University of Illinois Fall 2018

ECE 544NA: Pattern Recognition

Lecture 13: October 16th

Lecturer: Alexander Schwing Scribe: Dongwook Kim

1 Background

Inference program:

$$y* = \underset{\hat{y}}{\text{argmax}} \sum_r f_r(\hat{y_r})$$

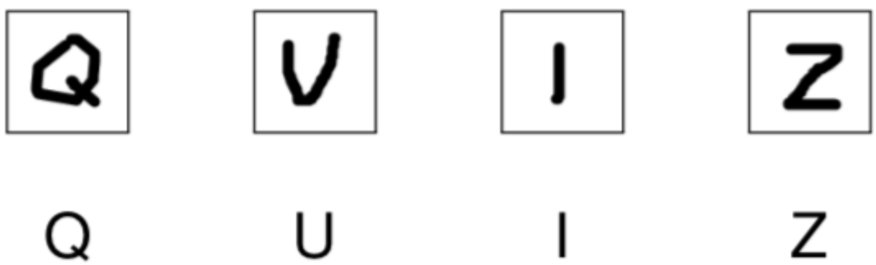


Figure 1: An inference example

Example:
$F(w, x, y_1, y_2, y_3, y_4) = f_1(w, x, y_1 + f_2(w, x, y_2) + f_3(w, x, y_3) + f_4(w, x, y_4) + f_{1,2}(w, x, y_1, y_2) + f_{2,3}(w, x, y_2, y_3) + f_{3,4}(w, x, y_3, y_4)$

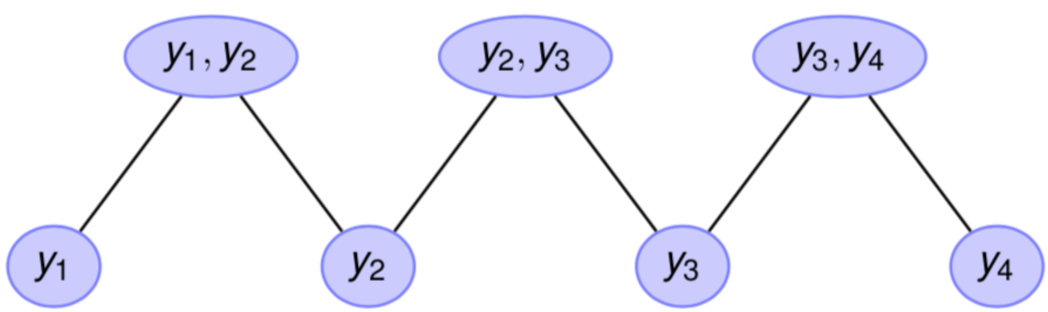


Figure 2: A visual decomposition of 'quiz' example

In the above figure, the edges denote subset relationships

2 Integer linear program

In this example, integere linear program maximize score function with constraints defined by binary in this case.

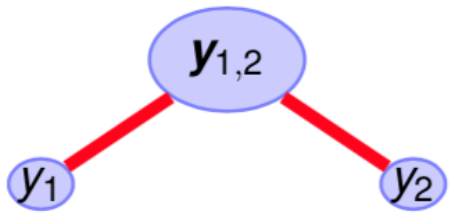


Figure 3: Visualization

$$\max_{b_1,b_2,b_{12}} \begin{bmatrix} b_1(1) \\ b_1(2) \\ b_2(1) \\ b_2(2) \\ b_{12}(1,1) \\ b_{12}(2,1) \\ b_{12}(1,2) \\ b_{12}(2,2) \end{bmatrix}^T \begin{bmatrix} f_1(1) \\ f_1(2) \\ f_2(1) \\ f_2(2) \\ f_{12}(1,1) \\ f_{12}(2,1) \\ f_{12}(1,2) \\ f_{12}(2,2) \end{bmatrix}$$

such that

$$b_r(y_r) = \in \{0,1\} \quad \forall r, y_r$$

$$\sum_{y_r} b_r(y_r) = 1 \quad \forall r$$

$$\sum_{y_p \backslash y_r} b_p(y_p) = b_r(y_r) \quad \forall r, y_r, p \in P(r)$$

In other words, integer linear program can be expressed like below.

$$\sum_{r,y_r} b_r(y_r) f_r(y_r)$$

such that

$$b_r(y_r) = \in \{0,1\} \quad \forall r, y_r$$

Local probability b_r
Marginalization

Advantage: global optimum, very good solvers available
Disadvantage: very slow for larger problems

3 Linear programming relaxation

In LP relaxation, there is no more integer constraint. Therefore, the first constaint about binary limitation introduced in the previous section can be ingored, therefore, allowing $b_r(y_r)$ can have any valude between 0 and 1. Due to this relaxation, LP relaxation has better score function which provides upper bound for the ILP case.

$$\sum_{r,y_r} b_r(y_r) f_r(y_r)$$

such that

$$\cancel{b_r(y_r) = \in \{0,1\} \quad \forall r, y_r}$$

Local probability b_r
Marginalization

For example, please see the below figure.

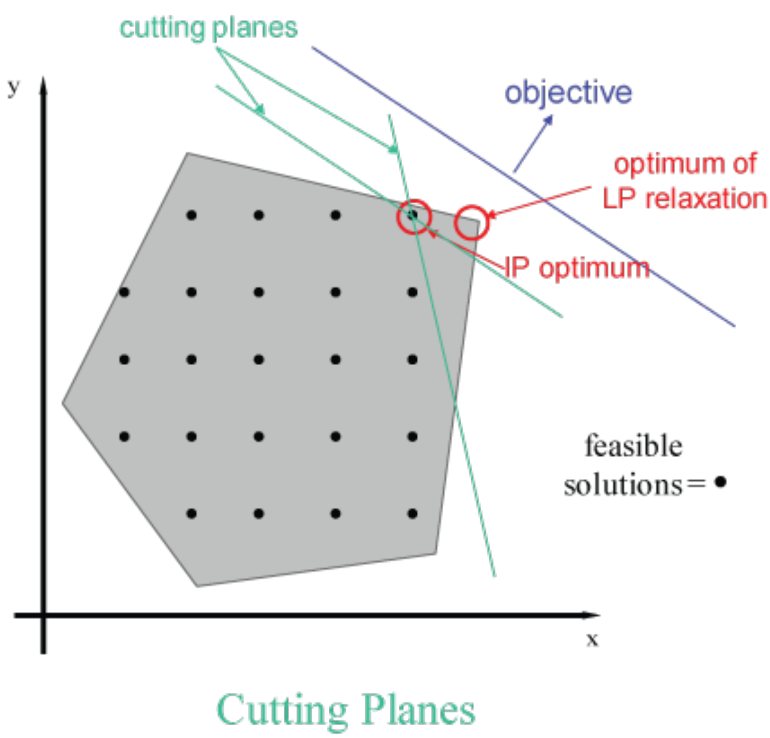


Figure 4: Illistration of ILP and LP relaxation. [3]

In the example, ILP optimum and LP optimum are differnet. In case of ILP optimum where x and y can take only integer, the optimal value cannot be better than LP optimal value where x and y can take any values.

Advantage: global optimum for LP, very good solvers available
Disadvantage: no global optimum for ILP, slow for larger problems

4 Message passing

4.1 Brief idea of belief propagation

Marginalization with partial sum with 'belief'.

$$\sum_{x_2,x_3,x_4} P(x_1,x_2,x_3,x_4) = P(x_1)\sum_{x_2} P(x_2|x_1)\sum_{x_3} P(x_3|x_2)\sum_{x_4} P(x_4|x_3)$$

[2]

A huge amount of calculations can be saved by partially summing up computation. For example, let's say each variables can have 10 values. Instead of performing 10x10x10=1000 times to marginalize, only 10+10+10=30 times can be done with partial sum. [2]

4.2 Brief idea of message passing

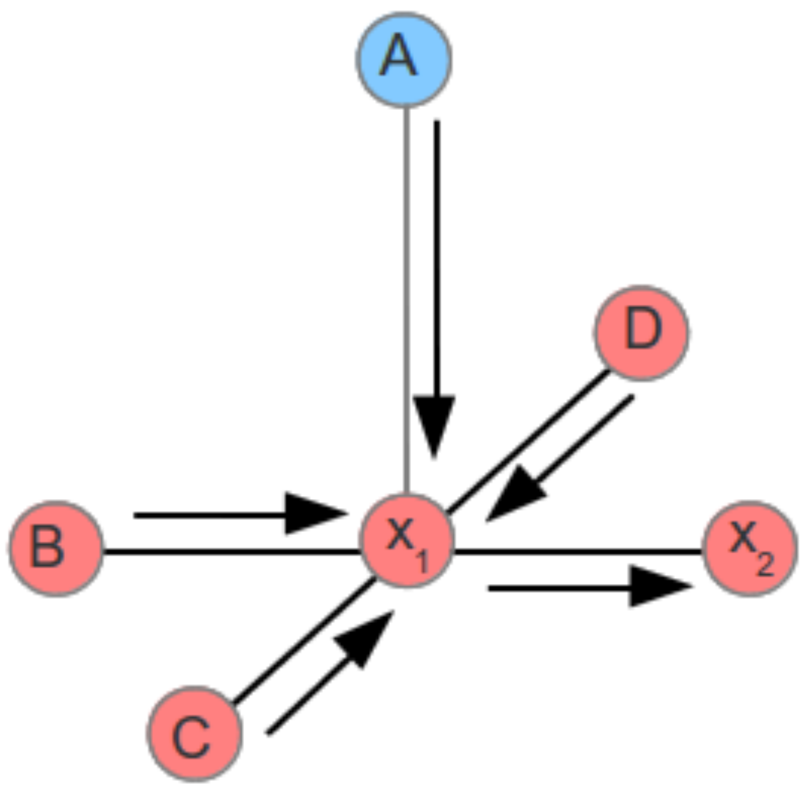


Figure 5: An example of LBP. [1]

Loopy belief propagation is message passing algorithm. For example, let's say x_i want to pass a message to x_j. Before it pass the message, it collects all the data related to j from adjacent nodes such as A, B, C, and D. It can be expressed like below.

$$msg_{i->j}(l)$$

[1] The expression means belief message of node i about node j regarding label l. [1]. Note that node i does not collect message from node j.

4.3 Detail of message passing

The inference program can be expressed like below maximization problem with given constraints as shwon below.

$$\max \sum_{r,y_r} b_r(y_r) f_r(y_r)$$

$$\text{such that}$$

$$b_r(y_r) \geq 0 \quad \forall r, y_r$$

$$\sum_{y_r} b_r(y_r) = 1 \quad \forall r$$

$$\sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r) \quad \forall r, y_r, p \in P(r)$$

By taking Lagrangian,

$$L = \sum_{r,y_r} b_r(y_r) f_r(y_r) + \sum_{r,y_r,p \in P(r)} \lambda_{r->p}(y_r) \Big(\sum_{y_p \setminus y_r} b_p(y_p) - b_r(y_r) \Big)$$

$$\sum_{r,y_r} b_r(y_r) \Big(f_r(y_r) - \sum_{p \in P(r)} \lambda_{r->p}(y_r) + \sum_{c \in C(r)} \lambda_{c->r}(y_c) \Big)$$

Now, the problem is reformulated to maximize Lagrangian with respect to primal vaviables subject to remaining contraints:

$$\max_b L()$$

$$\text{such that}$$

$$b_r(y_r) \geq 0 \quad \forall r, y_r$$

$$\sum_{y_r} b_r(y_r) = 1 \quad \forall r$$

Then, dual function is given as below.

$$g(\lambda) = \sum_r \max_{y_r} \Big(f_r(y_r) - \sum_{p \in P(r)} \lambda_{r->p}(y_r) + \sum_{c \in C(r)} \lambda_{c->r}(y_c) \Big)$$

Dual program is below.

$$\min_\lambda g(\lambda)$$

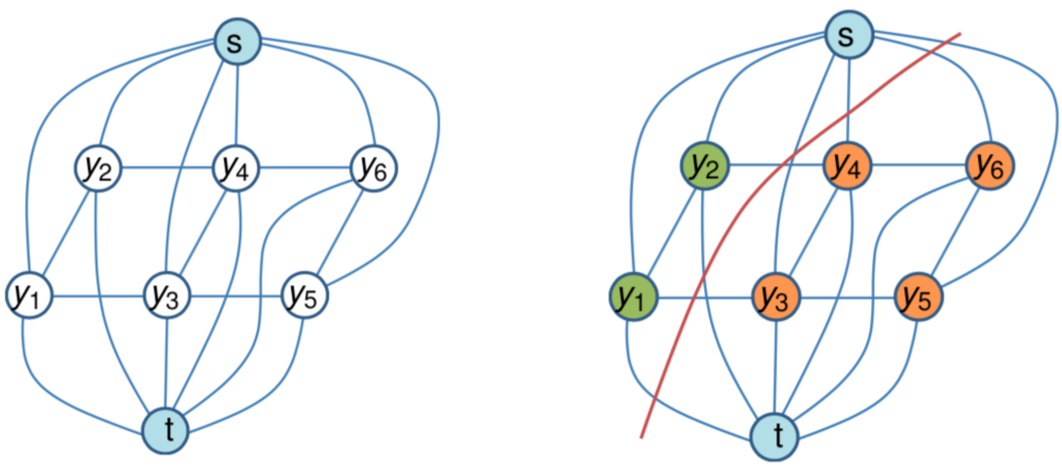
Advantage: Efficient due to analytically computable sub-problems
Disadvantage: Special care required to find LP relaxation optimum

Properties of dual program:

- Convex program
- Not strongly convex
- Unconstrained
- Lagrange multipliers are messages

5 Graph-cut

Graph cur solvers can be used to maximize maximum flow through a weighted graph as well as to minimize minimum cost cut in a weighted graph.
Example) Ford-Fulkerson algorithm



(a) (b)

Figure 6: (a) A graph (b) A graph-cut

In the graph, the top node is called source and the bottom node is called terminal. Each node represents variables y_d.

Scoring functions as arrays can be respresented as below.

$$\begin{bmatrix} f_1(y_1 = 1) & f_1(y_1 = 2) \end{bmatrix}$$

Let's express local scoring function arrays as below.

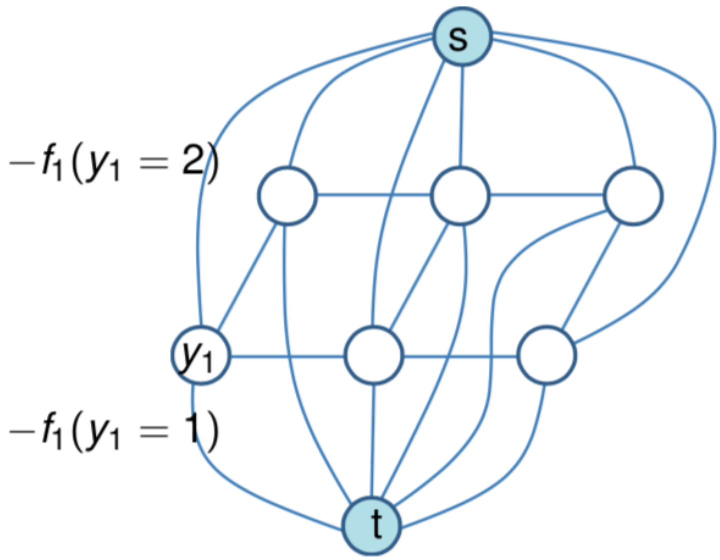


Figure 7: A graph example

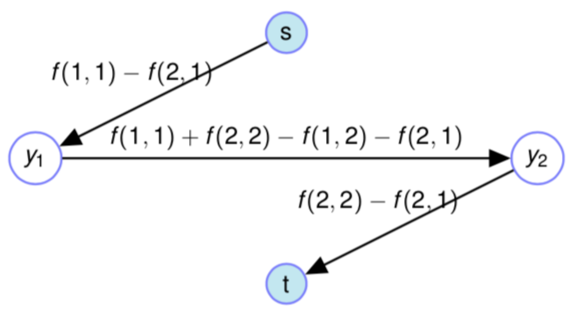


Figure 8: A graph example

$$\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix}$$

$$= \begin{bmatrix} f_{12}(1,1) - f_{12}(1,2) + f_{12}(2,2) & f_{12}(1,1) - f_{12}(1,2) + f_{12}(2,2) \\ f_{12}(1,1) - f_{12}(1,2) + f_{12}(2,2) & f_{12}(1,1) - f_{12}(1,2) + f_{12}(2,2) \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 \\ f_{12}(2,1) - f_{12}(1,1) & f_{12}(2,1) - f_{12}(1,1) \end{bmatrix}$$

$$+ \begin{bmatrix} f_{12}(2,1) - f_{12}(2,2) & 0 \\ f_{12}(2,1) - f_{12}(2,2) & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & f_{12}(1,2) + f_{12}(2,1) - f_{12}(1,1) - f_{12}(2,2) \\ 0 & 0 \end{bmatrix}$$

Requirements for optimality:

$$f_{12}(1,1) + f_{12}(2,2) - f_{12}(1,2) - f_{12}(2,1) \geq 0$$

References

[1] Loopy belief propagation, markov random field, stereo vision, http://nghiaho.com/page_id=1366.

[2] Markov random fields, graph cuts, belief propagation, https://slideplayer.com/slide/6409764/.

[3] Mixed-integer programming (mip), http://www.gurobi.com/resources/getting-started/mip-basics.