Yuma Optimization

Datura AI

September 8, 2024

Complete Algorithm

We now write out the algorithm as in the code. In the next section, we discuss some simplifications.

Firstly, a convenient naming: A vector whose entries are positive and sum to one is called a *probability vector*. Thus, post-normalization, we have probability vectors for rows/columns.

Input:

An $n \times n$ weight matrix W_0.

1. *Max upscale W_0:* Define the Max upscale function f as

$$f(w) = \frac{u_{16}}{\max(w)} w, \tag{1}$$

where $u_{16} := 2^{16} - 1$, and the input w is a (row) vector of length n. To each row of W_0, apply f:

$$W_0 = \sum_{i,j=1}^{n} w_{ij}^0 |i\rangle\langle j| \to u_{16} \sum_{i,j=1}^{n} \frac{w_{ij}^0}{m_i} |i\rangle\langle j| =: W_1, \tag{2}$$

where $m_i := \max_j w_{ij}^0$.

2. *Row-normalize the matrix W_1:*

$$W_1 = \sum_{i,j=1}^{n} w_{ij}^1 |i\rangle\langle j| \to \sum_{i,j=1}^{n} \frac{w_{ij}^1}{\sum_k w_{ik}^1} |i\rangle\langle j| =: W_2, \tag{3}$$

where

$$w_{ij}^1 = \frac{u_{16} w_{ij}^0}{m_i}. \tag{4}$$

It follows that

$$\frac{w_{ij}^1}{\sum_k w_{ik}^1} = \frac{u_{16} w_{ij}^0 / m_i}{u_{16} \sum_k w_{ik}^0 / m_i} = \frac{w_{ij}^0}{\sum_k w_{ik}^0}, \tag{5}$$

and thus

$$W_2 = \sum_{i,j=1}^{n} \frac{w_{ij}^0}{s_i^0} |i\rangle\langle j|, \tag{6}$$

where $s_i^0 = \sum_{k=1}^{n} w_{ik}^0$ is the sum of the ith row vector. It is sufficient to directly row-normalize the original weight matrix. The Max-upscale step is vacuous.

3. *Apply cutoff with consensus vector:* Given the consensus vector v,

$$W_2 = \sum_{i,j=1}^{n} \frac{w_{ij}^0}{s_i^0} |i\rangle\langle j| \to \sum_{i,j=1}^{n} \min\left(v_j, \frac{w_{ij}^0}{s_i^0}\right) |i\rangle\langle j| =: W_3. \tag{7}$$

4. *Compute the* `bonds_delta` *matrix D as:*

$$D := \sum_{i,j=1}^{n} w_{ij}^3 \cdot c_i |i\rangle\langle j|, \tag{8}$$

   where $w_{ij}^3 = \min\left(v_j, \frac{w_{ij}^0}{s_i^0}\right)$. Here we scale each row of W_3 with the corresponding scalar entry of the *stake vector* c.

5. *Column normalize D:*

$$D := \sum_{i,j=1}^{n} c_i w_{ij}^3 |i\rangle\langle j| \to \sum_{i,j=1}^{n} \frac{c_i}{s_j^D} w_{ij}^3 |i\rangle\langle j| =: D', \tag{9}$$

   where $s_j^D := \sum_{i=1}^{n} c_i w_{ij}^3$.

6. *Compute current Bond matrix B_t:*

   The current iterate of the bonds matrix is computed as

$$B_t := \mathrm{EMA}(B_{t-1}) = (1-\alpha) B_{t-1} + \alpha D' \equiv \sum_{i,j=1}^{n} b_{ij}^t |i\rangle\langle j|. \tag{10}$$

7. *Column normalize B_t:*

$$B_t' := \sum_{i,j=1}^{n} \frac{b_{ij}^t}{s_j^B} |i\rangle\langle j|, \tag{11}$$

   where $s_j^B := \sum_i b_{ij}^t$. Note that if B_{t-1} is already column normalized, then there is no need to column normalize B_t. This is because if B_{t-1} is column-normalized, then each of its columns is a probability vector. Since we column-normalized D', its columns are also probability vectors. And it can be shown that the weighted sum (with weights adding to 1) of two probability vectors is also a probability vector. This is indeed what we do in the EMA step.

8. *Compute dividends vector y:*

$$y = B_t' \cdot z, \tag{12}$$

   where z is the *incentive vector*.

1 Thoughts and Comments

Let us discuss what are the important steps. We assume we are given the weight matrix which is already row-normalized.

- The first important step is the cutoff step. This implies that there is no value in choosing a weight vector r^* whose entries dominate corresponding entries of v, as it will get cut-off. The first thing to study is whether there is value in having higher weights after cutoff. I suspect this is the case.

- The next important step is in the computation of D. Here, we scale each row differently (with the application of the stake vector s). This necessarily means that it would be difficult to have a large value of dividends if the stake assigned to us is small.

- The final important step is in computing the dividends y. Here it helps if larger values of the row $d_\star$ is in the same positions of the larger values of the incentive vector z.

2 Solution Approaches

2.1 ϵ-spiking

The hypothesis we are testing is whether spiking the value at any one entry, at the cost of losing a small amount at every other entry, is advantageous or not.

Consider the pre-normalized vectors $\overline{r} = (\overline{r}_1, \ldots, \overline{r}_i, \ldots, \overline{r}_n)$ and $\overline{r}^\epsilon = (\overline{r}_1, \ldots, \overline{r}_k + \epsilon, \ldots, \overline{r}_n)$. That is, $\overline{r}^\epsilon$ is the same as $\overline{r}$ except at the kth entry, where it is $\overline{r}_k + \epsilon$ for a *small* positive number ϵ. We want to see if $\overline{r}$ or $\overline{r}^\epsilon$ leads to a higher reward, keeping everything else constant.

Without loss of generality, we can assume that $\overline{r}$ is already normalized: $\sum_{i=1}^n \overline{r}_i = 1$. We denote the normalized versions without the 'bar'. Hence $r = \overline{r}$. Now let us consider $\overline{r}^\epsilon$. The sum of $\overline{r}^\epsilon$ must be $1 + \epsilon$. Thus the normalized version of $\overline{r}^\epsilon$ is:

$$r^\epsilon = \frac{\overline{r}^\epsilon}{\mathrm{sum}(\overline{r}^\epsilon)} = \frac{\overline{r}^\epsilon}{1+\epsilon} = \frac{1}{1+\epsilon}(\overline{r}_1, \ldots, \overline{r}_k + \epsilon, \ldots, \overline{r}_n). \tag{13}$$

Let us compare the entries of r and r^ϵ:

$$r_i - r_i^\epsilon = \begin{cases} \frac{\epsilon}{1+\epsilon} r_i & \text{if } i \neq k, \\ \frac{\epsilon}{1+\epsilon}(r_k - 1) & \text{if } i = k. \end{cases} \tag{14}$$

The first type of terms ($i \neq k$) are positive, which indicates $r_i > r_i^\epsilon$, except for the $i = k$ case where the difference is negative (as $r_k < 1$), and thus $r_k < r_k^\epsilon$.

Thus, by *spiking* the kth entry, we gain at k at the cost of losing everywhere else except at k. Is this advantageous? Well, it depends on a lot of factors. For example, if the spiking leads to the entry r_k^ϵ exceeding the corresponding element v_k of the consensus vector v, then the gain is diminished or, in the worst case, negated altogether.

For now, let us assume that both r and r^ϵ are dominated by v, and thus the cutoff function leaves both the vectors invariant. Since we assume that the cutoff step leaves the vectors invariant, we next have to compute the `bonds_delta` matrix D. Let us denote the two cases as D and D^ϵ. Since the stake vector c is constant, we have:

$$D = \begin{pmatrix} c_1 r_{11} & \cdots & c_1 r_{1n} \\ \vdots & \ddots & \vdots \\ c_n r_{n1} & \cdots & c_n r_{nn} \end{pmatrix}, \quad D^\epsilon = \begin{pmatrix} c_1 r_{11}^\epsilon & \cdots & c_1 r_{1n}^\epsilon \\ \vdots & \ddots & \vdots \\ c_n r_{n1} & \cdots & c_n r_{nn} \end{pmatrix}. \tag{15}$$

Since c_1 is a positive number, we have the ordering preserved. That is:

$$c_1 r_{1k} > c_1 r^\epsilon_{1k} \quad \text{if } k \neq i \quad \text{and} \quad c_1 r_{1k} < c_1 r^\epsilon_{1k} \quad \text{if } k = i. \tag{16}$$

We next column-normalize D and D^ϵ. Upon column normalization, we obtain:

$$D' := \begin{pmatrix} \frac{c_1 r_{11}}{c_1 r_{11}+m_1} & \frac{c_1 r_{12}}{c_1 r_{12}+m_2} & \cdots & \frac{c_1 r_{1n}}{c_1 r_{1n}+m_n} \\ \frac{c_2 r_{21}}{c_1 r_{11}+m_1} & \frac{c_2 r_{22}}{c_1 r_{12}+m_2} & \cdots & \frac{c_2 r_{2n}}{c_1 r_{1n}+m_n} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad D'^\epsilon := \begin{pmatrix} \frac{c_1 r^\epsilon_{11}}{c_1 r^\epsilon_{11}+m_1} & \frac{c_1 r^\epsilon_{12}}{c_1 r^\epsilon_{12}+m_2} & \cdots & \frac{c_1 r^\epsilon_{1n}}{c_1 r^\epsilon_{1n}+m_n} \\ \frac{c_2 r_{21}}{c_1 r^\epsilon_{11}+m_1} & \frac{c_2 r_{22}}{c_1 r^\epsilon_{12}+m_2} & \cdots & \frac{c_2 r_{2n}}{c_1 r^\epsilon_{1n}+m_n} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \tag{17}$$

where $m_j = \sum_{i=2}^n c_i r_{ij} = \text{RowSum}_j(D') - c_1 r_{1j} = \text{RowSum}_j(D'^\epsilon) - c_1 r^\epsilon_{1j}$. That is, m_j is the sum over the jth column excluding the first rows of both D and D^ϵ.

The column normalization does not change the ordering either, so we still have the previous standing of r dominating at every entry except k, where r^ϵ dominates.

The bonds matrix computation step does not change the ordering either. Moreover, we can also ignore the contribution B_t has to the dividends as it contributes the same in both cases – with and without spiking. Now, we compute the rewards.

We can ignore the rewards part from B_t (as its contribution is the same in both cases) and just focus on the D' and D'^ϵ parts. We then have:

$$y = \langle d, z \rangle \quad \text{and} \quad y^\epsilon = \langle d^\epsilon, z \rangle, \tag{18}$$

where d and d^ϵ are the first rows of D' and D'^ϵ. We are interested in seeing if y or y^ϵ is larger. Let us compute:

$$y^\epsilon - y = \langle d^\epsilon - d, z \rangle \equiv \langle \Delta, z \rangle, \tag{19}$$

where $\Delta_i = d^\epsilon_i - d_i$.

Thus, the reward difference is:

$$\sum_{i=1}^n z_i (d^\epsilon_i - d_i) = \sum_{i=1}^n z_i \Delta_i. \tag{20}$$

Noting that

$$d^\epsilon_i = \frac{c_i r^\epsilon_i}{c_i r^\epsilon_i + m_i} \quad \text{and} \quad d_i = \frac{c_i r_i}{c_i r_i + m_i}, \tag{21}$$

we have

$$y^\epsilon - y = \sum_{i=1}^n c_i z_i \left(\frac{r^\epsilon_i}{c_i r^\epsilon_i + m_i} - \frac{r_i}{c_i r_i + m_i} \right) = \sum_{i=1}^n z_i \left(\frac{r^\epsilon_i}{r^\epsilon_i + \frac{m_i}{c_i}} - \frac{r_i}{r_i + \frac{m_i}{c_i}} \right) = \sum_{i=1}^n z_i \Delta_i, \tag{22}$$

where

$$\Delta_i = \frac{r^\epsilon_i}{r^\epsilon_i + \frac{m_i}{c_i}} - \frac{r_i}{r_i + \frac{m_i}{c_i}}. \tag{23}$$

The aim is to study when Δ_i becomes positive (and when it becomes negative). In particular, we want to see how the quantities

$$\frac{r^\epsilon_i}{r^\epsilon_i + \frac{m_i}{c_i}} \quad \text{and} \quad \frac{r_i}{r_i + \frac{m_i}{c_i}} \tag{24}$$

compare. Since $\frac{m_i}{c_i}$ is a positive number, we can show that the ordering is preserved. That is,

$$r_i^\epsilon > r_i \implies \frac{r_i^\epsilon}{r_i^\epsilon + \frac{m_i}{c_i}} > \frac{r_i}{r_i + \frac{m_i}{c_i}} \iff \Delta_i \geq 0. \tag{25}$$

However, the amount by which it differs is partly controlled by $\frac{m_i}{c_i}$ in the denominator. For example, if

$$\frac{m_i}{c_i} \gg r_i^\epsilon \implies \Delta_i \approx 0. \tag{26}$$

What this in turn means is that any difference the ϵ-spiking might have contributed to the ith difference Δ_i is diminished if $m_i \gg c_i$. This is a blessing in disguise. What this allows us to do is to spike entries with small $\frac{m_i}{c_i}$. This will contribute to a positive Δ_i at the points we spike. However, as we have seen, we will be at a disadvantage at other entries, but this is alright because at other entries we will have (relatively) larger values of $\frac{m_i}{c_i}$, and thus the negative Δ_i is (relatively) diminished.

Thus by spiking at values where $\frac{m_i}{c_i}$ is high, we should be able to obtain a net positive reward gain.