Optimal Transportation, Matching Problem and Hedonic Model

YuHaiyang, WangXiangtong, ZhangZhiyi, XuJiahe and WangSong

0. Review of professor's problems

Hedonic pricing is a problem in economics where buyers and sellers match together according to their preferences to buy (respectively sell) different goods. Try to understand the mathematical structure of this problem (it can be formulated in terms of optimal transport maps). Can the resulting mathematical problem be given a factories and mines type of interpretation (hint, imagine that you have two, rather than one, type of factory, producing say iron and aluminum, and you want to build your mines in locations that minimize some total transport cost). Can the problem also be rewritten as a matching (that is, pure optimal transport) problem?

1 Optimal Transportation Model

1.1 Background

Transport theoryis a name given to the study of optimal transportation and allocation of resources. The problem was formalized by the French mathematician Gaspard Mongein 1781. Major advances were made in the field during World War II by the Soviet mathematician and economist Leonid Kantorovich. So,the problem is sometimes known as the Monge-Kantorovich transportation problem.

1.2 Example

$c : \mathbf{R}^2 \times \mathbf{R}^2 \to [0, \infty)$

$$c(T) := \sum_{m \in M} c(m, T(m))$$

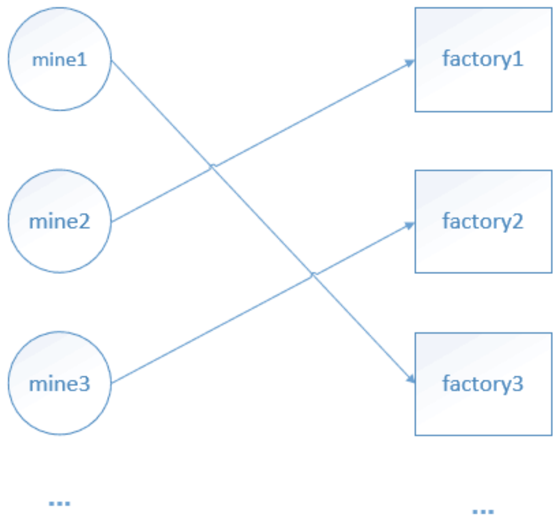


Figure 1: Mines and factories

1.3 Abstract formulation of the problem

Let X and Y be two separable metric spaces such that any probability measure on X (or Y) is a Radon measure (i.e. they are Radon spaces). Let $c : X \times Y \to [0, +\infty]$ be a Borel-measurable function. Given probability measures μ on X and ν on Y, Monge's formulation of the optimal transportation problem is to find a transport map $T : X \to Y$ that realizes the infimum

$$\inf \left\{ \int_X c(x, T(x)) \, \mathrm{d}\mu(x) \middle| T_*(\mu) = \nu \right\}$$

where $T_*(\mu)$ denotes the push forward of μ by T. A map T that attains this infimum (i.e. makes it a minimum instead of an infimum) is called an "optimal transport map".

2 Matching Problem Model

Let X, Y, and Z be finite, disjoint sets, and let T be a subset of $X \times Y \times Z$. That is, T consists of triples (x, y, z) such that $x \in X$, $y \in Y$, and $z \in Z$. Now $M \subseteq T$ is a 3-dimensional matching if the following holds: for any two distinct triples $(x_1, y_1, z_1) \in M$and $(x_2, y_2, z_2) \in M$, we have $x_1 \neq x_2$, $y_1 \neq y_2$, and $z_1 \neq z_2$.

3 Hedonic Price Model

3.1 Introduction

Hedonic pricing is a model, which identifies price factors, according to the premise that price is determined both by internal characteristics of the good being sold and external factors affecting it.

A hedonic pricing model is often used to estimate quantitative values for ecosystem or environmental services that directly impact market prices for homes.

3.2 Formula

$$P = c + \sum_{n=1}^{N} \beta_n X_n + \epsilon$$

P: the price of house

X_i: the different characters of house

ϵ: deviation

β_i: the coefficient of each character

$$X = (X_1, X_2, \ldots, X_n)$$

3.3 Competitive market

To simplify the model, we suppose the market is a competitive market.

4 Standing hypotheses of hedonic model

To allow for the possibility that some agents choose not to participate, we augment the spaces $X := X_0 \cup \{\phi_X\}$ $Y := Y_0 \cup \{\phi_Y\}$ and$Z := Z_0 \cup \{\phi_Z\}$ by including an isolated point in each: a partner ϕ_X for any unmatched sellers, a partner ; ϕ_Y for any unmatched buyers, and the null contract ϕ_Z. Preferences are encoded into functions representing the utility $u(x, z)$ of product $z \in Z$ to buyer $x \in X$, and the utility (disutility or cost) $v(y, z)$ of product $z \in Z$ to seller $y \in Y$. These utility functions $u : X \times Z \to R \cup \{-\infty\}$ and $v : Y \times Z \to R \cup \{+\infty\}$ are specified a priori, along with non-negative Borel measures μ_0 on X_0 and ν_0 on Y_0 of finite total mass representing the distribution of buyer and seller types throughout the population. The utility functions are constrained so that neither the dummy buyer type ϕ_X nor the dummy seller type ϕ_Y can participate in any exchange save the null contract:

$$u(\phi_X, z) = \begin{cases} 0 & \text{if } z = \phi_Z \\ -\infty & \text{else,} \end{cases}$$

$$u(\phi_Y, z) = \begin{cases} 0 & \text{if } z = \phi_Z \\ +\infty & \text{else,} \end{cases} \tag{1}$$

while the measures μ_0 and ν_0 are extended to X and Y by assigning mass $\nu_0(Y_0) + 1$ and $\mu_0(X_0) + 1$ to the points ϕ_X and ϕ_Y respectively:

$$\mu := \mu_0 + (\nu_0(Y_0) + 1)\delta_{\phi_X} \qquad \nu := \nu_0 + (\mu_0(X_0) + 1)\delta_{\phi_Y} \tag{2}$$

The augmented measures balance $\mu[X] = \nu[Y] < \infty$.

To guarantee the convergence of various integrals, and attainment of various suprema and infima, we assume throughout (and tacitly hereafter) that $u(x, z) > -\infty$ extends upper semicontinuously to the completion of $X \times Z$ and $v(y, z) < \infty$ lower semicontinuously to the completion of $Y \times Z$. We normalize the utility of the null-contract to be zero

$$u(x, \phi_Z) = 0 = v(y, \phi_Z) \tag{3}$$

which can be achieved without loss of generality if the reserve utilities $u(x, \phi_Z) \in L^1(X, d\mu)$ and $v(y, \phi_Z) \in L^1(X, d\nu)$ are continuous and integrable, by subtracting them from u and v.

Define the pairwise surplus function

$$s(x, y) = \sup_{z \in Z} u(x, z) - v(y, z) \tag{4}$$

We assume that for each pair the supremum is attained. Further, in case u or$v$ is discontinuous or Z fails to be compact, we assume the set of contracts

$$Z(x,y) = \arg\max_{z \in Z} u(x,z) - v(y,z) \tag{5}$$

that maximize the surplus (4) is non-empty, compact, and depends upper hemicontinuously on $(x,y) \in X \times Y$. It is well-known that there exists a measurable selection, i.e., a Borel function $z_0 : X \times Y \to Z$ such that $z_0(x,y)$ is contained in $Z(x,y)$ for all (x,y).

Suppose $P : Z \to \mathbf{R} \cup \{\pm\infty\}$denotes the competitive market price of quality $z \in Z$. To allow non-participation, it is subject to the constraint $P(\phi_Z) = 0$. We assume that buyer utility is linear in price so that in such a market, the indirect utility available to buyer type $x \in X$ is defined by the quasi-linear utility maximization

$$U(x) = \sup_{z \in Z}\{u(x,z) - P(z)\}$$

Here $U(x) \geq 0$ is non-negative since $\phi_Z \in Z$; each buyer x retains the right not to consume. Similarly, we assume seller utility is linear in price so that the indirect utility available to seller type $y \in Y$ is given by the utility maximization

$$V(y) = \sup_{z \in Z}\{P(z) - v(y,z)\}$$

with $V(y) \geq 0$ and vanishing in the case of non-participation.

Let α be a non-negative measure on $X \times Y \times Z$. The support of α refers to the smallest closed set $\mathrm{Spt}(\alpha) \subseteq X \times Y \times Z$ of full mass. The measure α represents an assignment of buyers and sellers to each other and to products. We use the push-forward notation to denote its marginal projections $\pi^X_\#\alpha$ and $\pi^Y_\#\alpha$ under mappings such as $\pi^X(x,y,z)=x$ and $\pi^Y(x,y,z)=y$ on $X\times Y\times Z$. The pair (α,P) is an hedonic equilibrium if these projections coincide with the initial measures on each set:

$$\pi^X_\#\alpha=\mu$$

$$\pi^Y_\#\alpha=\nu$$

and if, for α-almost all points $(x,y,z)\in \mathrm{Spt}\alpha$, we have that

$$U(x)=u(x,z)-P(z)$$

$$V(y)=P(z)-v(y,z)$$

5 The associated matching problem

$s:X\times Y\to[0,\infty)$ is upper semicontinuous by our assumptions (3), and the set $Z(x,y)$ where the supremum is attained (5) is non-empty, compact-valued, and upper hemicontinuous. Our normalizations (1)-(3) permit either buyer or seller to go unmatched (to match with a null type) and force the utility of the unmatched state to be zero:

$$s(x,\phi_Y)=u(x,\phi_Z)=0$$

$$s(\phi_X, y) = -v(\phi_Z, y) = 0$$

One can then define a pairwise matching problem by the set of *buyers* (X, μ), the set of *sellers* (Y, ν), and the pairwise surplus defined by the surplus function s. An *assignment* (or a *matching*) is defined as a measure γ on $X \times Y$, the marginals of which coincide with μ and ν. Using the same notations as above, we thus write that

$$\pi_X^{\#}\gamma = \mu$$

$$\pi_Y^{\#}\gamma = \nu$$

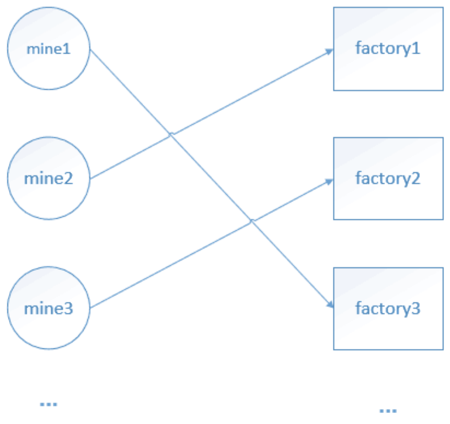
where the projection mappings $\pi^X(x; y) = x$ and $\pi^Y(x; y) = y$ this time are defined on $X \times Y$. If $(x, y) \in (X_0 \times Y_0) \cap \mathrm{Spt}(\gamma)$, we say that x and y are matched. A buyer may be matched to multiple sellers and vice versa. If $x \in X_0$ and there is no $y \in Y_0$ such that $(x, y) \in \mathrm{Spt}(\gamma)$, we say that x is unmatched (and similarly for y).

A payoff corresponding to γ is a pair of functions $\overline{U} : X \to \mathbf{R}$ and $\overline{V} : Y \to \mathbf{R}$ with the normalization $\overline{U}(\phi_X) = 0$ such that for any $(x, y) \in \mathrm{Spt}(\gamma)$

$$\overline{U}(x) + \overline{V}(y) \leq s(x, y)$$

Finally, an outcome is defined as a triple $(\gamma, \overline{U}, \overline{V})$ where $(\overline{U}, \overline{V})$ is a payoff corresponding to γ

6 Factories and mines interpretation



7 How to Realize the Problem

7.1

From what above-mentioned we can simplify the problem as follow:

1.We have two sets A and B, they have the same number of elements

2.We know the cost of every connection

3.We need to find a one to one match for every element in two sets

4.The flow can be different between edges, we need to minimize the total cost

7.2

Obviously this is a network problem. As constraint conditions involve the cost of every flow and one-to-one condition, this problem cant be solved by The famous max-flow min-cut theorem .

By establishing an origin and a destination and making the cost of edge be the product of the flow and corresponding cost , then the problem can be solved by cost-flow algorithms.

But the time efficiency is not pleasant. Because the graph is too dense, there are too many edges on it.

7.3

I found a better way to solve the problem.

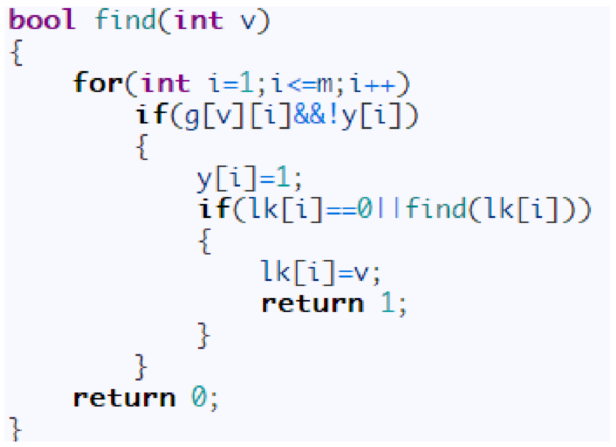
As we can see, the number of elements are the same in 2 sets and besides the origin and destination, there are only 2 layers in the graph.

According to the condition: one-to-one match, we can easily remind the famous algorithm : Hungarian Algorithm

Remember, there is no cost in problems solved by Hungarian Algorithm

7.4

Cardinal code in Hungarian Algorithm



```
bool find(int v)
{
    for(int i=1;i<=m;i++)
        if(g[v][i]&&!y[i])
        {
            y[i]=1;
            if(lk[i]==0||find(lk[i]))
            {
                lk[i]=v;
                return 1;
            }
        }
    return 0;
}
```

Base on the inspiration of cost-flow solution ,we only need to solve the 2-set match problem with cost on edges.

As it is quite brutal in the process of Hungarian Algorithm, we might calculate the best result in the process, but we cant ensure it is the best solution.

7.5

When we get to the best result we can easily find that:

$$\sum_{i=1}^{x_i \in X} lx(x_i) + \sum_{i=1}^{y_i \in Y} ly(y_i) = K = \sum weight(x_i, y_i)$$

$weight(a, b)$ means the cost of edge (a, b)

$lx(x_i)$ means the contribution of x_i to the whole system

$ly(y_i)$ means the contribution of y_i to the whole system

Also we can draw the conclusion that for every edge in the best deployment we have:

$$lx(x) + ly(y) = weight(x, y)$$

7.6

In the beginning, we can only set $lx(x_i)$ to be the max cost of edges that link to point x_i, all $ly(y_i)$ be 0. Then we need to adjust the value of $lx(x_i)$ and $ly(y_i)$ to reach the equation.

Before we reach the equation , we can easily find that

$$lx(x) + ly(y) \geq weight(x, y)$$

So the problem become that we should ensure that every element has a match and at the same time we need to adjust $lx(x_i)$ and $ly(y_i)$ to reach the equation.

Overall, we just need to adjust the value of $lx(x_i)$ and $ly(y_i)$ and then we use Hungarian algorithm to check if it is a one-to-one match. The threshold will only be decreasing. So we can make sure that the solution is the best.

7.7

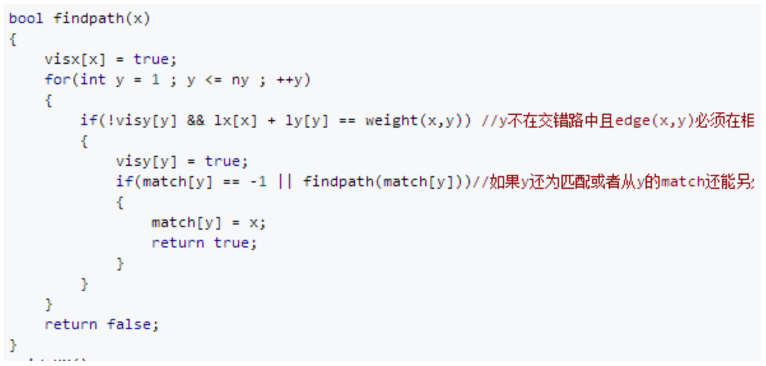
The strategy to reach the threshold:

We set a parameter: cnt

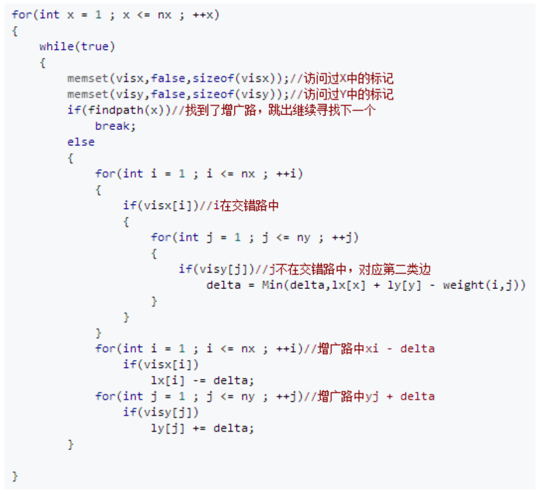
When we failed to get a one-to-one match All the $lx(x_i)$ which have a match got to minus cnt. (to decrease its contribution or importance) To make the system be in a balance, all the $ly(y_i)$ which have a match got to plus cnt.

We can get the best result only when the equation holds , so the set of cnt can only be the difference of the cost of edges whose end points are both be chose already.

7.8 Cardinal code



```
bool findpath(x)
{
    visx[x] = true;
    for(int y = 1 ; y <= ny ; ++y)
    {
        if(!visy[y] && lx[x] + ly[y] == weight(x,y)) //y不在交错路中且edge(x,y)必须在
        {
            visy[y] = true;
            if(match[y] == -1 || findpath(match[y]))//如果y还为匹配或者从y的match还能另
            {
                match[y] = x;
                return true;
            }
        }
    }
    return false;
}
```



```
for(int x = 1 ; x <= nx ; ++x)
{
    while(true)
    {
        memset(visx,false,sizeof(visx));//访问过X中的标记
        memset(visy,false,sizeof(visy));//访问过Y中的标记
        if(findpath(x))//找到了增广路，跳出继续寻找下一个
            break;
        else
        {
            for(int i = 1 ; i <= nx ; ++i)
            {
                if(visx[i])//i在交错路中
                {
                    for(int j = 1 ; j <= ny ; ++j)
                    {
                        if(visy[j])//j不在交错路中，对应第二类边
                            delta = Min(delta,lx[x] + ly[y] - weight(i,j))
                    }
                }
            }
            for(int i = 1 ; i <= nx ; ++i)//增广路中xi - delta
                if(visx[i])
                    lx[i] -= delta;
            for(int j = 1 ; j <= ny ; ++j)//增广路中yj + delta
                if(visy[j])
                    ly[j] += delta;
        }

}
```

7.9

By now we can find that the time cost is still quite unpleasant

Finding that there are many redundant operations in trading the graph And the process of calculating cnt is too slow. By using some tree structure, the process of calculating cnt can be much faster

Code : https://paste.ubuntu.com/p/BZ6ZDj7gqr/