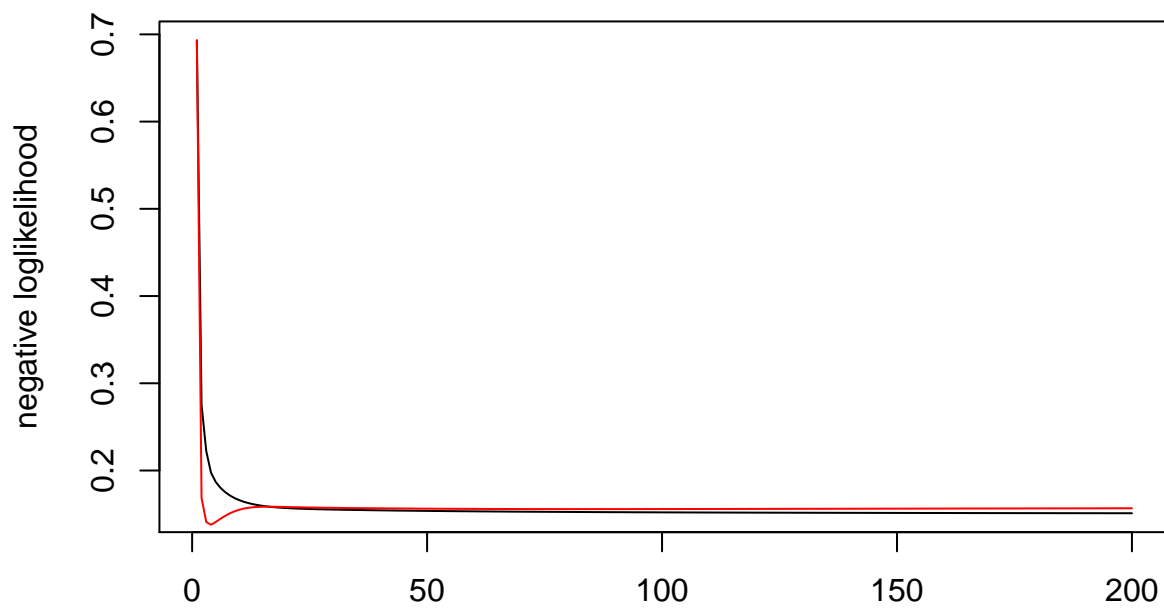
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_integer(),
##   X2 = col_character()
## )

## See spec(...) for full column specifications.
source('~/.GitHub/SDS385-course-work/Excercise 1/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]

black line is the likelihood for traning data, red for test data

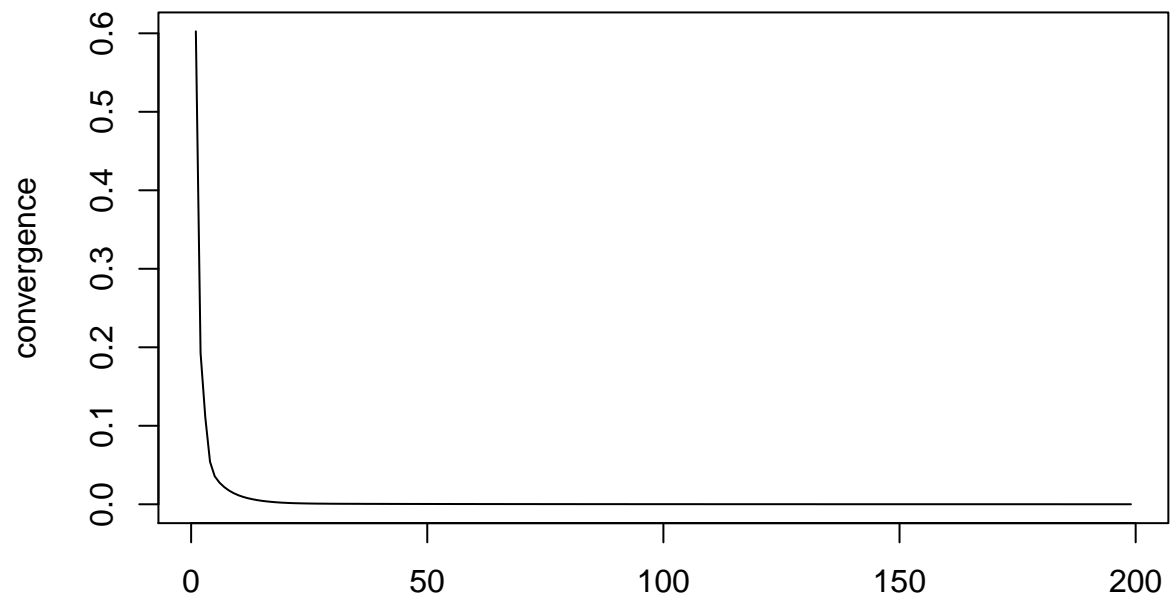
ite=200
eps=0.02
result=gradientdecent(trainX,trainy,testX,testy,beta0,eps,ite)
plot(result$negloglikelihood,type='l',ylab='negative loglikelihood',xlab='',sub='eps=0.02')
lines(result$testnegloglikelihood,col='red')
```



eps=0.02

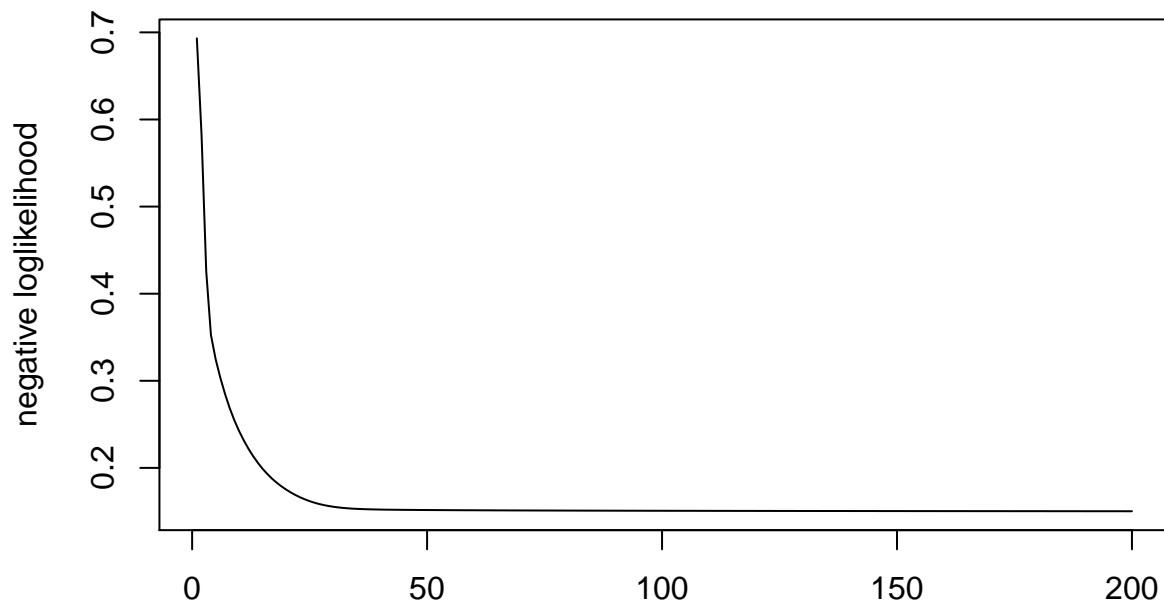
we can use percentage change of likelihood as a measure of convergence

```
convergence=abs(result$negloglikelihood[2:ite]-result$negloglikelihood[1:(ite-1)])/(result$negloglikelihood[1:(ite-1)])
plot(convergence,type='l',xlab='',ylab='convergence',sub='eps=0.02')
```



eps=0.02

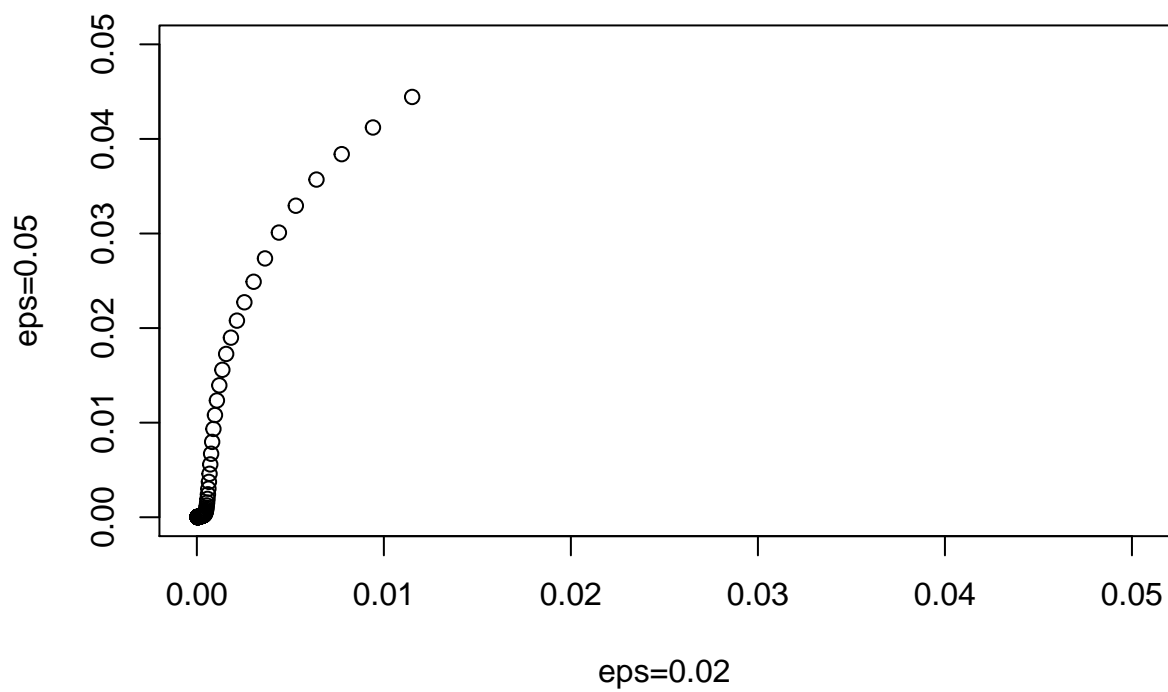
```
eps=0.05  
result2=gradientdescent(trainX,trainy,testX,testy,beta0,eps,ite)  
plot(result2$negloglikelihood,type='l',ylab='negative loglikelihood',xlab='',sub='eps=0.05')
```



eps=0.05

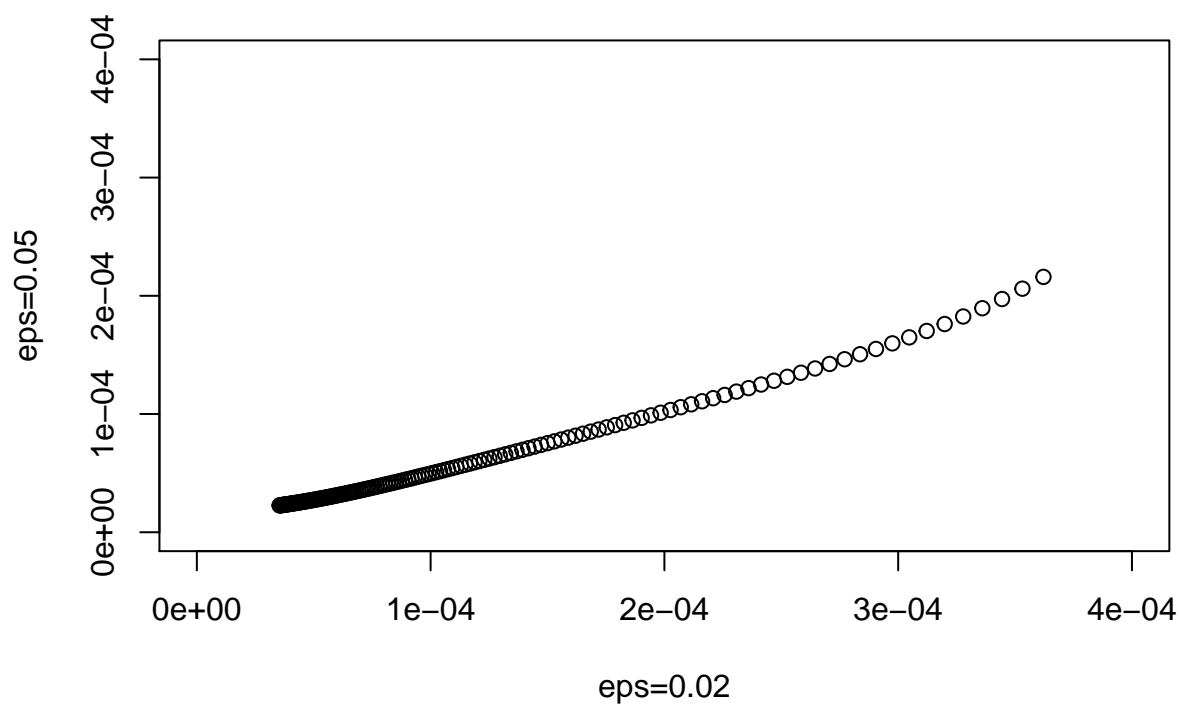
compare the convergence performace of gd with two different step sizes. At the start of the decent, longer step size result better convergence

```
convergence2=abs(result2$negloglikelihood[2:ite]-result2$negloglikelihood[1:(ite-1)])/(result2$neglogli
plot(convergence[10:ite],convergence2[10:ite],xlab ="eps=0.02",ylab="eps=0.05",xlim=c(0,0.05),ylim=c(0,
```



However, if we zoom into the later stages, smaller step size is better

```
plot(convergence[50:ite],convergence2[50:ite],xlab ="eps=0.02",ylab="eps=0.05",xlim=c(0,0.0004),ylim=c(
```



Newton method, enjoy the speed of convergence

```
ite=20
result=newtonmethod(trainX,trainy,testX,testy,beta0,ite)
plot(result$negloglikelihood,type='l',ylab='negative loglikelihood',xlab='',sub='newton method')
lines(result$testnegloglikelihood,col='red')
```

