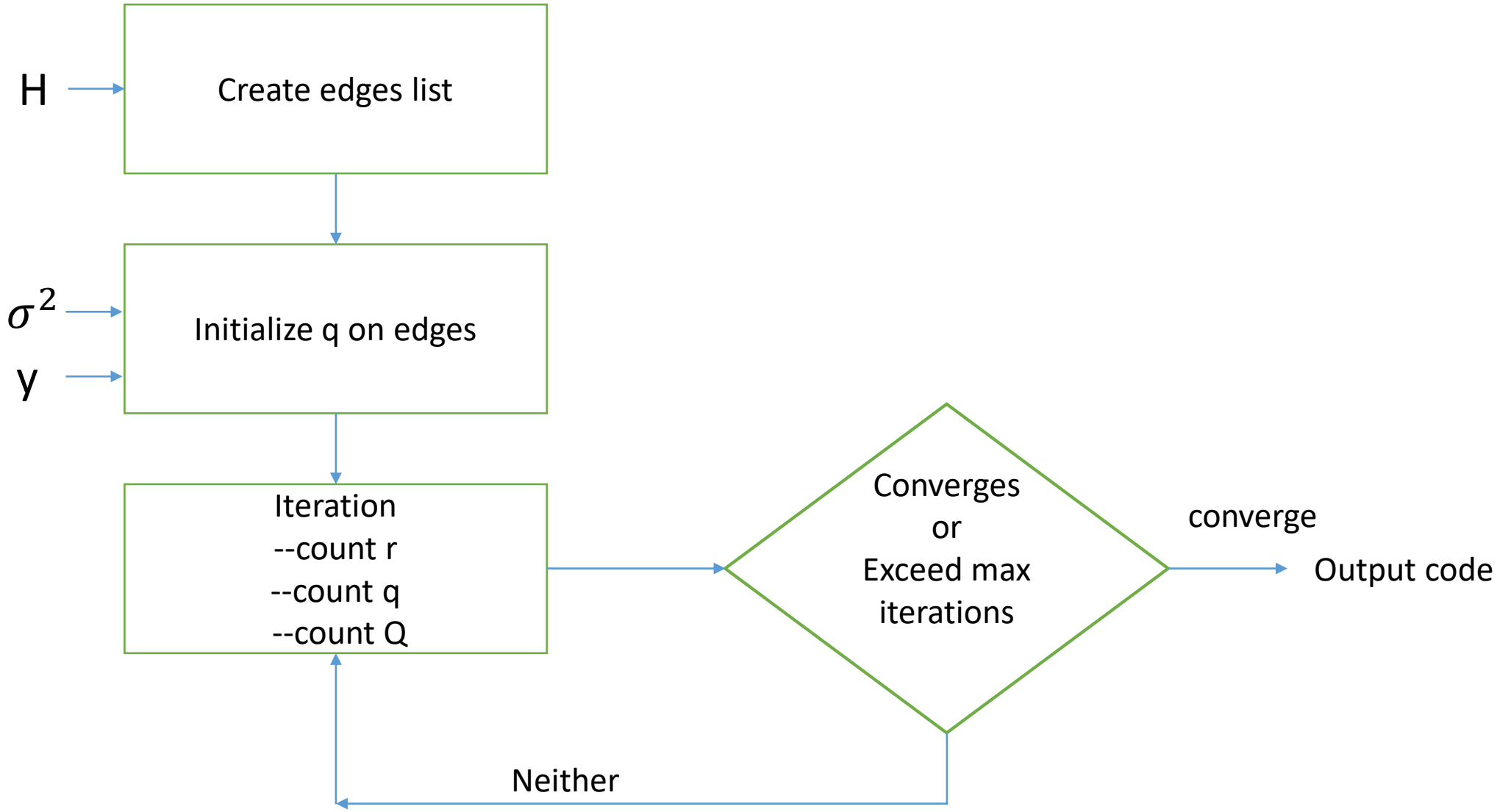
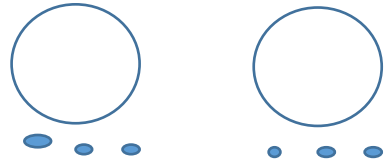

H
Create edges list
σ^2
y
Initialize q on edges
Iteration
--count r
--count q
--count Q
Converges
or
Exceed max
iterations
converge
Output code
Neither

*Type: Edge

- $q0\ , q1$: $q_{ij}(0), q_{ij}\ (1)$
- $r0\ , r1$: $r_{ji}(0),\ r_{ji}\ (1)$

*Edge 編號、分組



- 按照c node連出左到右的順序
- 組別存成

  int cnode[]：(直接編號) c1個一組

  int vnode[第i個vnode][第j條連出去的]：參照cnode編號

receive

- 接收code(讀txt檔)
- 先計算初始q值

$$\frac{1}{1+\exp(-\frac{2y_j t}{\sigma^2})}$$

$$\frac{2y_j}{\sigma^2}$$

initialize

- 將q值放進edge裡

*count_r

- 第一個迴圈是跑r值所在的edge的group
- 第二個迴圈是跑r值所在的edge
- 第三個迴圈是跑算r值要用的edge(取q值)

$$r_{ij}(0) = \frac{1}{2} + \frac{1}{2}\prod_{j' \in v_i \backslash \mathrm{j}} [1 - 2q_{ij'}(1)]$$

$$LLR(r_{ij}) = \prod_{j' \in V_i \backslash j} sign(LLR(q_{ij'})) \cdot \phi\left(\sum_{j' \in V_i \backslash j} \phi(\left|LLR(q_{ij'})\right|) \right)$$

$$r_{ij}(1) = 1 - r_{ij}(0)$$

fi(x)

- Log domain 才需要

- $$\phi(\mathrm{x}) = -\log\left(\tanh\frac{x}{2}\right)$$

*count_q

- 第一個迴圈是跑q值所在的edge的group
- 第二個迴圈是跑q值所在的edge
- 第三個迴圈是跑算q值要用的edge(取r值)

$$q_{ij}(0) = K_{ij}(1 - P_j) \prod_{i' \in C_j \setminus i} r_{i'j}(0)$$

$$q_{ij}(1) = K_{ij} P_j \prod_{i' \in C_j \setminus i} r_{i'j}(1)$$

$$LLR(q_{ij}) = LLR(x_j) + \sum_{i' \in C_j \setminus i} LLR(r_{i'j})$$

*count_Q

- 第一個迴圈是跑vnode
- 第二個迴圈是跑算q值要用的edge(取r值)

*check

- 用cnode找每一行的1當同位置codeword也是1，sum = sum+1
- Sum 不被2整除=>還沒收斂

iteration

- 算r算q
- 檢查是否超過疊代次數限制
- 檢查收斂
- 輸出每次的Q跟預估codeword

test

```
iteration 1:
Q :     0.7686  0.8694  0.9647  0.5076  0.7426
0.9479  0.7199  0.9853
c = 1 1 1 1 1 1 1 1

iteration 2:
Q :     0.1751  0.1836  0.9668  0.2330  0.4637
0.9722  0.8305  0.9919
c = 0 0 1 0 0 1 1 1

iteration 3:
Q :     0.7232  0.4609  0.9667  0.6308  0.8091
0.9498  0.6802  0.9909
c = 1 0 1 1 1 1 1 1

iteration 4:
Q :     0.5307  0.2683  0.9731  0.1477  0.4095
0.9811  0.8392  0.9922
c = 1 0 1 0 0 1 1 1

iteration 5:
Q :     0.7564  0.5799  0.9672  0.3425  0.7573
0.9558  0.8102  0.9896
c = 1 1 1 0 1 1 1 1

iteration 6:
Q :     0.4544  0.2089  0.9736  0.2136  0.6243
0.9686  0.8412  0.9924
c = 0 0 1 0 1 1 1 1

iteration 7:
Q :     0.7399  0.3381  0.9692  0.4086  0.7869
0.9567  0.7754  0.9923
c = 1 0 1 0 1 1 1 1
```

iteration = 1
Q1 =0.7686 0.8694 0.9647 0.5076 0.7426 0.9479 0.7199 0.9853

c = 1 1 1 1 1 1 1 1

iteration = 2
Q1 = 0.1751 0.1836 0.9668 0.2330 0.4637 0.9722 0.8305 0.9919

c = 0 0 1 0 0 1 1 1

iteration = 3
Q1 = 0.7232 0.4609 0.9667 0.6308 0.8091 0.9498 0.6802 0.9909

c = 1 0 1 1 1 1 1 1

iteration = 4
Q1 = 0.5307 0.2683 0.9731 0.1477 0.4095 0.9811 0.8392 0.9922

c = 1 0 1 0 0 1 1 1

iteration = 5
Q1 = 0.7564 0.5799 0.9672 0.3425 0.7573 0.9558 0.8102 0.9896

c = 1 1 1 0 1 1 1 1

iteration = 6
Q1 = 0.4544 0.2089 0.9736 0.2136 0.6243 0.9686 0.8412 0.9924

c = 0 0 1 0 1 1 1 1

iteration = 7
Q1 = 0.7399 0.3381 0.9692 0.4086 0.7869 0.9567 0.7754 0.9923

c = 1 0 1 0 1 1 1 1

test

```
iteration 1:
Q1 :    -1.2002 -1.8953 -3.3092 -0.0306 -1.0597
-2.9008 -0.9439 -4.2042
c = 1 1 1 1 1 1 1 1

iteration 2:
Q1 :    1.5468  1.4922  -3.3721 1.1913  0.1455
-3.5547 -1.5889 -4.8064
c = 0 0 1 0 0 1 1 1

iteration 3:
Q1 :    -0.9636 0.1597  -3.3695 -0.5342 -1.4442
-2.9399 -0.7554 -4.6958
c = 1 0 1 1 1 1 1 1

iteration 4:
Q1 :    -0.1260 1.0060  -3.5885 1.7551  0.3618
-3.9470 -1.6528 -4.8423
c = 1 0 1 0 0 1 1 1

iteration 5:
Q1 :    -1.1363 -0.3181 -3.3868 0.6568  -1.1426
-3.0748 -1.4533 -4.5524
c = 1 1 1 0 1 1 1 1

iteration 6:
Q1 :    0.1771  1.3366  -3.6086 1.3079  -0.5124
-3.4322 -1.6687 -4.8717
c = 0 0 1 0 1 1 1 1

iteration 7:
Q1 :    -1.0517 0.6781  -3.4519 0.3750  -1.3111
-3.0966 -1.2404 -4.8641
c = 1 0 1 0 1 1 1 1

iteration 8:
Q1 :    -0.6200 1.1183  -3.5748 1.5695  -0.3944
-3.6310 -1.7212 -4.8980
c = 1 0 1 0 1 1 1 1
```

```
iteration = 1
LQ = -1.2002   -1.8953   -3.3092   -0.0306   -1.0597   -2.9008   -0.9439   -4.2042
c = 1    1    1    1    1    1    1    1
iteration = 2
LQ = 1.5499    1.4922   -3.3721    1.1913    0.1455   -3.5547   -1.5889   -4.8064
c = 0    0    1    0    0    1    1    1
iteration = 3
LQ = -0.9605    0.1568   -3.3680   -0.5354   -1.4442   -2.9399   -0.7545   -4.6958
c = 1    0    1    1    1    1    1    1
iteration = 4
LQ = -0.1229    1.0031   -3.5876    1.7531    0.3659   -3.9473   -1.6520   -4.8420
c = 1    0    1    0    0    1    1    1
iteration = 5
LQ = -1.1331   -0.3222   -3.3854    0.6521   -1.1379   -3.0733   -1.4512   -4.5529
c = 1    1    1    0    1    1    1    1
iteration = 6
LQ = 0.1830    1.3318   -3.6083    1.3031   -0.5077   -3.4307   -1.6673   -4.8708
c = 0    0    1    0    1    1    1    1
iteration = 7
LQ = -1.0455    0.6718   -3.4495    0.3697   -1.3064   -3.0952   -1.2390   -4.8631
c = 1    0    1    0    1    1    1    1
```