Problem Set 2

Problem 1: MAP IP VS. LP

1.Suppose that G is a cycle with an even number of vertices. For each k > 0, what is the MAP assignment in the case that w_a = a for all a ∈ $\{1, \dots, k\}$?

Solution: we know that the distribution is non-uniform:

$$p(x) = \frac{1}{Z}\prod_{i\in V}\phi(x_i)\prod_{(i,j)\in E}\psi(x_i, x_j)$$

Where $\phi(x_i) = e^{x_i}$, $\psi(x_i, x_j) = 1_{x_i \neq x_j}$,

So: $$p(x) = \frac{1}{Z}\prod_{i\in V} e^{x_i}\prod_{(i,j)\in E} 1_{x_i \neq x_j}$$

MAP assignment: $X = \max_x p(x)$, the graph G is a cycle and with even number of vertices. For i^{th}node, we can assign the number of k or (k - 1) colors to the i^{th}node, and then we can assign (k - 1) or k colors to the $i + 1^{th}$node.

The first kind of assignment:

Assign k colors to all the nodes with odd index, and assign (k-1) colors to the nodes with even index. $x_i, i \in (1,2, \dots ,2n)$

$$x_i = \begin{cases} k & i \in (1, 3, \dots, 2n-1) \\ k-1, & i \in (2, 4, \dots, 2n) \end{cases}$$

The second kind of assignment:

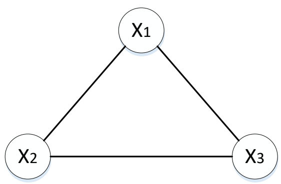
Assign (k-1) colors to all the nodes with odd index, and assign k colors to the nodes with even index. $x_i, i \in (1,2, \dots ,2n)$

$$x_i = \begin{cases} k & i \in (2, 4, \dots, 2n) \\ k-1, & i \in (1, 3, \dots, 2n-1) \end{cases}$$

For instance, if the number of the vertices is $|V| = 4$,the max assignment can be:

(k, k-1, k, k-1) or (k-1, k, k-1, k).

2. Write down the MAP IP and LP for a cycle on three vertices with k=3 and weights w1 = 1, w2 = 2, w3 = 3. What is the optimal value of the MAP IP? What is the optimal value of the MAP LP.



The graph is just like above, we should maximize the following formula:

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i) \log \phi_i (x_i) + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log \psi_{ij} (x_i, x_j)$$

We know that $\phi(x_i) = e^{x_i}, \psi(x_i, x_j) = 1_{x_i \neq x_j}$, so we should maximize the following:

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i)\, x_i + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log 1_{x_i \neq x_j}$$

If $x_i = x_j$, we can conclude that the $\log 1_{x_i \neq x_j}$ will be negative infinity, so in this case, $\tau_{ij}(x_i, x_j)$ should be 0, where $x_i = x_j$. That is:

$\tau_{12}(x_1 = 1, x_2 = 1) = 0$, $\tau_{12}(x_1 = 2, x_2 = 2) = 0$, $\tau_{12}(x_1 = 3, x_2 = 3) = 0$

$\tau_{13}(x_1 = 1, x_3 = 1) = 0$, $\tau_{13}(x_1 = 2, x_3 = 2) = 0$, $\tau_{12}(x_1 = 3, x_3 = 3) = 0$

$\tau_{23}(x_2 = 1, x_3 = 1) = 0$, $\tau_{23}(x_2 = 2, x_3 = 2) = 0$, $\tau_{23}(x_2 = 3, x_3 = 3) = 0$

We also have:

$\tau_1(x_1 = 1)+\tau_1(x_1 = 2)+\tau_1(x_1 = 3) = 1$;

$\tau_2(x_2 = 1)+\tau_2(x_2 = 2)+\tau_2(x_2 = 3) = 1$;

$\tau_3(x_3 = 1)+\tau_3(x_3 = 2)+\tau_3(x_3 = 3) = 1$;

We know that: $\sum_{x_j} \tau_{ij}(x_i, x_j) = \tau_i(x_i)$, so we can get,

$\tau_1(x_1 = 1)=\tau_{12}(x_1 = 1, x_2 = 1)+\tau_{12}(x_1 = 1, x_2 = 2)+\tau_{12}(x_1 = 1, x_2 = 3)$;

$\tau_1(x_1 = 2)=\tau_{12}(x_1 = 2, x_2 = 1)+\tau_{12}(x_1 = 2, x_2 = 2)+\tau_{12}(x_1 = 2, x_2 = 3)$;

$\tau_1(x_1 = 3)=\tau_{12}(x_1 = 3, x_2 = 1)+\tau_{12}(x_1 = 3, x_2 = 2)+\tau_{12}(x_1 = 3, x_2 = 3)$;

Thus:

$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i)\, x_i + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log 1_{x_i \neq x_j}$;

$= \max_{\tau}\{1 * \tau_1(x_1 = 1) + 2 * \tau_1(x_1 = 2) + 3 * \tau_1(x_1 = 3) + 1 * \tau_2(x_2 = 1) + 2 * \tau_2(x_1 = 2) + 3 * \tau_2(x_2 = 3) + 1 * \tau_3(x_3 = 1) + 2 * \tau_3(x_3 = 2) + 3 * \tau_3(x_3 = 3)\}$

Case1, for the Integer Programming(IP):

$\tau_i(x_i) \in \{0,1\}$, for each $i \in V \; and \; x_i$

$\tau_{ij}(x_i, x_j) \in \{0,1\}$, for each $(i,j) \in E \; and \; x_i, x_j$

x_1, x_2, and x_3 are symmetric, we can assume that

$\tau_1(x_1 = 3) = 1$, $\tau_2(x_2 = 2) = 1$ and $\tau_3(x_3 = 1) = 1$;

Or $\tau_1(x_1 = 3) = 1$, $\tau_3(x_3 = 2) = 1$ and $\tau_2(x_2 = 1) = 1$;

$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i)\, x_i + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log 1_{x_i \neq x_j}$;

= 3 * 1 + 2 * 1 + 1* 1 = 6

Case2, for the LP:

We know that $\phi(x_i) = e^{x_i}$, $\psi(x_i, x_j) = 1_{x_i \neq x_j}$, so we should maximize the following:

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i)\, x_i + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log 1_{x_i \neq x_j}$$

$$== \max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i)\, x_i$$

Such that:

$\sum_{x_i} \tau_i(x_i) = 1$;

$\sum_{x_j} \tau_{ij}(x_i, x_j) = \tau_i(x_i)$;

$\tau_i(x_i) \geq 0$ and $\tau_i(x_i) \leq 1$;

$\tau_{ij}(x_i, x_j) \geq 0$ and $\tau_{ij}(x_i, x_j) \leq 1$

We can see this is a linear programming problem, solve this problem, we can get:

$\tau_1(x_1 = 1)$=$\tau_2(x_2 = 1)$=$\tau_3(x_3 = 1)$=0;

$\tau_1(x_1 = 2)$=$\tau_2(x_2 = 2)$=$\tau_3(x_3 = 2)$=1/2;

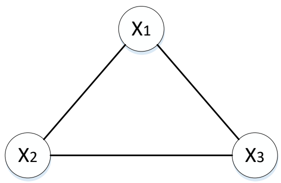
$\tau_1(x_1 = 3)$=$\tau_2(x_2 = 3)$=$\tau_3(x_3 = 3)$=1/2;

The max value for LP is: (1*0 + 2*1/2 + 3*1/2) * 3 = 7.5

Problem2: Loopy Belief Propagation

1. Implement the sum-product algorithm for the coloring problem.

This is a detailed interpretation about the sum-product algorithm:



Part1: compute the all the messages.

We can get $\phi(x_i) = e^{x_i}$, $\psi(x_i, x_j) = 1_{x_i \neq x_j}$, $x_i \in \{1,2,3\}$ from the problem 1.

Initialize all the messages based on the graph above:

$$m^0_{1\to2}(x_2) = m^0_{1\to3}(x_3) = m^0_{2\to1}(x_1) = m^0_{2\to3}(x_3) = m^0_{3\to1}(x_1) = m^0_{3\to2}(x_2) = 1$$

Then:

$$m^t_{i\to j}(x_j) = \sum_{x_i} \{\phi(x_i)\psi(x_i, x_j) \prod_{k \in N(i)\setminus j} m^{t-1}_{k\to i}(x_i)\}$$

For instance:

$$m^1_{1\to2}(x_2=1)=\sum_{x_1}\{\phi(x_1)\psi(x_1,x_2=1)\prod_{k\in N(i)\setminus j} m^0_{3\to1}(x_1)\}$$

Where x_1 can be {1,2,3}, thus we can get the value of $m^1_{1\to2}(x_2=1)$, then we can get the other two values $m^1_{1\to2}(x_2=2)$, $m^1_{1\to2}(x_2=3)$.

Then we should normalize the message for avoiding overflow, the normalization constant can be computed by this formula:

$$\eta^t_{i\to j}=1/\sum_{x_i}^{t} m^t_{i\to j}(x_j)$$

Then update the value of the message using the normalization constant:

$m^1_{1\to2}(x_2=1)$ = $1/(m^1_{1\to2}(x_2=1)$ +$m^1_{1\to2}(x_2=2)$ + $m^1_{1\to2}(x_2=3))$

$m^1_{1\to2}(x_2=2)$ = $1/(m^1_{1\to2}(x_2=1)$ +$m^1_{1\to2}(x_2=2)$ + $m^1_{1\to2}(x_2=3))$

$m^1_{1\to2}(x_2=3)$ = $1/(m^1_{1\to2}(x_2=1)$ +$m^1_{1\to2}(x_2=2)$ + $m^1_{1\to2}(x_2=3))$

Similarly, we can get all the messages about the graph:

$m^1_{1\to2}(x_2=1)$,$m^1_{1\to2}(x_2=2)$,$m^1_{1\to2}(x_2=3)$, $m^1_{1\to3}(x_3=1)$, $m^1_{1\to3}(x_2=2)$, $m^1_{1\to3}(x_3=3)$;

$m^1_{2\to1}(x_1=1)$,$m^1_{2\to1}(x_1=2)$,$m^1_{2\to1}(x_1=3)$, $m^1_{2\to3}(x_3=1)$, $m^1_{2\to3}(x_2=2)$, $m^1_{2\to3}(x_3=3)$;

$m^1_{3\to1}(x_1=1)$,$m^1_{3\to1}(x_1=2)$,$m^1_{3\to1}(x_1=3)$, $m^1_{3\to2}(x_2=1)$, $m^1_{3\to2}(x_2=2)$, $m^1_{3\to2}(x_2=3)$;

Finally, iterate for its times, we will get the final values for all the messages.

Part2: compute all the Beliefs,

The formula for computing the belief is:

$$b_i(x_i)=\phi(x_i)\prod_{k\in N(i)} m_{k\to i}(x_i)$$

For instance, based on the graph above, we can get the beliefs:

$b_1(x_1=1)=\phi(x_1=1)m_{2\to1}(x_1=1)m_{3\to1}(x_1=1)$;

$b_1(x_1=2)=\phi(x_1=2)m_{2\to1}(x_1=2)m_{3\to1}(x_1=2)$;

$b_1(x_1=3)=\phi(x_1=3)m_{2\to1}(x_1=3)m_{3\to1}(x_1=3)$;

We need to normalize these values for avoiding the overflow, the normalization constant:

$$Z=1/\sum_{k\in(1,2,3)} b_1(x_1=k)$$

We can get the beliefs after the normalization:

$b_1(x_1 = 1) = 1/(b_1(x_1 = 1) + b_1(x_1 = 2) + b_1(x_1 = 3))$;

$b_1(x_1 = 2) = 1/(b_1(x_1 = 1) + b_1(x_1 = 2) + b_1(x_1 = 3))$;

$b_1(x_1 = 3) = 1/(b_1(x_1 = 1) + b_1(x_1 = 2) + b_1(x_1 = 3))$;

Similarly, we can get all the beliefs about the nodes:

$b_1(x_1 = 1)$; $b_1(x_1 = 2)$; $b_1(x_1 = 3)$;

$b_2(x_2 = 1)$; $b_2(x_2 = 2)$; $b_2(x_2 = 3)$;

$b_3(x_3 = 1)$; $b_3(x_3 = 2)$; $b_3(x_3 = 3)$;

We also need to compute the between two nodes, the formula is:

$$b_{ij}(x_i, x_j) = \phi(x_i)\phi(x_j)\psi(x_i, x_j) \prod_{k \in N(i)\setminus j} m_{k\to i}(x_i) \prod_{k \in N(j)\setminus i} m_{k\to j}(x_j)$$

For instance, we can all the belief values about b_{12} based on the graph above:

$b_{12}(x_1 = 1, x_2 = 1) = \phi(x_1 = 1)\phi(x_2 = 1)\psi(x_1 = 1, x_2 = 1)m_{3\to 1}(x_1 = 1)m_{3\to 2}(x_2 = 1)$;

We can see there are 9 possible combinations for $b_{12}(x_1, x_2)$; all the $b_{ij}(x_i, x_j)$ are:

$b_{12}(x_1 = 1, x_2 = 1)$; $b_{12}(x_1 = 1, x_2 = 2)$; $b_{12}(x_1 = 1, x_2 = 3)$

$b_{12}(x_1 = 2, x_2 = 1)$; $b_{12}(x_1 = 2, x_2 = 2)$; $b_{12}(x_1 = 2, x_2 = 3)$

$b_{12}(x_1 = 3, x_2 = 1)$; $b_{12}(x_1 = 3, x_2 = 2)$; $b_{12}(x_1 = 3, x_2 = 3)$

There are 9 possible combinations for $b_{13}(x_1, x_3)$; all the $b_{ij}(x_i, x_j)$ are:

$b_{13}(x_1 = 1, x_3 = 1)$; $b_{13}(x_1 = 1, x_3 = 2)$; $b_{13}(x_1 = 1, x_3 = 3)$

$b_{13}(x_1 = 2, x_3 = 1)$; $b_{13}(x_1 = 2, x_3 = 2)$; $b_{13}(x_1 = 2, x_3 = 3)$

$b_{13}(x_1 = 3, x_3 = 1)$; $b_{13}(x_1 = 3, x_3 = 2)$; $b_{13}(x_1 = 3, x_3 = 3)$

There are 9 possible combinations for $b_{23}(x_1, x_3)$; all the $b_{ij}(x_i, x_j)$ are:

$b_{23}(x_2 = 1, x_3 = 1)$; $b_{23}(x_2 = 1, x_3 = 2)$; $b_{23}(x_2 = 1, x_3 = 3)$

$b_{23}(x_2 = 2, x_3 = 1)$; $b_{23}(x_2 = 2, x_3 = 2)$; $b_{23}(x_2 = 2, x_3 = 3)$

$b_{23}(x_2 = 3, x_3 = 1)$; $b_{23}(x_2 = 3, x_3 = 2)$; $b_{23}(x_2 = 3, x_3 = 3)$

We also need to normalize the $b_{ij}(x_i, x_j)$, for b_{12},the normalization constant is:

$$Z = 1/\sum_{x_i x_j} b_{12}(x_1, x_j)$$

Finally, after iterating for its times, we can get the final beliefs.

Part3: compute the partition function Z,

we know that the distribution is non-uniform:

$$p(x) = \frac{1}{Z}\prod_{i\in V}\phi(x_i)\prod_{(i,j)\in E}\psi(x_i,x_j)$$

Where $\phi(x_i) = e^{x_i}, \psi\left(x_i,x_j\right) = 1_{x_i \neq x_j}$,

That is: $p(x) = \frac{1}{Z}\prod_{i\in V} e^{x_i}\prod_{(i,j)\in E} 1_{x_i \neq x_j}$

$$D(q \parallel p) = -H(q) + \log Z - \sum_{x} q(x)\{\sum_{i\in V}\log\phi(x_i) + \sum_{(i,j)\in E}\log 1_{x_i \neq x_j}\}$$

When local maximization is achieved, we have:

$$\log Z = H(q) + \sum_{i\in V}\sum_{x_i} b_i(x_i)\log\phi(x_i) + \sum_{(i,j)\in E}\sum_{x_i x_j} b_{ij}(x_i,x_j)\log 1_{x_i \neq x_j}$$

$$H(q) = -\sum_{i\in V}\sum_{x_i} b_i(x_i)\log b_i(x_i) - \sum_{(i,j)\in E}\sum_{x_i x_j} b_{ij}(x_i,x_j)\log\frac{b_{ij}(x_i,x_j)}{b_i(x_i)b_j(x_j)}$$

We can get the value of Z.

The result of the matlab code for sumprod.m:

If input is a tree, the matrix is:

```
A=[0 1 0 0 1;
    1 0 1 1 0;
    0 1 0 0 0;     weight=[1,2,3]   its = 100
    0 1 0 0 0;
    1 0 0 0 0]
```

We can get $Z=$ sumprod(A,weight,its) = 2.3283e+06

If input is not a tree, the matrix is:

```
B=[0 1 1 1 1;
    1 0 1 1 1;
    1 1 0 1 1;     weight=[1,2,3]   its = 100
    1 1 1 0 1;
    1 1 1 1 0]
```

We can get $Z=$ sumprod(B,weight,its) = 2.0666e+04

If input is a tree with two nodes, the matrix is:

```
% weight1=[1,2];
C=[0 1;
   1 0];           weight=[1,2,3]   its = 100
```

We can get Z =sumprod(C,weight,100) =   446.1937

%conclude: if the graph is a tree, this code will provide the exact partition function value

If input is a tree with three nodes, the matrix is:

```
D=[0 1 1;
    1 0 0;         weight=[1,2,3]   its = 100
    1 0 0];
```

We can get Z =sumprod(D,weight,100) = 7.9462e+03

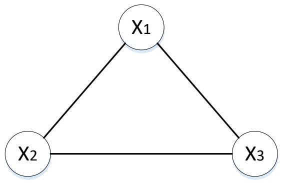
%conclude: if the graph is a tree, this code will provide the exact partition function value

Conclusion:

if the graph is a tree, this code will provide the exact partition function value;
if the graph is not a tree, this code only provides an approximation of the exact partition function value

2. Implement the sum-product algorithm for the coloring problem.

This is a detailed interpretation about the sum-product algorithm:



compute the all the messages.

We can get $\phi(x_i) = e^{x_i}$, $\psi(x_i, x_j) = 1_{x_i \neq x_j}$, $x_i \in \{1,2,3\}$ from the problem 1.

Initialize all the messages based on the graph above:

$$m_{1\to 2}^0(x_2) = m_{1\to 3}^0(x_3) = m_{2\to 1}^0(x_1) = m_{2\to 3}^0(x_3) = m_{3\to 1}^0(x_1) = m_{3\to 2}^0(x_2) = 1$$

Then:

$$m_{i\to j}^t(x_j) = \max_{x_i}\{\phi(x_i)\psi(x_i, x_j) \prod_{k\in N(i)\backslash j} m_{k\to i}^{t-1}(x_i)\}$$

For instance:

$$m_{1\to 2}^1(x_2 = 1) = \max_{x_i}\{\phi(x_1)\psi(x_1, x_2 = 1) \prod_{k\in N(i)\backslash j} m_{3\to 1}^0(x_1)\}$$

Where x_1 can be {1,2,3}, thus we can get the value of $m_{1\to 2}^1(x_2 = 1)$,

$$\max_{x_i}\{\phi(x_1 = 1)\psi(x_1 = 1, x_2 = 1)m_{3\to 1}^0(x_1 = 1), \phi(x_1 = 2)\psi(x_1 = 2, x_2 = 1)m_{3\to 1}^0(x_1 = 2), \quad \phi(x_1 = 3)\psi(x_1 = 3, x_2 = 1)m_{3\to 1}^0(x_1 = 3)\}$$

then we can get the other two values $m_{1\to 2}^1(x_2 = 2)$, $m_{1\to 2}^1(x_2 = 3)$.

Then we should normalize the message for avoiding overflow, the normalization constant can be computed by this formula:

$$\eta_{i\to j}^t = 1/\sum_{x_i}^{t} m_{i\to j}^t(x_j)$$

Then update the value of the message using the normalization constant, we can get all values about the message:

Similarly, we can get all the messages about the graph:

$m_{1\to 2}^1(x_2 = 1)$,$m_{1\to 2}^1(x_2 = 2)$,$m_{1\to 2}^1(x_2 = 3)$, $m_{1\to 3}^1(x_3 = 1)$, $m_{1\to 3}^1(x_2 = 2)$, $m_{1\to 3}^1(x_3 = 3)$;

$m_{2\to 1}^1(x_1 = 1)$,$m_{2\to 1}^1(x_1 = 2)$,$m_{2\to 1}^1(x_1 = 3)$, $m_{2\to 3}^1(x_3 = 1)$, $m_{2\to 3}^1(x_2 = 2)$, $m_{2\to 3}^1(x_3 = 3)$;

$m_{3\to 1}^1(x_1 = 1)$,$m_{3\to 1}^1(x_1 = 2)$,$m_{3\to 1}^1(x_1 = 3)$, $m_{3\to 2}^1(x_2 = 1)$, $m_{3\to 2}^1(x_2 = 2)$, $m_{3\to 2}^1(x_2 = 3)$;

Finally, iterate for its times, we will get the final values for all the messages.

The result of the matlab code for maxprod.m:

```
weight=[1,2,3];

% This matrix represents a tree graph with two nodes
% weight1=[1,2];
A=[0 1;
   1 0];
maxprod(A,weight,100) = 2     2
%conclude: it will output assignment that maximizes each singleton
belief


% This matrix represents a tree graph with three nodes
B=[0 1 1;
    1 0 0;
    1 0 0];
maxprod(B,weight,100)= 2     3     3
%conclude: it will output assignment that maximizes each singleton
belief


% This matrix represents a tree graph with five nodes
C=[0 1 0 0 1;
    1 0 1 1 0;
    0 1 0 0 0;
    0 1 0 0 0;
    1 0 0 0 0];

 maxprod(C,weight,100)= 3     2     3     3     2
 %conclude:it will output assignment that maximizes each singleton
belief

 % This matrix does not represent a tree graph
D=[0 1 1 1 1;
    1 0 1 1 1;
    1 1 0 1 1;
    1 1 1 0 1;
    1 1 1 1 0];

 maxprod(D,weight,100)= 2     2     2     2     2
```