Homework 4
Statistical Learning for Decision Making 2023

Li Ju
li.ju@it.uu.se

March 21, 2023

1 Problem 1

The algorithm of Robust Risk Minimization (RRM) is given as Algorithm 1.

Algorithm 1 Optimization algorithm for RRM

Require: $\tilde{\epsilon}, \mathbf{Z} = \{\mathbf{z}_1, \dots \mathbf{z}_n\}, \theta$ estimator $\hat{\Theta}(\boldsymbol{\omega}, \mathbf{Z})$
  $\boldsymbol{\omega}^0 \leftarrow n^{-1}\mathbf{1}, k = 1$
  while not converge **do**
    $\hat{\theta}^k \leftarrow \hat{\Theta}(\boldsymbol{\omega}^{k-1}, \mathbf{Z})$
    $\boldsymbol{\omega}^k \leftarrow \arg\min_{\boldsymbol{\omega}} \sum_{i=1}^{n} \omega_i^{k-1} \ell_{\theta^k}(\mathbf{z}_i)$ w.r.t. $\mathbb{H}(\boldsymbol{\omega}^k) \geq \ln(1-\tilde{\epsilon})n, \boldsymbol{\omega} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\omega} = 1$
  end while
  Return $\hat{\theta}^k$

The code implementation of the problem in deferred to Appendix A.1. The experiment results are shown as follows in Table 1:

Method	$\tilde{\epsilon}$	$\hat{\theta}$
RRM	0.1	[199.963, 200.520]
RRM	0.2	[198.214, 199.776]
RRM	0.3	[197.140, 200.156]
RRM	0.4	[196.316, 200.627]
ERM	-	[204.982, 200.595]

Table 1: Comparison of ERM and RRM for the estimation of θ

2 Problem 2

a). The graph model of the mediating structure is given as Figure 1. Theoretically we can say that a and y are dependent, unless x is observed.

We simulate the data generating process and the implementation is deferred to Appendix A.2.1. It is hard to test the independence between two variables from samples. Here we test the correlation $\rho(a, y)$ between a and y, given both x observed and unobserved. The results are plotted in Figure 2. It can be observed that $\rho(a, y)$ is large, indicating that a and y are correlated if x is unconditioning. Also $\rho(a, y|x)$ is much less, indicating that a and y are much-less correlated if x is conditioning. Although non-correlation does not necessarily imply independence, still the check is consistent with the theoretical result.

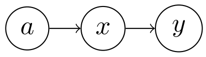

Figure 1: Graph model of 2a

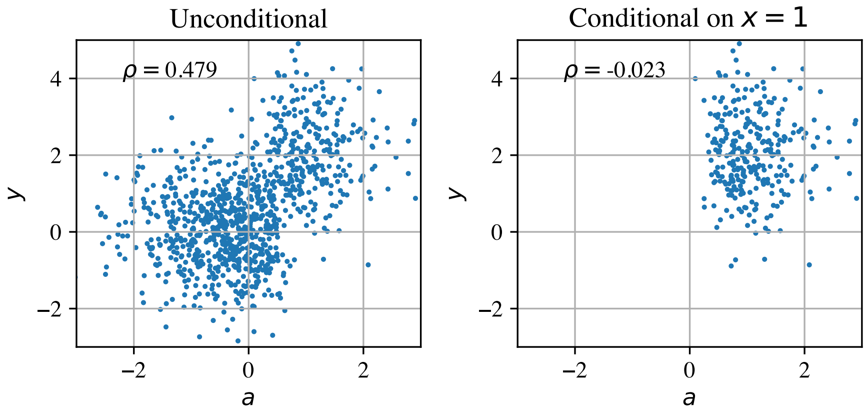

Figure 2: Plots of variable a and y in 2a

b). The graph model of the mediating structure is given as Figure 3. Theoretically we can say that a and y are dependent, no matter whether x is observed or not.

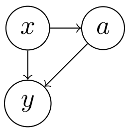

Figure 3: Graph model of 2b

The results are shown in Figure 4 with calculated $\rho(a, y)$ and $\rho(a, y|x)$. It is clear to see that a and y are highly correlated and dependent with both conditioning x or un-conditioning x. The results are consistent with the theoretical result.

c). The graph model of the mediating structure is given as Figure 5. Theoretically we can say that a and y are independent, unless x is observed.

The results are shown in Figure 6 with calculated $\rho(a, y)$ and $\rho(a, y|x)$. It is clear to see that a and y are un-correlated if x is un-conditioning. And $\rho(a, y|x)$ is significantly greater than $\rho(a, y)$ if conditioning on x, indicating that a and y are correlated. The results are consistent with the theoretical result.

3 Problem 3

The proof is derived as follows:

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}, y)}{\int p(\mathbf{x}, y) dy}$$

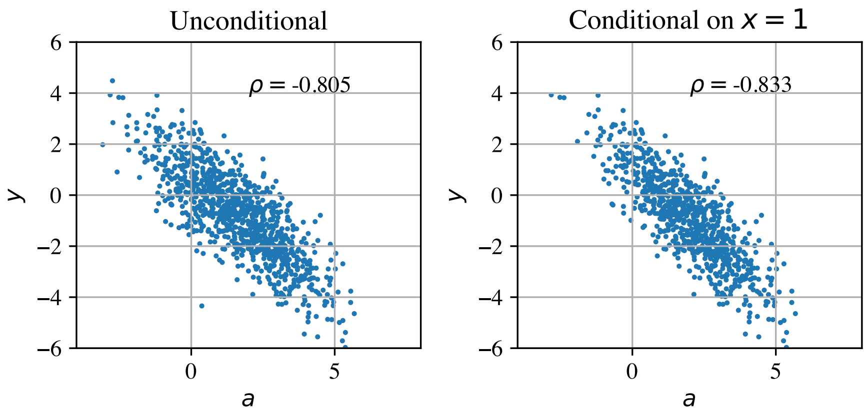

Figure 4: Plots of variable a and y in 2b

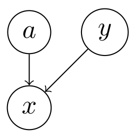

Figure 5: Graph model of 2c

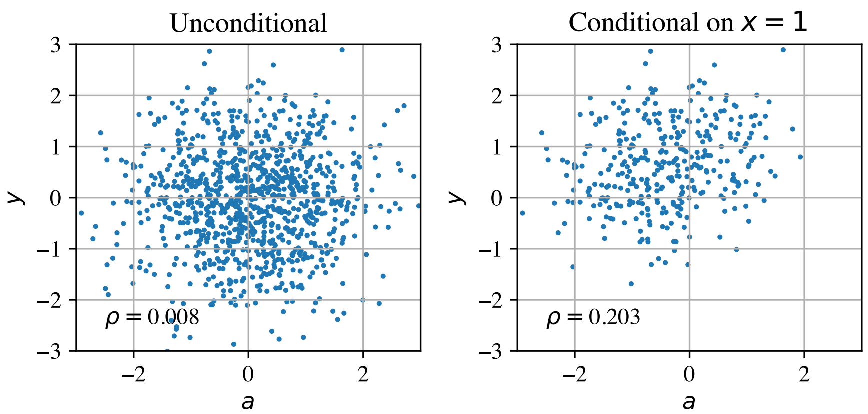

Figure 6: Plots of variable a and y in 2c

According to the graph model, $p(\mathbf{x}, y)$ can be factorized as follows:

$$\begin{aligned} p(\mathbf{x}, y) = & p(x_1)p(x_2)p(x_5|x_1, x_2) \cdot p(x_3)p(x_4)p(x_6|x_3, x_4) \cdot \\ & p(y|x_5, x_6) \cdot p(x_7)p(x_9|x_7) \cdot p(x_{11}|x_9, y) \cdot \\ & p(x_8)p(x_{10}|x_8) \cdot p(x_{12}|x_{10}, y) \cdot p(x_{13}|x_{11}) \cdot \\ & p(x_{14}|x_{11}, x_{12}) \cdot p(x_{15}|x_{12}) \end{aligned}$$

Then we have

$$\begin{aligned} p(y|\mathbf{x}) &= \frac{p(y|x_5, x_6) \cdot p(x_{11}|x_9, y) \cdot p(x_{12}|x_{10}, y)}{\int p(y|x_5, x_6) \cdot p(x_{11}|x_9, y) \cdot p(x_{12}|x_{10}, y)dy} \\ &= \frac{p(y|x_5, x_6) \cdot p(x_{11}|x_9, y) \cdot p(x_{12}|x_{10}, y) \cdot p(x_5)p(x_6)p(x_9)p(x_{10})}{\int p(y|x_5, x_6) \cdot p(x_{11}|x_9, y) \cdot p(x_{12}|x_{10}, y) \cdot p(x_5)p(x_6)p(x_9)p(x_{10})dy} \\ &= \frac{p(\mathbf{x}_s, y)}{\int p(\mathbf{x}_s, y)dy} = \frac{p(\mathbf{x}_s, y)}{p(\mathbf{x}_s)} \\ &= p(y|\mathbf{x}_s) \end{aligned}$$

where $\mathbf{x}_s = (x_5, x_6, x_9, x_{10}, x_{11}, x_{12})$.

4 Problem 4

a). There is casual association between a and y, given $c = \emptyset$ observed.

b). There is no casual association between a and y, given $c = \{w_1\}$ observed.

c). There is no casual association between a and y, given $c = \emptyset$ observed.

A Codes

A.1 Code for Problem 1

```python
import numpy as np
from scipy import stats
import cvxpy as cp
# from matplotlib import pyplot


n = 100
eps = 0.2
eps_tildes = [0.1, 0.2, 0.3, 0.4]

# parameters for true distributions
mu = np.array([200, 200])
sigma = np.array([[400, 50], [50, 400]])
v = 2.5
c = 2 * v / (v - 2) * sigma


# generate samples from true mixed distributions
corrupted_nums = stats.bernoulli.rvs(eps, size=n).sum()

uncorrupted = stats.multivariate_normal.rvs(
    mu, sigma, size=n-corrupted_nums,
)

corrupts = stats.multivariate_t.rvs(
    mu, c, df=v, size=corrupted_nums,
)

zs = np.concatenate([uncorrupted, corrupts])
np.random.shuffle(zs)

# pyplot.scatter(x=uncorrupted[:, 0],
#                y=uncorrupted[:, 1],
#                c="blue")
# pyplot.scatter(x=corrupts[:, 0],
#                y=corrupts[:, 1],
#                c="red")

# pyplot.show()


def loss(zs, theta):
    return ((zs - theta) ** 2).sum(axis=1)


def estimate_theta(weights, zs):
    return np.average(zs, axis=0, weights=weights)


def estimate_weights(zs, theta, eps_tilde):
    losses = loss(zs, theta).transpose()
```

```
    # print(losses)
    weights = cp.Variable(n, pos=True)
    prob = cp.Problem(
        cp.Minimize(losses @ weights),
        [np.ones(n).transpose() @ weights == 1,
         cp.sum(cp.entr(weights)) >=
         np.log((1-eps_tilde)*zs.shape[0]),
         ])
    prob.solve(verbose=False)
    # prob.solve(verbose=True)
    return weights.value


for eps_tilde in eps_tildes:

    weights = 1/n*np.ones(n)
    last_theta = np.array([0., 0.])
    threshold = 1e-4
    improvement = 1e3

    while improvement >= threshold:
        theta = estimate_theta(weights, zs)
        improvement = np.linalg.norm(theta - last_theta)
        last_theta = theta
        weights = estimate_weights(zs, theta, eps_tilde)
    print(f"Eps_tilde: {eps_tilde} with theta: {theta}")
print(f"ERM theta {estimate_theta(None, zs)}")
```

A.2 Code for Problem 2

A.2.1 Code for Problem 2a

```
from scipy import stats
import numpy as np
from matplotlib import pyplot as plt

n = 1000


def s(x):
    return 1/(1+np.exp(-x))


a = stats.norm.rvs(0, 1, size=n)
x = np.array([stats.bernoulli.rvs(
    s(10*(each-0.5))) for each in a])
y = np.array([stats.norm.rvs(
    2*each, 1) for each in x])

f, (ax1, ax2) = plt.subplots(1, 2, figsize=(6, 3))
ax1.scatter(a, y, s=2)
ax2.scatter(a[x.astype(bool)], y[x.astype(bool)],
            s=2)

unc_coref = np.corrcoef(a, y)[0, 1]
```

```python
c_coref = np.corrcoef(a[x.astype(bool)], y[x.astype(bool)])[0, 1]

ax1.annotate(r"$\rho=$"+f"{unc_coref:.3f}", (-2.2, 4))
ax2.annotate(r"$\rho=$"+f"{c_coref:.3f}", (-2.2, 4))


ax1.set_xlabel(r"$a$")
ax1.set_ylabel(r"$y$")

ax2.set_xlabel(r"$a$")
ax2.set_ylabel(r"$y$")

ax1.set_xlim((-3, 3))
ax1.set_ylim((-3, 5))
ax2.set_xlim((-3, 3))
ax2.set_ylim((-3, 5))

ax1.set_title("Unconditional")
ax2.set_title(r"Conditional on $x=1$")

plt.tight_layout()
plt.savefig("hw4_2a.pdf", dpi=500)
```