Week 10

Lecture-19& Lecture 20 : Graph Cuts and Network flow graphs -Max flow Min cut*

The class notes contains Graph cut, setting up flow network , describing max-flow, cuts in network, residual networks, max-flow min cut theorem and Max flow algorithm(ford and fulkerson)

10.1 Graph cut-Pseudo Boolean function

$f : 0, 1- > R$

$$f(x) = 3x_1 + 4\overline{x_1} + 5x_2 + 3\overline{x_2} + 6\overline{x_1}x_2$$
$$x_1, x_2 belongs 0, 1$$

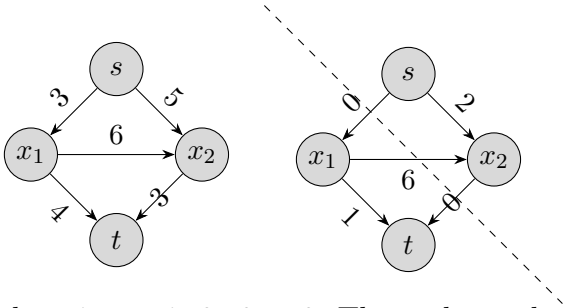
Goal min f(x). By brute force method assign values to x1 and x2 using a truth table and find the value of f(x). The complexity of this method is 2^n.

10.1.1 Graph cut

Construction of the graph for the above equation. (1) Every cut of the graph corresponds to some assignment of the variables. (2) min cut = min cost assignment

For the polynomial $f(x) = 3x_1 + 4\overline{x_1} + 5x_2 + 3\overline{x_2} + 6\overline{x_1}x_2$
The following graph is constructed.It is to note that $\overline{x_1}x_2$ or $\overline{x_2}x_1$ is allowed and not x_1x_2. The bar will give the direction of the arc in the graph. Now the flow graph can be worked out using min flow and the following graph results. tikz

*Lecturer: Anand Mishra . Scribe: Kalpana Vankayala MT19AIE250.

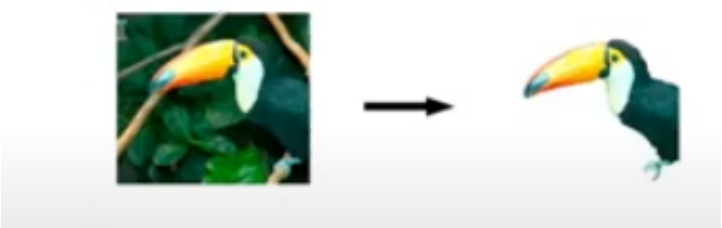


So the min cut is 3+3 = 6. The node on the source side is taken as zero i.e x_2= 0 and node on the target side x_1= 1. Substituting in f(x) we get f(x) = 6. which is the minimum of the function given above.

10.1.2 Applications of Graph cut

Image segmentation in computer vision applications:

In image segmentation problem we have to separate into foreground and background pixels of the given image by minimising a cost /energy function. There is an energy minimisation



function and we need to find global minima of the energy function.

$$E(x) = \sum_i c_i x_i + \sum_{i,j} c_{i,j} x_i (1 - x_j)$$

The matrix will contain the pixel values of the image. consider a 3X3 matrix as follows.

$$\begin{bmatrix} 250 & 240 & 255 \\ 5 & 230 & 9 \\ 6 & 235 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

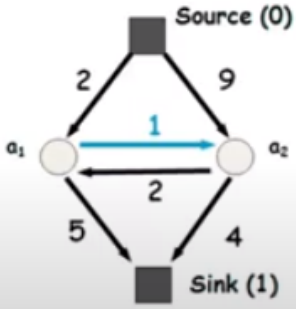
Let the range of the pixels [0.255] $x_1, x_2, x_3.....x_9$ $p(x_1 = 250$ The function can be formulated as.. $(255 - p(x_1))x_1 + p(x_1)\overline{x_1}$

$$f(x) = \sum_{i=1}^{9} (255 - p(x_i))x_i + p(x_i)\overline{x_i} \tag{10.1}$$

$$f(x) = \sum_{i,j}(c_{ij}x_i\overline{x_j} + c_{ji}\overline{x_i}x_j) \tag{10.2}$$

Now the problem has been formulated as above and hence we apply graph cut and find value ofx_ix_j and we can get the equivalent or segmented matrix as output.
Therefore, any cut corresponds to assignment of x and cost of cut is equal to energy or function of x.
Consider the following example for graph construction and formulation.

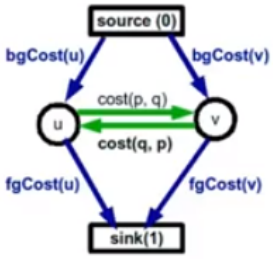
$$E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



10.1.3 Implementation issues:

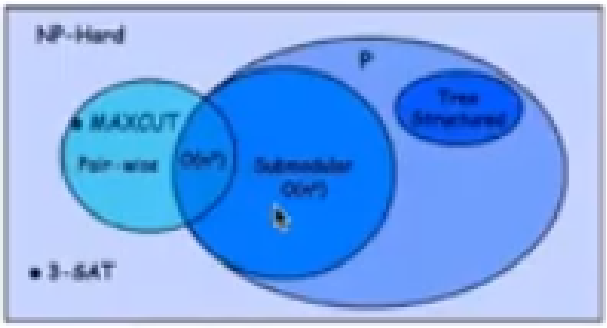
The demo code for the implementation is shown in the below .

```
graph *g

for all pixels p
      /* Add a node */
  node p = g->add_node()
  g -> add_tweights(p, fgCost(p), bgCost(p))
end

for all adjacent pixels p, q
   add_edge(p, cost(p,q), cost(q,p))
end

flow = g -> maxflow

for all pixels p
  Print  g -> what_segment(p)
end;
```



10.1.4 Types of Optimisation function supported by Graph cut

General energy functions: NP hard to solve can be approximated.
Easy energy functions: (sub modular functions) Graph representable and solvable in polynomial time.

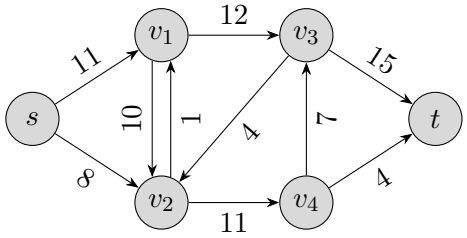


What are submodular functions?
Let f be a function defined over set of boolean variables.$X = (x_1, x-2, ...x_n$ then
All functions with one boolean variable is submodular
A function with two boolean variable is submodular if $f(0,0) + f(1,1) <= f(0,1) + f(1,0)$
In general a function is submodular if all its projection to two variables is submodular.

10.2 Flow Networks- min cut max flow algorithm

10.2.1 Flow Networks

Flow networks are graphs G= (V,E) set of vertices, edges. These are directed graphs. The two distinguished vertices are the source(s) and the sink(t). Each edge is defined by (u,v) directed from u to v with non negative capacity c(u,v) and c(u,v) is zero if there is no road between u and v.



Consider the following flow network.

Notation say 1:3 (1 represents f:flow, 3 represents c: capacity). All of the flow entering the node has to leave the network node. With this constraint we have to find the max flow. The flow value is give by $f = |f| = f(s, V) = f(V, t)$.

10.2.2 cut

A cut is any partition (S,T) where s belongs to S and t belongs to T. Lemma: —f— = f(S,T) for any cut (S,T) i.e across any cut we can see the flow. Corollary: The —f—¡= c(S,T) for any cut(S,T) Therefore, given any cut we have a bound on the flow value and hence we calculate maximum flow for min cut.

10.2.3 Residual graph

Residual graph: This is a new graph Gf=(V,Ef) with strictly positive residual capacities. cf(u,v) = c(u,v)-f(u,v) ¿0

10.2.4 Augmenting path

Augmenting path: Any path from s to t in Gf which doesnt have max flow but we can increase flow. And the value by which we can increase the flow is residual capacity.

Residual capacity is the minimum capacity cf(P) = min cf(u,v) where u,v belong to P. P is the augmenting path.

10.2.5 Ford Fulkerson Algorithm for max flow:

Step1: f(u,v) ¡- 0,
Step2:
While and augumenting path in Gf exists.(We do a breadth first or depth first search to find the augmenting path)
Step3:
do augment f by cf(P)

So, if the above algorithm terminates or converges we get the max flow. lets prove it.
Proof: Max-flow min cut theorem The following are equivalent.
1)—f— = c(S,T) for some cut (S,T) (Which implies some cut is saturated)
2) f is the maximum flow
3) f admits no augmenting paths
We need to show that (3) implies (2) by showing that (1)implies(2), (2)implies(3) and (3)implies(1)

Proof. (1) implies (2)
Since f¡= c(S,T) (edge capacity constraints) For any cut c(S,T) , the assumption that —f—= c(S,T) implies f is max flow and f cant be increased.
(2) implies (3)
proof by contradiction:
If there were an augmenting path, the flow value could be increased by some amount contradicting maximality of f. As the residual capacity is greater than zero which is the min of that augmenting path.
(3) implies(1)

□

Suppose f admits no augmenting paths. That means t is not reachable from s.

Define S=u belongs to V, there exists a path in Gf from s to u
Here all the vertices that are reachable from s are collected into a set S. T= V minus S

but s belongs to S and t belongs to T.
Then (S,T) is a cut. Now the vertex u in S and v in T for a residual graph Gf. Here cf(u,v)=0 since if cf(u,v) ¿ 0 then u belongs to S not v belongs to T as assumed.
Thus f(u,v) = c(u,v)
cf(u,v) = c(u,v) -f(u,v) =0
Therefore there is no edge in the residual graph as every edge is saturated and all capacity is used.
Summing over u belongs to S and v belongs to T yields f(S,T) = c(S,T)

10.2.6 Complexity analysis of Ford Fulkerson algorithm:

Selecting an augmenting path and the best one to bad one analysis.
The mitigation was thought by Edmond's-karp who did a small change to the Ford fulkerson also known as Edmonds karp algorithm.

Which says, if breadth first search for an augmenting path will give the lowest complexity to the ford fulkerson algorithm.
The breadth first augmenting path is the shortest path in Gf from s to t where each edge has a weight 1.
The complexity is of the order O(VE) in the worst case if we use breadth first path.
The overall complexity is order O(VE2)as breadth first algorithm has O(E) time . The other algorithms are being worked to lower this bound and the fastest came in 2011 by King, Rao and Tarjan.
Orlin came up with O(VE) algorithm using fast matrix multiply.

Application of network flow can be used in Base Ball team Eliminations .

End of Class