HW12

May 23, 2020

1 HW12

1.1 Alireza Darvishi 96109674

[1]:
```
from scipy.sparse import coo_matrix
import numpy as np
import cvxpy as cp
import matplotlib.pyplot as plt
```

1.1.1 A6.30.b

[2]:
```
from team_data import *
A1 = coo_matrix((train[:, 2], (range(m), train[:, 0])), shape=(m, n)).toarray()
A2 = coo_matrix((-train[:, 2], (range(m), train[:, 1])), shape=(m, n)).toarray()
A = A1 + A2
```

[3]:
```
a = cp.Variable(n)
objective = cp.Minimize(cp.sum(cp.logistic(-2 / sigma * A @ a)))
constraint = [a <= 1, a >= 0]
prob = cp.Problem(objective, constraint)
prob.solve()
a_hat = a.value - min(a.value)
```

1.1.2 A6.30.c

[4]:
```
prediction = np.zeros(shape=(m, 3))
i = 0
for j in range(n):
    for k in range(j + 1, n):
        prediction[i, :] = [j, k, np.sign(a_hat[j] - a_hat[k])]
        i = i + 1
print(
    " prediction accuracy:",
    round(100 * sum(prediction[:, 2] == test[:, 2]) / m, 2),
    "%",
```

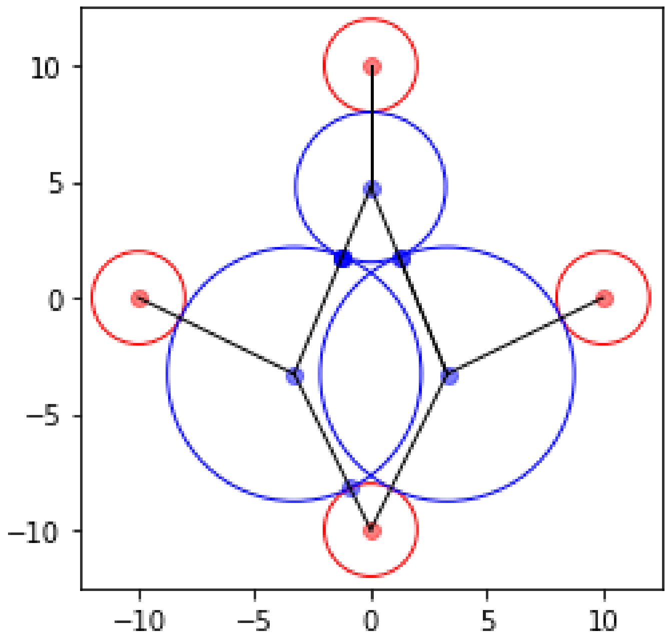
```
    "\n biased prediction accuracy:",
    100 * round(sum(train[:, 2] == test[:, 2]) / m, 2),
    "%",
)
```

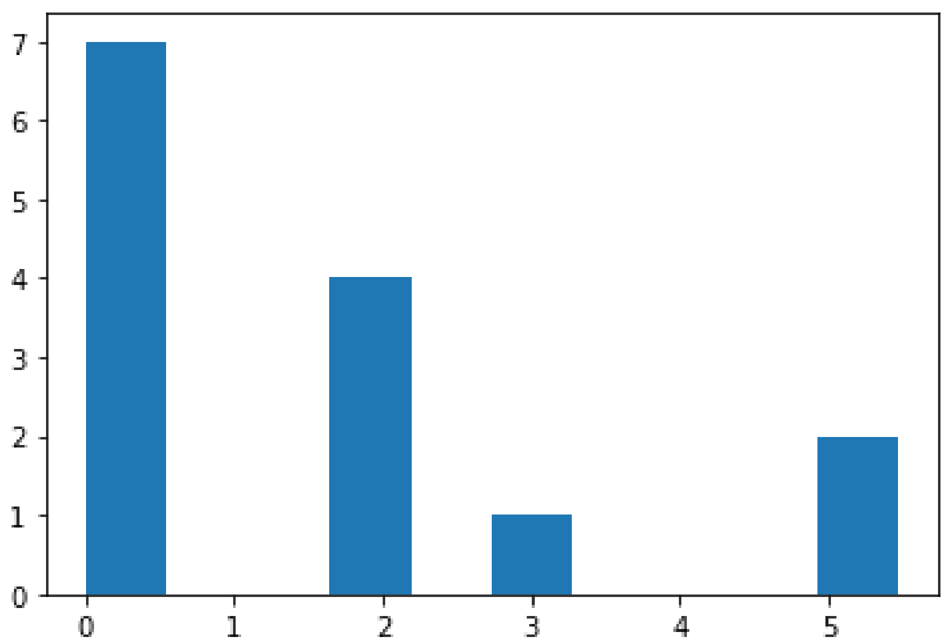
```
prediction accuracy: 75.56 %
biased prediction accuracy: 71.0 %
```

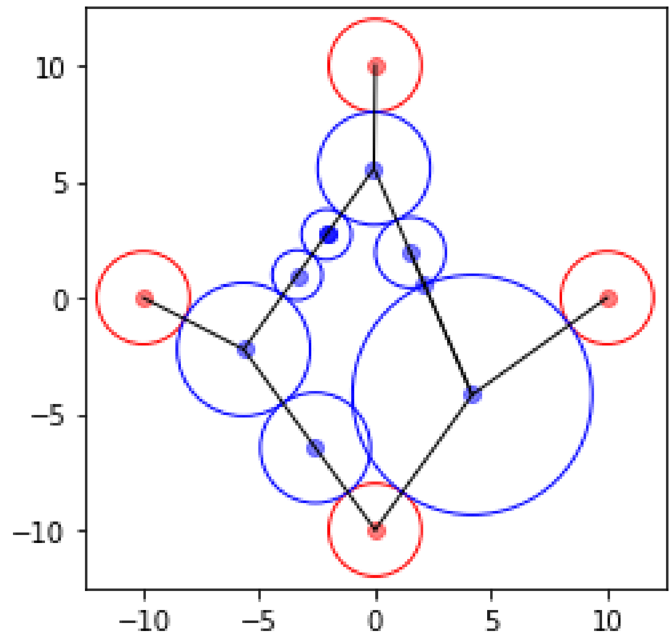
1.1.3 A7.23

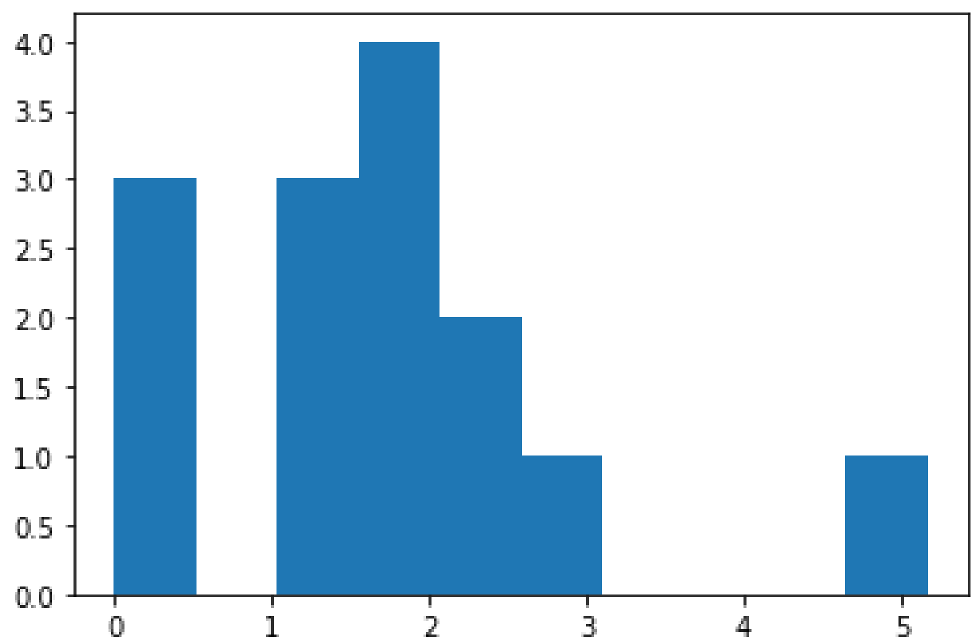
[5]:
```
from disks_data import *

C = cp.Variable(shape=(n, 2))
R = cp.Variable(n)
constraint = [C[:k] == Cgiven, R[:k] == Rgiven, R >= 0]
constraint += [
    cp.norm(C[row[0, 0]] - C[row[0, 1]]) <= R[row[0, 0]] + R[row[0, 1]]
    for row in Gindexes
]
objective1 = cp.Minimize(cp.sum(R))
objective2 = cp.Minimize(cp.sum(R ** 2))
problem1 = cp.Problem(objective1, constraint)
problem2 = cp.Problem(objective2, constraint)
problem1.solve()
plot_disks(C.value, R.value, Gindexes.astype(int))
plt.hist(R.value)
problem2.solve();
plot_disks(C.value, R.value, Gindexes.astype(int))
plt.hist(R.value);
```


10
5
0
−5
−10
−10
−5
0
5
10


7
6
5
4
3
2
1
0
0
1
2
3
4
5





As we can see in histograms, fewer radiuses are near zero in the second histogram but instead, the maximum radius is less than the first solution. In the first solution there are more radiuses equal to zero.

1.1.4 A14.30

[6]:
```
from currency_exchange_data import *

X = cp.Variable(shape=(n, n))
values = np.zeros(n)
for i in range(n):
    values[i] = np.sqrt(F[i, 0] / F[0, i])

c_f = c_init - cp.sum(X, 0) + cp.diag(X @ (1 / F.T))
constraint = [cp.diag(X) == 0, c_init >= cp.sum(X, 0), c_req <= c_f , X>=0]
objective = cp.Maximize(values @ c_f)
problem = cp.Problem(objective, constraint)
print("optimal cost is:",round(c_init @ values - problem.solve(),2))
```

```
optimal cost is: 7.72
```

علیرضا درویشی - 96109674 - تمرین 12 - بهینه سازی محدب

6.30

الف) برای هر بازی، درستنمایی نتیجه‌ی بازی برابر با:

$$\mathrm{Prob}\left(v \geq -y_i \left(a_j^{(i)} - a_k^{(i)} \right) \right)$$

است که در نتیجه درستنمایی کل برابر خواهد بود با:

$$\prod_{i=1}^{m} \left(1 - F\left(-y_i \left(a_j^{(i)} - a_k^{(i)} \right) \right) \right)$$

$$= \prod_{i=1}^{m} \frac{1}{1 + e^{-\frac{2y_i(a_j - a_k)}{\sigma}}}$$

حال با لگاریتم گرفتن از تابع درستنمایی داریم:

$$f = -\sum_{i=1}^{m} \log\left(1 + e^{-\frac{2y_i(a_j - a_k)}{\sigma}} \right)$$

که باید بیشینه شود. هر یک از اعداد در توان هم یکی از عناصر بردار

ستونی $-2\frac{Aa}{\sigma}$ هستند. پس:

Date Year Month Day

Subject

مسئله ی نهایی:

$$\text{minimize} \quad \sum_{i=1}^{m} \log\left(1 + e^{-\frac{2}{\sigma}(Aa)_i}\right)$$

$$0 \le a \le 1$$

7.23)

آ) شرط اشتراک داشتن دایره‌ها را می‌توان به شکل

$$\|C_i - C_j\|_2 \le r_i + r_j$$

نوشت. یعنی فاصله‌ی مرکز دو دایره‌ی فوق کمتر یا مساوی مجموع شعاع آن‌ها باشد. پس:

$$\min \quad f(r)$$

$$\|C_i - C_j\| \le r_i + r_j \quad \text{for} \quad (i,j) \in I$$

$$C_i = C_i^{fix}, \quad r_i = r_i^{fix} \quad \text{for} \quad i = 1, \dots, k$$

$$r_i \ge 0 \quad \text{for} \quad i = 1, \dots, n$$

که برای کمینه کردن مساحت، $f(r) = \sum r_i^2$ است

و برای کمینه کردن محیط ، $f(r)=\sum r_i$ است .

14.7

آ). شرط دوم را به شکل زیر بازنویسی می‌کنیم :

- وجود ندارد p طوری که : (1) $p\geq 0 ,\quad 1^Tp=1 ,\quad R^Tp=0$

چون اگر p شرایط قبلی را داشته باشد ، کافی است در یک ضریب ضرب شود و به $p'=\frac{p}{1^Tp}$ تبدیل شود از طرفی اگر $1^Tp=1$ باشد ، $1^Tp\neq 0$ برقرار است . حال :

$$R'=\begin{bmatrix} R & 1 \end{bmatrix}$$

پس (1) معادل است با : $\nexists p: \ R'^T p = e_{n+1} ,\quad p\geq 0$

که طبق لم فارکاش معادل است با :

$$\exists x'\in R^{n+1}: \ R'x'\leq 0 ,\quad e_{n+1}^T x'>0$$

حال اگر در x' به جای x' از x استفاده کنیم داریم:

$$\exists x' \in R^{n+1} : R'x' \geq 0 \ , \ e^{n+1}x' < 0$$

اگر قرار دهیم: $x' = [x, k]$ داریم:

$$\exists x \in R^n, k \in R : \ Rx + k\mathbf{1} \geq 0 \ , \ k < 0$$

پس:

$$\exists x \in R^n, k \in R : \ k > 0 \ , Rx \geq k$$

که معادل است با:

$$\exists x \in R^n \ , \ Rx > 0$$

ب) با توجه به بخش قبل استراتژی خوب وجود دارد اگر و تنها اگر

P یافت نشود که $R^T P = 0 , P \geq 0 , \Sigma P_i = 1$

$$(R^T P)_i = \sum_{j \neq i} -P_j + \lambda_i P_i = 0 \Rightarrow -1 + (\lambda_i + 1) P_i = 0$$

$$\Rightarrow P_i = \frac{1}{1 + \lambda_i}$$

Date Year Month Day Subject

پس استراتژی خوب وجود دارد اگر و تنها اگر:

(به ازای یک i : $\frac{1}{1+\lambda_i} < 0$) یا ($\sum \frac{1}{1+\lambda_i} \neq 1$)

اما چون $\lambda_i > 0$ پس شرط دوم اتفاق نمی افتد

14.30

ابتدا شرایط مسئله را می نویسیم:

مجموع پول های تبدیل شده از هر واحد نباید از مقدار اولیه بیشتر شود. پس

$$X^T 1 \leq c^{init}$$

حال برای بدست آوردن مقدار نهایی هر ارز داریم:

$$c^f = c^{init} - X^T 1 + diag\left(X \odot \frac{1}{F^T} \right)$$

که:

$$\left(\frac{1}{F^T}\right)_{ij} = (F^T)_{ij}^{-1}$$

حال کافی است:

$$c^f \geq c^{req}$$

حال تابع هدف را باید برحسب بردار v، ارزش هر ارز برحسب دلار نوشت:

$$f = v^T c^f$$

$\max \ v^T c^f$ — پس:

$s.t. \ c^f \geq c^{req}$

$$c^f = C^{init} - x^T 1 + \mathrm{diag}\left(x \ \frac{1}{F^T}\right)$$

$$x^T 1 \leq C^{init}$$

$$x \geq 0$$

مسئله‌ی خطی است.

18.21

الف) $E_i(S_i)$ را با کمک تابع $Pos(x)$ نوشت

$$Pos(x) = \max\{x, 0\} = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

حال: $E_i(S_i) = \frac{1}{2} K_i^{ext} \, Pos(S_i - N_i)^2 + \frac{1}{2} K_i^{comp} \, Pos(N_i - S_i)^2$

چون برای $S_i \geq N_i$ بخش دوم صفر است و برای $S_i \leq N_i$ بخش اول

صفر است. در این حالت $E_i(S_i)$ به سادگی محدب است.

چون $Pos(x)$ محدب است و از روی ترکیب های

مجدد به $E_i(S_i)$ رسیده‌ایم. حال:

$$\min \quad \sum E_i(S_i)$$

$$\text{s.t.} \quad \sum S_i = W - \sum w_i$$

ب) شرط‌های ۱، ۲، ۳ به دلیل عدم وجود نامساوی، فقط

همان: $\sum S_i = W - \sum w_i$ است.

شرط گرادیان:

$$\frac{d}{dx} Pos(x)^2 = 2Pos(x) \quad \text{و} \quad \frac{d}{dx} Pos(-x)^2 = -2Pos(-x)$$

$$\Rightarrow \frac{\partial}{\partial S_i} E_i(S_i) = k_i^{ext} Pos(S_i - N_i) - k_i^{Comp} Pos(N_i - S_i)$$

حال داریم:

$$L(s,v) = \sum E_i(S_i) + v(W' - \sum S_i) \quad , \quad W' = W - \sum w_i$$

$$\frac{\partial L}{\partial S_i} = \frac{\partial E_i(S_i)}{\partial S_i} - v = 0$$

حال با توجه به اینکه $\frac{\partial E(s_i)}{\partial s_i} > 0$ برای $s_i > N_i$ ، $\frac{\partial E(s_i)}{\partial s_i} < 0$

برای $s_i < N_i$. این ۳ حالت را داریم:

(1) $v = 0$:

$$\frac{\partial E(s_i)}{\partial s_i} = 0 \Rightarrow s_i = N_i \Rightarrow \boxed{W = \sum W_i + \sum N_i}$$

(2) $v > 0$:

$$s_i > N_i \Rightarrow k_i^{ext}(s_i - N_i) = v \Rightarrow s_i = N_i + \frac{v}{k_i^{ext}}$$

$$\sum s_i = \sum N_i + v\left(\sum \frac{1}{k_i^{ext}}\right) = W - \sum W_i \Rightarrow v = \frac{W - \sum W_i - \sum N_i}{\sum \frac{1}{k_i^{ext}}}$$

$$\Rightarrow \boxed{s_i = N_i + \frac{1}{k_i^{ext}}\left[\frac{W - \sum W_i - \sum N_i}{\sum \frac{1}{k_i^{ext}}}\right]}$$

(3) $v < 0$

$s_i < N_i$

$$\Rightarrow \boxed{s_i = N_i + \frac{1}{k_i^{comp}}\left[\frac{W - \sum W_i - \sum N_i}{\sum \frac{1}{k_i^{comp}}}\right]}$$