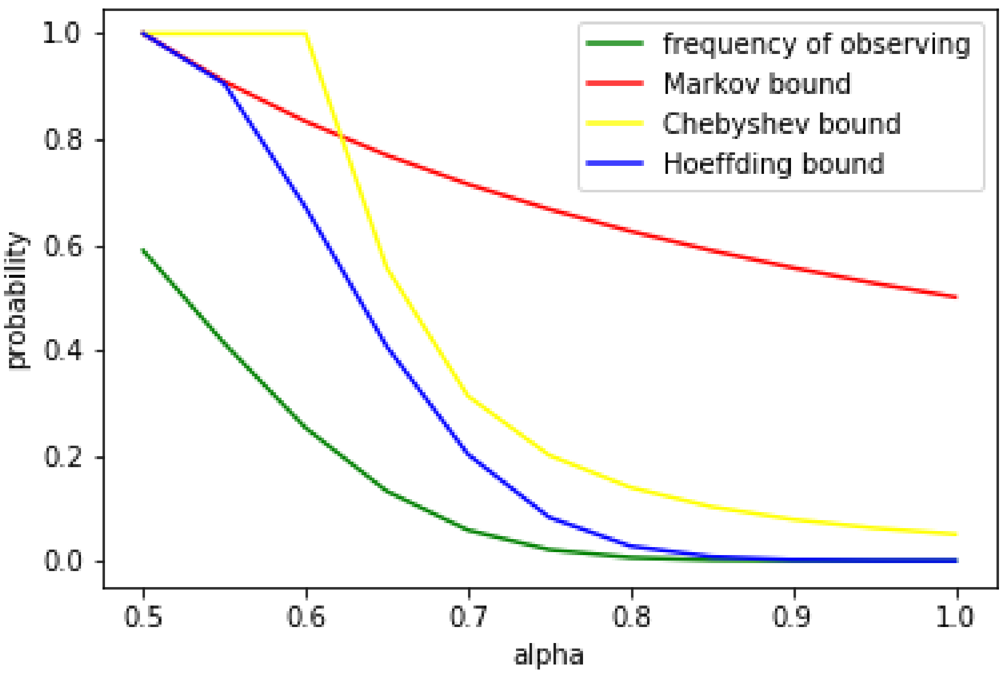
ZHAOYANG XU + CHM564

December 3, 2019

Contents

1 Illustration of Hoeffding's Inequality



2. We can see that, the $Hoeffding's\ bound$ converges to $frequency\ of\ observing$ much faster than the $Chebyshev\ bound$. Especially, when $\alpha \geq 0.85$ the $Hoeffding's\ bound$ almost equal to $frequency\ of\ observing$.

3. From *python*, we can get the probability with Hoeffiding's bound

$$P(\tfrac{1}{20}\sum_{i=1}^{20} X_i \geq 0.95) = 0.0003035391380788668$$
$$P(\tfrac{1}{20}\sum_{i=1}^{20} X_i \geq 1) = 4.5399929762484854e-05$$

From Assignment 1, we can get the exact probability

$$P(\tfrac{1}{20}\sum_{i=1}^{20} X_i \geq 0.95) = 2.002716064453125e-05$$
$$P(\tfrac{1}{20}\sum_{i=1}^{20} X_i \geq 1) = 9.5367431640625e-07$$

The Hoeffiding's bound in $\alpha = 0.95$ is really close to the exact probability. But with the α increasing, the Hoeffiding's bound become much closer and the convergence is much faster, at least at the rate of e^{-20}.

2 The effect of scale (range) and normalisation of random variables in Hoeffding's inequality

Theorem 2.3

$$P(\sum_{i=1}^{n} X_i - E[\sum_{i=1}^{n} X_i] \geq \epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n}(b_i-a_i)^2}}$$

Because $X_i \in [0,1]$, so

$$P(\sum_{i=1}^{n} X_i - E[\sum_{i=1}^{n} X_i] \geq \epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n}(1-0)^2}}$$

Let $\epsilon = nk$,left side

$$\begin{aligned}
P(\sum_{i=1}^{n} X_i - E[\sum_{i=1}^{n} X_i] \geq nk) &= P(\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}E[\sum_{i=1}^{n} X_i] \geq k) \\
&= P(\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}\sum_{i=1}^{n} E[X_i] \geq k) \\
&= P(\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}\sum_{i=1}^{n} \mu \geq k) \\
&= P(\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}\sum_{i=1}^{n} \mu \geq k) \\
&= P(\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}n\mu \geq k) \\
&= P(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq k)
\end{aligned}$$

right side

$$e^{\frac{-2n^2k^2}{\sum_{i=1}^{n}1^2}} = e^{\frac{-2n^2k^2}{n}} = e^{-2nk^2}$$

so Corollary 2.5 be proved $\quad P(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq k) \leq e^{-2nk^2}$

3 Distribution of Student's Grades

1. Markov's inequality:

$$P(X \geq \epsilon) \leq \frac{E[X]}{\epsilon}$$

$$\begin{aligned} P(\hat{Z} \leq z) &= P(-\hat{Z} \geq -z) \\ &= P(100 - \hat{Z} \geq 100 - z) \\ &= P(\hat{Q} \geq 100 - z) \leq \frac{E[\hat{Q}]}{100 - z} \end{aligned}$$

$$E[\hat{Q}] = E[100 - \hat{Z}] = E[100] - E[\hat{Z}] = 100 - 50 = 50$$
$$\frac{50}{100-z} = 0.05$$
$$z = -900$$

2. Chebyshev's inequality:

$$P(|X - E[X]| \geq \epsilon) \leq \frac{Var[X]}{\epsilon^2}$$

$$\begin{aligned} P(\hat{Z} \leq z) &= P(\hat{Q} \geq 100 - z) \\ &= P(\hat{Q} - E[\hat{Q}] \geq 50 - z) \\ &\leq \frac{Var[\hat{Q}]}{(50 - z)^2} \end{aligned}$$

$$Var[\hat{Q}] = E[\hat{Q}^2] - E[\hat{Q}]^2 = 2500$$
$$\frac{2500}{(50 - z)^2} = 0.05$$
$$z = -173.6$$

3. Hoeffding's inequality:

$$P(\sum_{i=1}^n X_i - E[\sum_{i=1}^n X_i] \leq -\epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}}$$
$$z = 3.7$$

4. So, we can find that, only Hoeffding's inequally can get a non-vacuous value is 3.7.

4 The Airline Question

1. We can see it as a binomial distribution
   Markov's bound:

	X_1	X_2
X	1	0
P	0.95	0.05

$$
\begin{aligned}
P(\sum_{i=1}^{100} X_i \geq 100) &\leq \frac{E[\sum_{i=1}^{100} X_i]}{100} \\
&= \frac{100 * 0.95}{100} \\
&= 0.95
\end{aligned}
$$

Chebyshev's bound:

$$
\begin{aligned}
P(\sum_{i=1}^{100} X_i \geq 100) &= P(\sum_{i=1}^{100} X_i - E[\sum_{i=1}^{100} X_i] \geq 100 - E[\sum_{i=1}^{100} X_i]) \\
&\leq \frac{Var[\sum_{i=1}^{100} X_i]}{(100 - E[\sum_{i=1}^{100} X_i])^2} \\
&= \frac{100 * 0.95 * 0.05}{(100 - 95)^2} \\
&= 0.19
\end{aligned}
$$

Hoeffding's bound:

$$
\begin{aligned}
P(\sum_{i=1}^{100} X_i \geq 100) &= P(\sum_{i=1}^{100} X_i - E[\sum_{i=1}^{100} X_i] \geq 100 - E[\sum_{i=1}^{100} X_i]) \\
&\leq 2e^{\frac{-2\epsilon^2}{\sum_{i=1}^{100} (1-0)^2}} \\
&= 0.61
\end{aligned}
$$

So,we choose the Chebyshev's bound.

2. From 1. we use Chebyshev's inequality

$$
\begin{aligned}
P(\sum_{i=1}^{10000} X_i = 9500) &\leq P(\sum_{i=1}^{10000} X_i \geq 9500) \\
&= P(\sum_{i=1}^{10000} X_i - E[\sum_{i=1}^{10000} X_i] \geq 9500 - E[\sum_{i=1}^{10000} X_i]) \\
&\leq \frac{Var[\sum_{i=1}^{10000} X_i]}{(9500 - E[\sum_{i=1}^{10000} X_i])^2} \\
&= \frac{np(1-p)}{(9500-np)^2}
\end{aligned}
$$

Then,we calculate partial Derivative of p, to find the extremum.

5 Logistic Regression

5.1 Cross-entropy error measure

(a) The likelihood for i.i.d S:

$$\prod_i^N P(y_i|x_i)$$

We assume $p = [y = +1]; q = h(x_n)$,the likelihood of the training set:

$$\prod_i q^{(N-n)p}(1-q)^{n(1-p)}$$

so,the negative log-likelihood, divided by N is

$$
\begin{aligned}
-\frac{1}{N} ln(\prod_i^N P(y_i|x_i)) &= -\frac{1}{N} ln(\prod_i q^{(N-n)p}(1-q)^{n(1-p)})) \\
&= \sum_i^N p ln\frac{1}{q} + (1-p) ln\frac{1}{1-q} \\
E_{in}(w) &= \sum_i^N [y_n = +1] ln\frac{1}{h(x_n)} + [y_n = -1] ln\frac{1}{1-h(x_n)}
\end{aligned}
$$

The $h(x_n)$ from maximum likelihood minimizes the negative log-likelihood, and also minimizes the sample error.

(b) the error function:

$$
\begin{aligned}
\frac{1}{N}\sum_{n=1}^{N} ln(1+e^{-y_n w^T x_n}) &= -\frac{1}{N}\sum_{n=1}^{N} ln(\theta(-y_n w^T x_n)) \\
&= -\frac{1}{N}\sum_{i=1}^{N} ln(P(y_i|x_i)) \\
&= -\frac{1}{N} ln(\prod_{i}^{N} P(y_i|x_i))
\end{aligned}
$$

from(a)

$$\tfrac{1}{N}\textstyle\sum_{n=1}^{N} ln(1+e^{-y_n w^T x_n}) = \sum_i^N [y_n=+1] ln\tfrac{1}{h(x_n)} + [y_n=-1] ln\tfrac{1}{1-h(x_n)}$$

5.2 Logistic regression loss gradient

For labels in {-1,1}

in-sample error:

$$E_{in}(w) = \tfrac{1}{N}\textstyle\sum_{n=1}^{N} ln(1+e^{-y_n w^T x_n})$$

Partial derivative of w

$$
\begin{aligned}
\bigtriangledown E_{in}(w) &= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+e^{-y_n w^T x_n}}(e^{-y_n w^T x_n}+1)' \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+e^{-y_n w^T x_n}} e^{-y_n w^T x_n}(-y_n w^T x_n)' \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+\frac{1}{e^{y_n w^T x_n}}} e^{-y_n w^T x_n}(-y_n x_n) \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{e^{y_n w^T x_n}}{1+e^{y_n w^T x_n}} e^{-y_n w^T x_n}(-y_n x_n) \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+e^{y_n w^T x_n}}(-y_n x_n) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n x_n}{1+e^{y_n w^T x_n}}
\end{aligned}
$$

logistic function $\theta(s) = \frac{1}{1+e^{-s}}$, so

$$
\begin{aligned}
\bigtriangledown E_{in}(w) &= -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n x_n}{1+e^{y_n w^T x_n}} \\
&= \frac{1}{N}\sum_{n=1}^{N} -y_n x_n \theta(-y_n w^T x_n)
\end{aligned}
$$

its equals:

$$-\frac{1}{N}\sum_{n=1}^{N}[\frac{y_n+1}{2}-\theta(w^Tx_n)]x_n$$

from this function, when the example is 'misclassified' ,the difference $\frac{y_n+1}{2}-\theta(w^Tx_n)$ is larger than a correctly classified one.

For labels in {0,1}
from {-1,1},we can get

$$-\frac{1}{N}\sum_{n=1}^{N}[\frac{y_n+1}{2}-\theta(w^Tx_n)]x_n \ \frac{y_n+1}{2}\in[0,1]$$

for {0,1}, we can get $y\in[0,1]$ directly.
so

$$-\frac{1}{N}\sum_{n=1}^{N}[y_n-\theta(w^Tx_n)]x_n \ \frac{y_n+1}{2}\in[0,1]$$

5.3 Logistic regression implementation

```
def gradientDescent(x,y):

    eta = [] %define eta
    t = [] %define the time of steps
    w = [] %define the parameter

    for i in range(0,t):

        loss = np.zeros(shape=(10,3))

        for j in range(0,10):

            hypothesis = np.dot(w,x1[j].T)
            loss[j] = np.dot(y[j],x1[j])/(1 + math.exp(np.dot(y[j],
                hypothesis)))

        gradient = loss.sum(axis=0)/(-10)

        w = w - eta * gradient

        return w
```

Firstly, we define $w(0)$. For the first step, we calculate $hx = w(0)^Tx$, and then calculate the gradient $g_0 = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_nx_n}{1+e^{y_nw^T(0)x_n}}$. Update the $w(0)$ with η and $gradient(0)$. After that, use $w(1)$ to repeat this process, until the w(t).

I define the x with $[[1,0],[1,1],[1,2]...[1,8],[1,9]]$, y is a 1*10 matrix and every elements is a random number 1 or 0. And assume $w=[0,0,0]$, $t=10$, $\eta=0.005$
we can get an final $w=[-0.21571744,-1.01579983,-0.21571744]$

5.4 Iris flower data

I use linear regression parameters as $w(0)=[0.53779839,-6.06028193,-0.54198853]$ and $\theta=0.01$, $t=10$.
so, we can get the parameters $w=[0.60310129,-6.05716725,-0.53041294]$

$$y = 0.60310129x_1 - 6.05716725x_2 - 0.53041294$$

By 0-1 loss:

$$Error_{train} = \frac{61}{62}$$
$$Error_{test} = \frac{13}{26}$$