Sta 250 Homework 1

Chen Zihao 915490404

Problem 1. Convex Sets and Convex Functions

Prove whether the following sets of functions are convex or not

(a) $\{x \in \mathbb{R}^n | Ax = b\}$ where $A \in \mathbb{R}^{mxn}$,$b \in \mathbb{R}^m$

Pf:

$\forall x_1, x_2 \in \{x \in \mathbb{R}^n | Ax = b\}$,$\forall \alpha \in [0, 1]$,

$$A(\alpha x_1 + (1-\alpha)x_2) = \alpha A x_1 + (1-\alpha)Ax_2 = \alpha b + (1-\alpha)b = b$$

which means $\alpha x_1 + (1-\alpha)x_2 \in \{x \in \mathbb{R}^n | Ax = b\}$

So that {$x \in R^n | Ax = b$ where $A \in \mathbb{R}^{mxn}$,$b \in \mathbb{R}^m$} is a convex sets.

(b){$x \in \mathbb{R}^n | ||x - x_0||_2 = r$},where $x_0 \in \mathbb{R}^n, r \in \mathbb{R}$

Pf:

let $x_0 = [0, 0]^T, r = \sqrt{5}, x_1 = [1, 2]^T, x_2 = [2, 1]^T$, so that $x_1, x_2 \in \{x \in \mathbb{R}^n | ||x||_2 = \sqrt{5}\}$,
$\forall \alpha \in (0, 1)$,

$\alpha x_1 + (1-\alpha)x_2 = [2-\alpha, 1+\alpha]^T$,

$$||\alpha x_1 + (1-\alpha)x_2||_2 = [(2-\alpha)^2 + (1+\alpha)]^{\frac{1}{2}} = (2\alpha^2 - 2\alpha + 5)^{\frac{1}{2}} \neq \sqrt{5}$$

So that {$x \in \mathbb{R}^n | ||x - x_0||_2 = r$,where $x_0 \in \mathbb{R}^n, r \in \mathbb{R}$}is not a convex sets.

(c) $f(x_1, x_2) = (x_1 x_2 - 1)^2$, where $x_1, x_2 \in \mathbb{R}$

Pf:

Let consider $(x_1, x_2) = (1, 1), (y_1, y_2) = (0, 0) \in \mathbb{R}^2$,$\alpha = 0.1 \in (0, 1)$

$$f(\alpha x_1 + (1-\alpha)y_1, \alpha x_2 + (1-\alpha)y_2) = f(0.1, 0, 1) = (0.1^2 - 1)^2 = 0.99^2$$
$$\alpha f(x_1, x_2) + (1-\alpha)f(y_1, y_2) = \alpha f(1, 1) + (1-\alpha)f(0, 0) = 0.9$$

$0.99 * 0.99 > 0.9$, so that $\exists (x_1, x_2), (y_1, y_2) \in \mathbb{R}^2, \exists \alpha \in (0, 1)$,

$$\alpha f(x_1, x_2) + (1-\alpha)f(y_1, y_2) < f(\alpha x_1 + (1-\alpha)y_1, \alpha x_2 + (1-\alpha)y_2)$$

So that it is not a convex function.

(d)$f(w_1, w_2) = ||w_1 - w_2||_2^2$, where $w_1, w_2 \in \mathbb{R}^2$

Pf:

$\forall x_1, x_2, y_1, y_2 \in \mathbb{R}^2$,$\forall \alpha \in [0,1]$,

$$\begin{aligned}
\alpha f(x_1, x_2) + (1-\alpha) f(y_1, y_2) &= \alpha ||x_1 - x_2||_2^2 + (1-\alpha)||y_1 - y_2||_2^2 \\
&= \sum_{i=1}^{2} [\alpha (x_{1i} - x_{2i})^2 + (1-\alpha)(y_{1i} - y_{2i})^2] \\
f(\alpha x_1 + (1-\alpha) y_1, \alpha x_2 + (1-\alpha) y_2)) &= \sum_{i=1}^{2} [\alpha (x_{1i} - x_{2i}) + (1-\alpha)(y_{1i} - y_{2i})]^2
\end{aligned}$$

Note $X_i = x_{1i} - x_{2i}, Y_i = y_{1i} - y_{2i}$

All I need to prove is that $\alpha X_i^2 + (1-\alpha) Y_i^2 \geq [\alpha X_i + (1-\alpha) Y_i]^2$

It is the same problem as proving $f(x) = x^2$ is a convex function.

$f(x) = x^2$ is a convex function

$\Rightarrow \alpha f(x_1, x_2) + (1-\alpha) f(y_1, y_2) \geq f(\alpha x_1 + (1-\alpha) y_1, \alpha x_2 + (1-\alpha) y_2))$

$\Rightarrow f(w_1, w_2)$is a convex function

Problem 2. Stationary points

(a) Identify stationary points for $f(x) = 2x_1 + 12x_2 + x_1^2 - 3x_2^2$? Are they local minimum/maximum; global minimum/maximum or saddle points? Why?

Answer:

$$\frac{\partial f(x)}{\partial x_1} = 2 + 2x_1$$

$$\frac{\partial f(x)}{\partial x_2} = 12 - 6x_2$$

let$\triangledown f(x) = 0$,we get (-1,2),

$$\triangledown^2 f(x) = \begin{bmatrix} 2 & 0 \\ 0 & -6 \end{bmatrix}$$

the stationary point is (-1,2), it is a saddle point.

(b)Assume $f : \mathbb{R}^n \to \mathbb{R}$ is strongly convex and is L-Lipchitz($||\triangledown f(x) - \triangledown f(y)||_2 \leq L||x - y||_2$) for any (x,y). Given an n by n symmetric matrix B with $MI \succeq B \succeq mI$ with $M \geq m > 0$, provide a valid step size η such that the sequence

$$x^{k+1} = x^k - \eta B \triangledown f(x^k)$$

converges to the minimizers of f.

The function is strongly covex and is L-Lipchitz $\Rightarrow$ all limit points are stationary points, all the stationary points are the global minimizers.

let $x^+ = x^{k+1}, x = x^k$

$$
\begin{aligned}
f(x^+) &\le f(x) + \triangledown f(x)^T(x^+ - x) + \frac{L}{2}||x^+ - x||^2 \\
&= f(x) + \triangledown f(x)^T(-\eta B \triangledown f(x)) + \frac{L}{2}|| - \eta B \triangledown f(x)||^2 \\
&= f(x) - \triangledown f(x)^T(\eta I - \frac{L\eta^2}{2} B^T) B \triangledown f(x)
\end{aligned}
$$

$\Rightarrow (\eta I - \frac{L\eta^2}{2} B^T)B$ should be a positive definite matrix

As $MI \succeq B \succeq mI$,$x \in [m, M]$

$$
\begin{aligned}
(\eta - \frac{L\eta^2}{2} x) x I &\succeq 0 \\
\Rightarrow (1 - \frac{L\eta}{2} x) &\ge 0 \\
\eta &\le \frac{2}{Lx} \\
\Rightarrow \eta &\le \frac{2}{LM}
\end{aligned}
$$

Problem 3. Gradient Descent

Given training data $\{x_i, y_i\}_{i=1}^n$, each $x_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$, we try to solve the following logistic regression problem by gradient descent:

$$
\min_{w \subset \mathbb{R}^d} \{ \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-y_i w^T x_i}) + \frac{1}{2}||w||_2^2 \} := f(w). \tag{1}
$$

Test the algorithm using the “heart_scale” datasetwith n = 270 and d = 13: the matrix X is stored in the file “X_heart”, and the vector y is stored in the file “y_heart”.

(a)

Implement the gradient descent algorithm with a fixed step size η. Find a small η_1 such that the algorithm converges. Increase the step size to η_2 so the algorithm cannot converge. Run 50 iterations and plot the iteration versus $log(f(x^k) - f(x^*))$ plot for η_1 and η_2. In practice it is impossible to get the exact optimal solution x^*, So use the minimum value you computed as $f(x^*)$ when you plot the figure. Report the $f(x^*)$ value you used for generating the plots.

```
#read data
x<-read.table('E:/hw1_data/X_heart')
y<-read.table('E:/hw1_data/y_heart')
#x<-read.table('E:/hw1_data/x_epsilonsubset')
#y<-read.table('E:/hw1_data/y_epsilonsubset')
#add a constant variables and put y in front
X<-as.matrix(cbind(y,1,x))
#the number of samples
n=nrow(X)
#the number of variables
p=ncol(X)-1
#iteration times
k=50
```

$$
\triangledown f(w) = \frac{1}{n} \sum_{i=1}^n \frac{-y_i e^{-y_i w^T x_i}}{1 + e^{-y_i w^T x_i}} x_i + W
$$

```r
#set f'(w) as a function
fw1<-function(w){
    #calculate f'(w)
    fw=rowSums(apply(X,1,function(X){
      e=exp(-X[1]*X[-1]%*%w)
     X[-1]*as.numeric((-X[1]*e)/(1+e))
    }))
    fw=fw/n+w
    fw
}
```

```r
#set f(w) as a function
fw<-function(w){
  f=1/2*sum(w^2)+1/n*sum(apply(X,1,function(X){log(1+exp(-X[1]*X[-1]%*%w))}))
  f
}
```

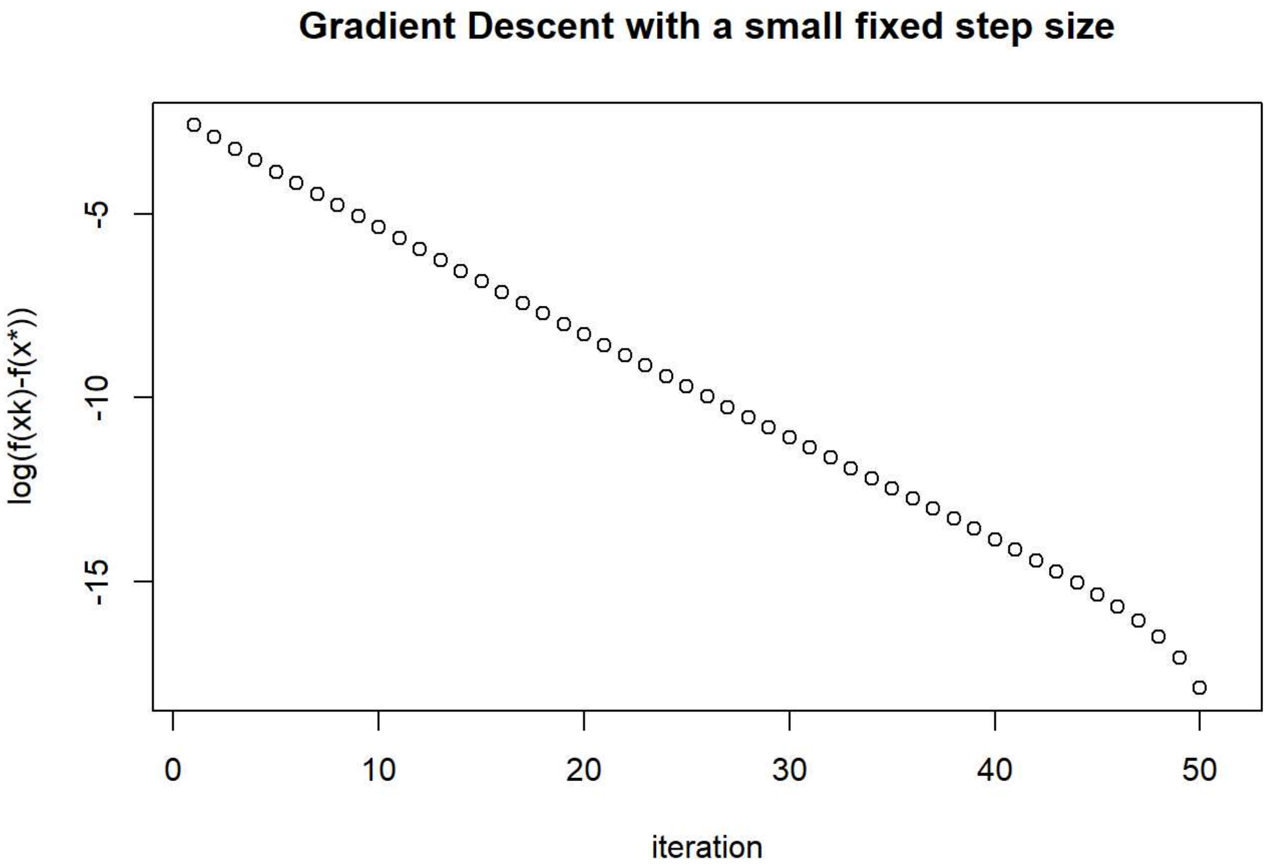
```r
#initial w
w=matrix(0, p, 1)
#To record w
wlist=w
#set eta
eta=0.1

for(j in 1:k){
  w=w-eta*fw1(w)#the new w
  wlist=cbind(wlist,w)#record w
}
```

```r
flist=0
for (j in 1:(k+1)) {
  flist=cbind(flist,fw(wlist[,j]))}
flist=flist[,2:(k+2)]
```

```r
min(flist)
```

```
## [1] 0.6184193
```

```r
plot(log(flist-min(flist)),xlab = "iteration",ylab = "log(f(xk)-f(x*))",main = "Gradient Descent with
 a small fixed step size")
```



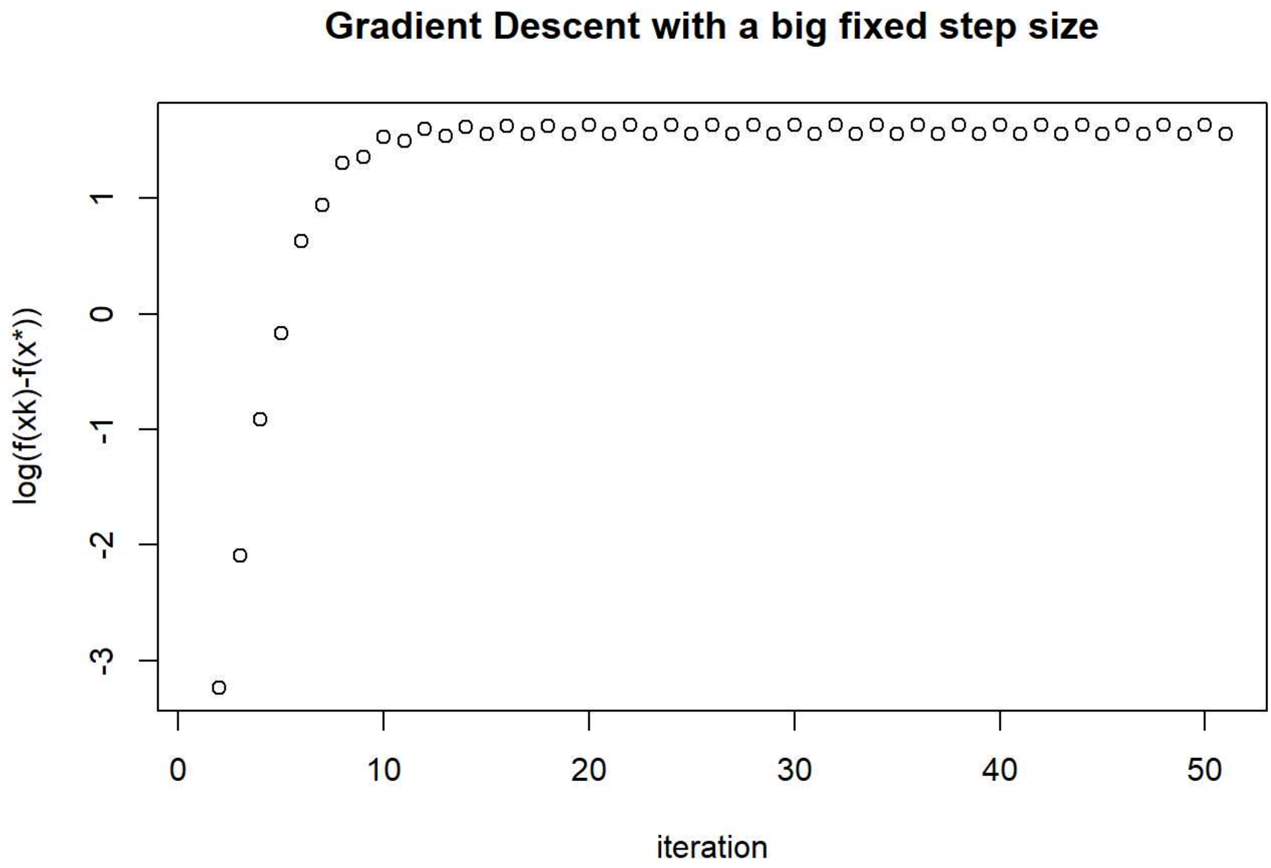

```
#initial w
w=matrix(0, p, 1)
#To record w
wlist=w
#set eta
eta=1.5

for(j in 1:k){
  w=w-eta*fw1(w)#the new w
  wlist=cbind(wlist,w)#record w
}
```

```
flist=0
for (j in 1:(k+1)) {
  flist=cbind(flist,fw(wlist[,j]))}
flist=flist[,2:(k+2)]
```

```
min(flist)
```

```
## [1] 0.6931472
```

```
plot(log(flist-min(flist)),xlab = "iteration",ylab = "log(f(xk)-f(x*))",main = "Gradient Descent with
 a big fixed step size")
```



(b)

Implement the gradient descent algorithm with backtracking line search. Plot the same iteration versus $log(f(x^k) - f(x^*))$

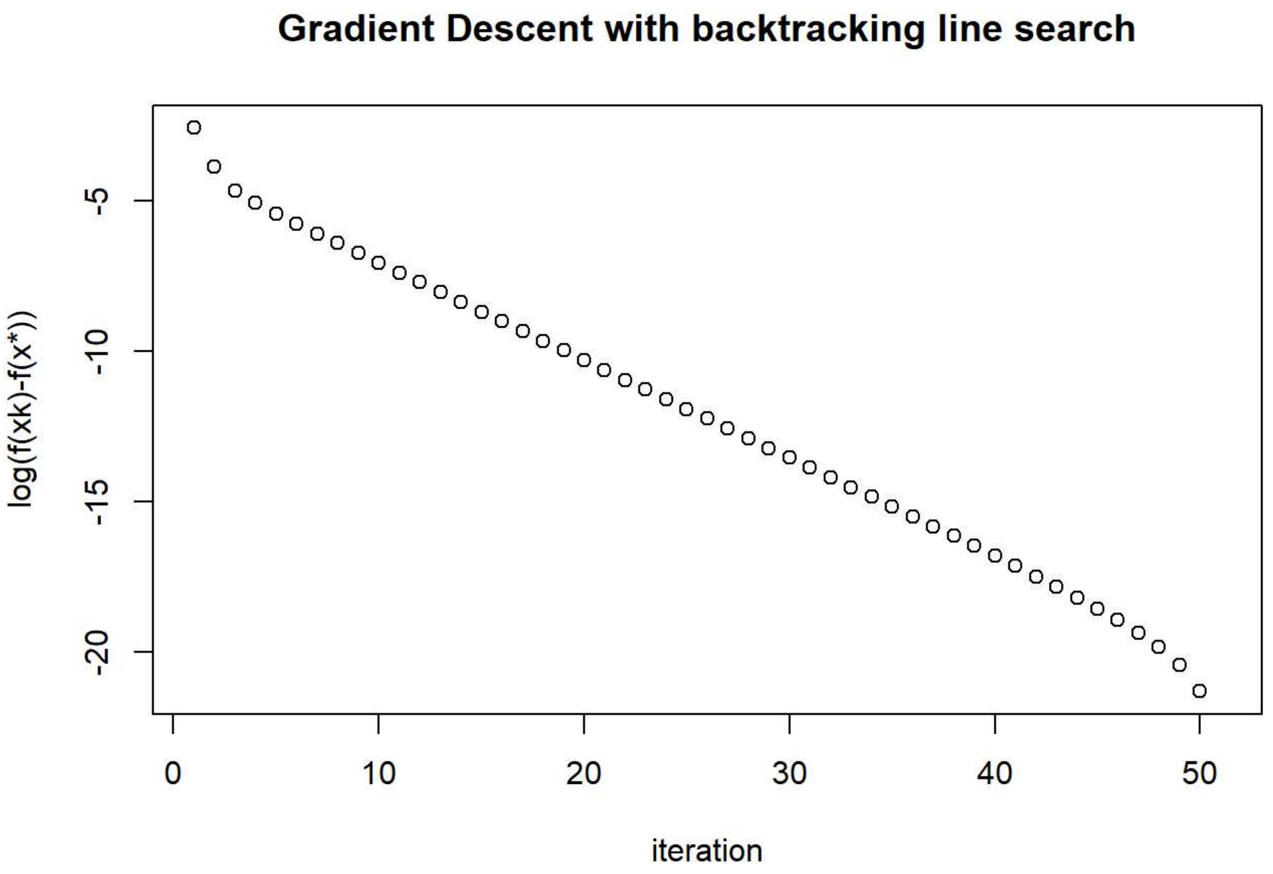
```
#initial w
w=matrix(0, p, 1)
#To record w
wlist=w

for(j in 1:k){
  g=fw1(w)
  eta=1
  while(fw(w-eta*g)-fw(w)>-0.01*eta*sum(g^2)){
    eta=eta/2
  }
  w=w-eta*g#the new w
  wlist=cbind(wlist,w)#record w
}
```

```
flist=0
for (j in 1:(k+1)) {
  flist=cbind(flist,fw(wlist[,j]))}
flist=flist[,2:(k+2)]
```

```
min(flist)
```

```
## [1] 0.6184192
```

```
plot(log(flist-min(flist)),xlab = "iteration",ylab = "log(f(xk)-f(x*))",main = "Gradient Descent with
 backtracking line search")
```



(c) larger data.

```
#read data
x<-read.table('E:/hw1_data/x_epsilonsubset')
y<-read.table('E:/hw1_data/y_epsilonsubset')
#add a constant variables and put y in front
X<-as.matrix(cbind(y,1,x))
#the number of samples
n=nrow(X)
#the number of variables
p=ncol(X)-1
#iteration times
k=11  # i find that after 9 times, it seems it is already around the limit point.But I still set the u
pper iteration times here to pretend i do not know it.
```

the functions are already in R

run the algorithm 2

```
#initial w
w=matrix(0, p, 1)
#To record w
wlist=matrix(0,p,(k+1))
wlist[,1]=w

for(j in 1:k){
  g=fw1(w)
  if (sum(g^2)<10^-15){break}#i think this is really small enough to give the conclusion
  eta=1
  while(fw(w-eta*g)-fw(w)>-0.01*eta*sum(g^2)){
    eta=eta/2
  }
  w=w-eta*g#the new w
  wlist[,j+1]=w#record w
}
```

After 9 iterations, it breaks and comes the conclusion that it hit the stationary point in this case. it is showed below.

```
head(wlist)
```

```
##      [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]    0  1.550000e-03  1.302031e-03  1.362520e-03  1.345814e-03
## [2,]    0  3.239952e-04  3.235151e-04  3.233797e-04  3.234269e-04
## [3,]    0 -6.922280e-06 -6.651840e-06 -6.283871e-06 -6.412551e-06
## [4,]    0  2.592786e-04  2.596882e-04  2.596096e-04  2.596369e-04
## [5,]    0  8.387627e-05  8.371886e-05  8.361728e-05  8.365266e-05
## [6,]    0  3.722696e-05  3.786646e-05  3.856279e-05  3.831768e-05
##               [,6]          [,7]          [,8]          [,9]         [,10]
## [1,]  1.350547e-03  1.349199e-03  1.349583e-03  1.349474e-03  1.349505e-03
## [2,]  3.234130e-04  3.234170e-04  3.234159e-04  3.234162e-04  3.234161e-04
## [3,] -6.374631e-06 -6.385503e-06 -6.382401e-06 -6.383285e-06 -6.383033e-06
## [4,]  2.596288e-04  2.596312e-04  2.596305e-04  2.596307e-04  2.596306e-04
## [5,]  8.364224e-05  8.364523e-05  8.364438e-05  8.364462e-05  8.364455e-05
## [6,]  3.838998e-05  3.836925e-05  3.837516e-05  3.837348e-05  3.837396e-05
##      [,11] [,12]
## [1,]     0     0
## [2,]     0     0
## [3,]     0     0
## [4,]     0     0
## [5,]     0     0
## [6,]     0     0
```

```
flist=vector(mode='numeric',k+1)
for (j in 1:(k+1)) {
  flist[j]=fw(wlist[,j])}
```

```
min(flist)
```

```
## [1] 0.6930365
```

```
plot(log(flist-min(flist)),xlab = "iteration",ylab = "log(f(xk)-f(x*))",main = "Gradient Descent with
 backtracking line search for bigger data")
```