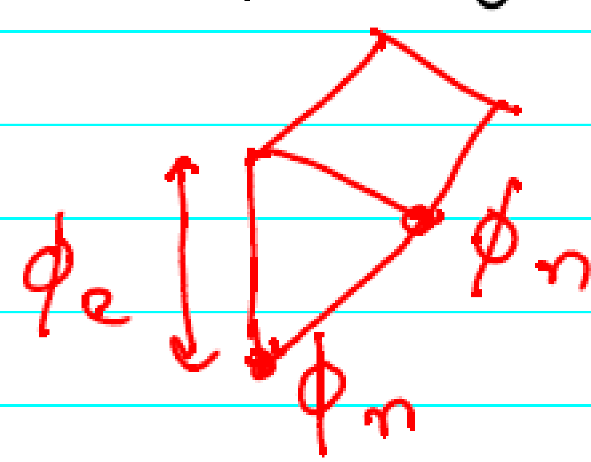
Recap from last lecture how we tried to pose inference in Markov network as an (Quadratic programming) optimisation problem

Integer linear program formulation for inference on max margin markov network

$$\hat{y} = \arg\max_{y} \; w^T \phi(x^i, y) + \ell(y^i, y)$$



$z_{n\ell}$ = A 0/1 variable associated with node n & label ℓ

$y = \begin{bmatrix} \ell_1 \\ \ell_2 \\ \ell_n \end{bmatrix} \rightarrow$ if $y(i) = \ell_i$ then $z_{i\ell_i} = 1$

$$\hat{z} = \arg\max_z \sum_n \sum_{\ell_n} z_{n\ell_n} \left(\sum_j w_{nj} \phi_{nj}(x, \ell) \right)$$

node features

$$+ \sum_n \sum_{\ell_n} \sum_{n'} \sum_{\ell_{n'}} z_{n\ell_n} z_{n'\ell_{n'}} \cdot \sum_k w_k \phi_{e,k}(\ell_n, \ell_{n'})$$

makes it a QP

set of edge features

s.t. $\sum_{\ell_n} z_{n\ell_n} = 1 \ \forall n$ & $z_{n,\ell_n} \in \{0, 1\}$

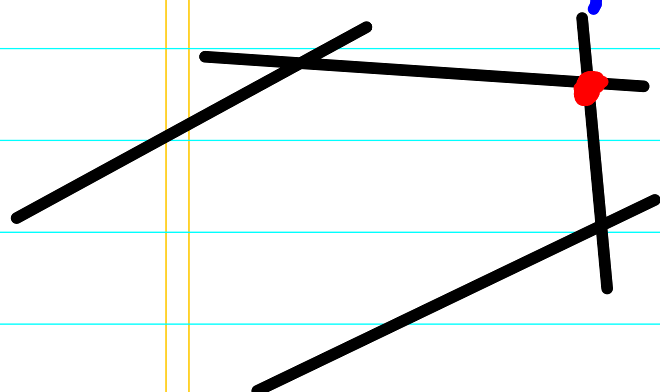
H/W: Replace the QP with a LP?

Soln: Replace with $z_{n\ell_n n'\ell_{n'}}$ & add constraints to make $z_{n\ell_n n'\ell_{n'}}$ with $z_{n\ell_n}$ & $z_{n'\ell_{n'}}$

PRECISE IDEA BEHIND JUNCTION TREES/FACTOR GRAPHS

Some notes on integer Linear programming

A] Linear programming (LP):



This set of pts x is called a polyhedron P

$$\max_x \; c^T x$$

s.t $Ax \le b$ & $x \in \mathbb{R}^n$

$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$ $\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$

$\boxed{\text{Claim:}}$ If the LP is bounded and rank$(A) = n$, $\exists$ an optimal solution that is also a vertex

$x^* \in P$ is a vertex

① iff there exists a subset of the inequalities of $A_I x \le b_I$ s.t rank$(A_I) = n$

② iff $\exists$ a $g \in \mathbb{R}^n$ s.t

$$x^* = \underset{x}{\operatorname{argmax}} \; g^T x$$

s.t $Ax \le b$ & $x \in \mathbb{R}^n$

③ iff $\exists$ a set of indices $B = \{l_1, l_2 \ldots l_n\}$ (i.e $|B| = n$) s.t A_B (the matrix consisting of the subset of A with all columns & only rows indexed by B) is non-singular and

$$\begin{pmatrix} a_{e_1}^T \\ a_{e_2}^T \\ \vdots \\ a_{e_n}^T \end{pmatrix} \leftarrow A_B x^* = b_B \rightarrow \begin{pmatrix} b_{e_1} \\ b_{e_2} \\ \vdots \\ b_{e_n} \end{pmatrix}$$

(i.e $x^* = A_B^{-1} b_B$)

$(\& A_{B^c} x^* \leq b_{B^c})$

Soln must be feasible

Options for solving:

① $\boxed{\text{Brute force}}$: Enumerate all vertices $\Rightarrow O\left({}^mC_n\right)$

② $\boxed{\text{Simplex algo}}$: Based on traversing from one vertex to its "most promising" adjacent vertex

$\boxed{\text{ie } c^T x' \geq c^T x}$

All in P

Say $x \to x'$

> 2 vertices $x_1 \in P$ & $x_2 \in P$ are adjacent if there exist $n-1$ linearly independent inequalities in $Ax \leq b$ that are active at both x_1 & x_2

Active means these inequalities are equalities

(Recall from earlier discussion on Lagrange multipliers)

OR equivalently

> 2 vertices $x_1 \in P$ & $x_2 \in P$ $(x_1 \neq x_2)$ are adjacent iff $\exists\, q \in \mathbb{R}^n$ s.t the set of optimal solns of
>
> $$\underset{Ax \leq b,\ x \in \mathbb{R}^n}{\operatorname{argmax}}\ q^T x \text{ is } \{\underbrace{\lambda x_1 + (1-\lambda) x_2}_{\text{Convex combination, ie line segment joining } x_1 \text{ to } x_2} \mid 0 \leq \lambda \leq 1\}$$

Q: How to move from one vertex to another?

Ans: Deriving the KKT necessary conditions

① $Ax^* \leq b$ ③ $\lambda \geq 0$

② $\nabla_x\left(-\lambda^T(Ax^* - b) + c^T x^*\right) = 0$ ④ $\lambda^T(Ax^* - b) = 0$

If B = (basis) set of indices relevant to a vertex st rank$(A_B) = n$, then the necessary conditions in ② state that $\lambda^T A = c^T$ & from ④, $\lambda_i = 0$ if $i \notin B$ & $\lambda_i \geq 0$ if $i \in B$, where $\lambda \in \mathbb{R}^m$. Also $x = A_B^{-1} b_B$.

The Simplex idea: At any vertex x with index set (also called Basis) B, compute λ and find $\lambda_i < 0$

$\lambda_B^T = c^T A_B^{-1}$

The idea is to remove i from B and add the next index k that would give an x' such that $c^T x' \geq c^T x$

In fact the dual linear program is:

$$\boxed{\begin{aligned} &\min_{\lambda} \; b^T\lambda \\ &\text{s.t. } A^T\lambda = c \\ &\lambda \geq 0 \end{aligned}}$$

DUAL LP

Let λ^* be solution

$\boxed{\textbf{STRONG DUALITY THM}}$: If the Primal LP is feasible & bounded then dual is also feasible & bounded 1

and $c^T x^* = b^T \lambda^*$

And infact, the dual of the dual is the primal LP (You can derive this by expressing $A^T\lambda = c$ as $A^T\lambda \leq c$ & $-A^T\lambda \leq -c$)

Further, if the primal is unbounded, the dual will be infeasible 2 AND

if the dual is unbounded, the primal will be infeasible 3

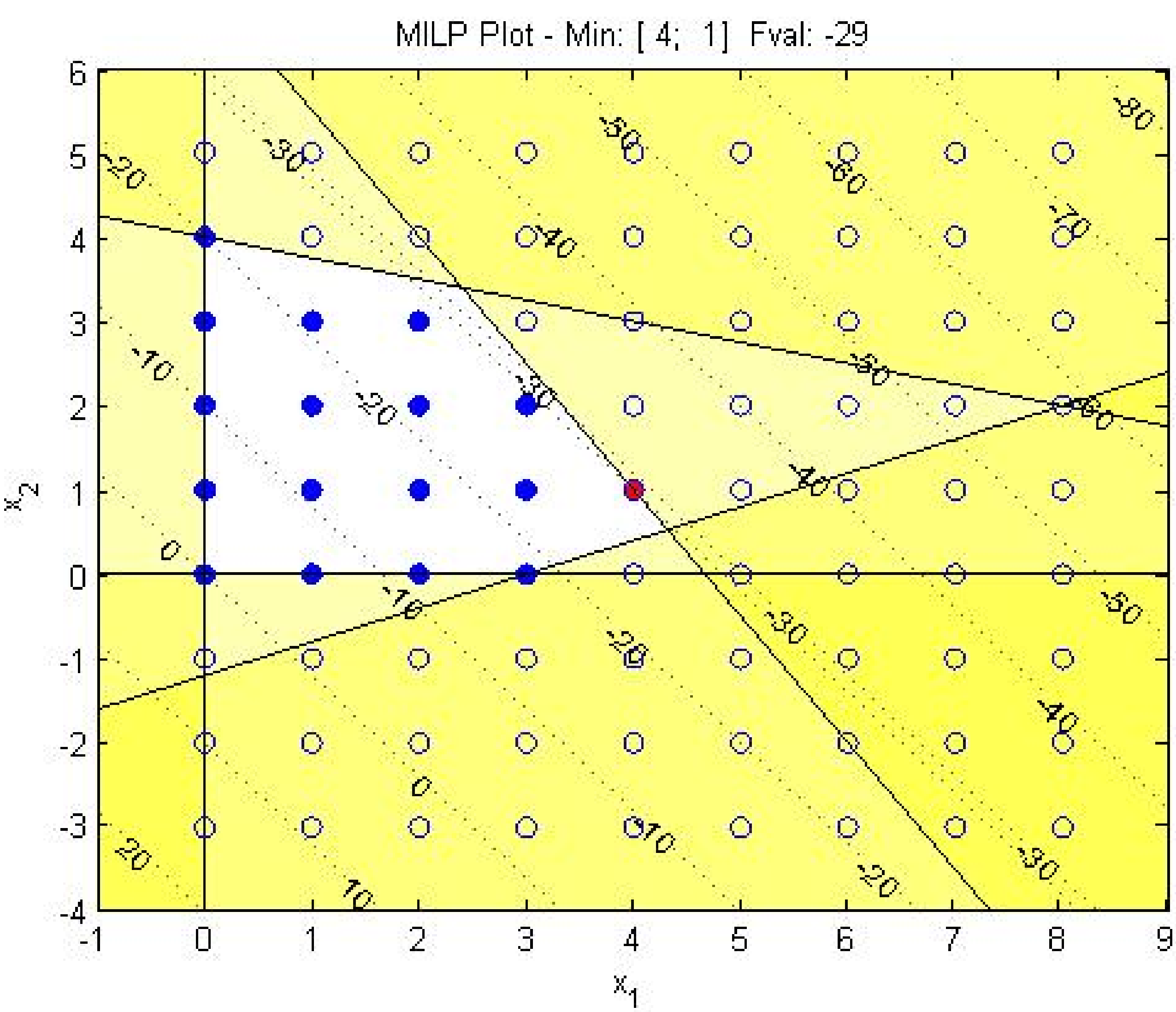
In fact, the $\boxed{\text{Farkas Lemma}}$ (obtained using duality analysis) states that $Ax \leq b$ is infeasible if and only if $\exists\, \lambda \geq 0$ s.t $\lambda^T A = 0$ & $\lambda^T b = -1$

Integer linear program:

$$\max \; c^T x$$
$$\text{s.t } Ax \le b$$
$$x \in \mathbb{Z}^n$$

(n dimensional space of integers)

Replacing $x \in \mathbb{R}^n$ with $x \in \mathbb{Z}^n$



eg:

$$\min_{x_1, x_2} \; -6x_1 - 5x_2$$

s.t: $x_1 + 4x_2 \le 16$

$6x_1 + 4x_2 \le 28$

$2x_1 - 5x_2 \le 6$

$0 \le x_1 \le 10$

$0 \le x_2 \le 10$

$x_1, x_2 \in \mathbb{Z}$

Unlike LP, feasible pts may not lie on constraints

Recall from quiz 1

3. The optimization problem in (1) is an *Integer Linear Program* (ILP).

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ & x_i \in \{0,1\} \end{array} \tag{1}$$

where $\mathbf{x} \in \Re^n$, $A \in \Re^{m\times n}$, $\mathbf{b} \in \Re^m$ and $\mathbf{c} \in \Re^n$. In a general method called relaxation, the constraint that x_i be zero or one is replaced with the linear inequalities $0 \leq x_i \leq 1$. The problem in (2) is called the *Relaxation of the Linear Program* (RLP).

Note: $\{x_i \mid x_i \in \{0,1\}\} \subseteq \{x_i \mid 0 \leq x_i \leq 1\}$

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ & 0 \leq x_i \leq 1 \end{array} \tag{2}$$

It turns out that the RLP (2) is far easier to solve than the original ILP (1).

Soln to 2 $\leq$ Soln to 1

Weak duality for ILP

$$\boxed{\begin{array}{l}\max_x \; c^Tx \\ Ax \leq b,\ x \in \mathbb{Z}^n\end{array}} \leq \boxed{\begin{array}{l}\max_x \; c^Tx \\ Ax \leq b,\ x \in \mathbb{R}^n\end{array}} = \boxed{\begin{array}{l}\min_\lambda \; b^T\lambda \\ A^T\lambda = c \\ \lambda \geq 0,\ \lambda \in \mathbb{R}^n\end{array}} \leq \boxed{\begin{array}{l}\min_\lambda \; b^T\lambda \\ A^T\lambda = c \\ \lambda \geq 0,\ \lambda \in \mathbb{Z}^m\end{array}}$$

Q: Are there conditions under which the inequalities (esp the first one) become equalities?

Ans: Yes!

A matrix $A \in \{0, \pm 1\}^{m \times n}$ is totally unimodular if the determinant of each square submatrix of A equals ± 1

Eg: $\begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$, $\begin{bmatrix} -1 & 1 & 0 & 1 \\ 0 & 0 & -1 & -1 \\ 1 & -1 & 1 & 0 \end{bmatrix}$

The following matrix is NOT totally unimodular:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Q: How to prove that a matrix is totally unimodular? Eg: Matrix A for a binary Markov n/w AND matrix

A for a triangulated Markov N/w with any # of labels: Both are totally unimodular

Ans: By induction on size of the problem. [H/W]

Markov Net Inference LP

same as $c^T x$

$$\max_{z} \sum_{j,m} z_j(m)\left[\mathbf{w}_N^\top \mathbf{f}_N(\mathbf{x}_j, m) + \ell_j(m)\right] + \sum_{jk,m,n} z_{jk}(m,n)\left[\mathbf{w}_E^\top \mathbf{f}_E(\mathbf{x}_{jk}, m, n) + \ell_{jk}(m,n)\right] \quad \Big\}\ \mathbf{q}^\top \mathbf{z}$$

$$\mathbf{q} = \mathbf{F}^\top \mathbf{w} + \ell$$

$z_k(n)$

0	1	0	0

$z_j(m)$

0
0
1
0

0	0	0	0
0	0	0	0
0	1	0	0
0	0	0	0

$z_{jk}(m,n)$

$z_j(m) > 0; \qquad z_{jk}(m,n) \geq 0;$

normalization $\sum_m z_j(m) = 1$

agreement $\sum_n z_{jk}(m,n) = z_j(m)$

CLAIM: Total unimodularity of A attained with triangulation of graph

$A\mathbf{z} = \mathbf{b}$

same as $Ax \leq b$

Has integral solutions **z** for chains, (hyper)trees

Can be fractional for untriangulated networks

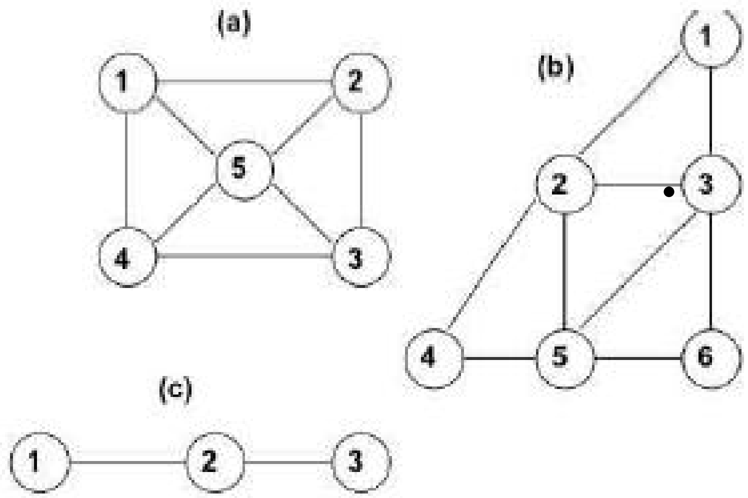
[Chekuri+al 01, Wainright+al 02]

Triangulated network/graph: defn 11 from

~cs717/notes/classNotes/graphicalModelsReading.pdf

Definition 11 *A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge. A graph is triangulated it is has no chordless cycles.*

Exercise: identify triangulated and non-triangulated graphs from below. Source: onlinecourses.science.psu.edu/stat504/node/184



- (b) and (c) are triangulated.
- (a) is not: [125][145][345][235] . (eliminating variable 5 yields 4-cycle)

In (a), to make it triangulated, what additional edges will you need? Ans:

Note: We can add edges to a graphical model without losing info (by defining larger clique potentials in terms of edges/nodes as before). Same does not hold when you delete edges!

Q: So what if in an LP, the matrix A is totally unimodular?

Theorem: If $A \in \mathbb{Z}^{m \times n}$ is totally unimodular and $b \in \mathbb{Z}^{m}$, then every vertex of the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \le b, x \ge 0\}$ is integral

Proof: We have that $\begin{pmatrix} A \\ -I \end{pmatrix} x \le \begin{pmatrix} b \\ 0 \end{pmatrix}$

Let $B \subseteq \{1, 2 \ldots m+n\}$ be the basis (set of indices) at some vertex x.

Let $B = B_1 \cup B_2$ s.t B_1 & B_2 are disjoint and $B_1 \subseteq \{1, \ldots m\}$ & $B_2 \subseteq \{m+1 \ldots m+n\}$

Then ① If $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ then $x(B_2 - m) = 0$

That is, all components of x indexed by set B_2 (modulo the offset of m) for which the inequality $-Ix \le 0$ is active will need to be 0.

(2) Let B_1 have k components, i.e $|B_1| = k \geq n$ and let A_1 be the submatrix of A having the same columns but only a subset of rows indexed by B_1. Let b_1 be a similar submatrix of b

$\Rightarrow A_1$ will have size of $k \times n$ & b_1 will have k rows and

$$A_1 x = b_1$$

$\Rightarrow$ Further since $x(B_2) = 0$, we could restrict our attention to the k columns of A that are NOT indexed by B_2. Let $\tilde{A}$ be the corresponding submatrix of A_1

$$\Rightarrow \tilde{A}\tilde{x} = b_1 \Rightarrow \tilde{x} = (\tilde{A})^{-1} b_1$$

$\Rightarrow$ By Cramer's Rule (page 202 of http://www.cse.iitb.ac.in/~cs717/notes/classNotes/extendedNotes.pdf),

$$(\tilde{A})^{-1} = \frac{1}{\det(\tilde{A})} * \operatorname{adj}(\tilde{A})$$

It is ± 1 because of total unimodularity of A where, $\operatorname{adj}(\tilde{A}) = \begin{bmatrix} \det(\tilde{A}_{11}) & -\det(\tilde{A}_{21}) & \cdots \\ -\det(\tilde{A}_{12}) & \det(\tilde{A}_{22}) & \cdots \end{bmatrix}$

Each is ± 1 because of total unimodularity of A

$\therefore \tilde{x} = (\tilde{A})^{-1} \tilde{b}$

$\in \mathbb{Z}^k$ (since $(\tilde{A})^{-1}$ as well as $\tilde{b}$ are integral)

In summary, $x \in \mathbb{Z}^n$ & $\therefore$ every vertex of P is integral (if A is totally unimodular)

COROLLARY: If A is totally unimodular, the soln to

$$\max \; c^T x$$
$$Ax \le b$$
$$x \in \mathbb{R}^n$$

is integral & therefore

$\boxed{\max_x c^T x \quad Ax \le b, x \in \mathbb{Z}^n} = \boxed{\max_x c^T x \quad Ax \le b, x \in \mathbb{R}^n} = \boxed{\min_\lambda b^T \lambda \quad A^T\lambda = c \quad \lambda \ge 0, \lambda \in \mathbb{R}^n} = \boxed{\min_\lambda b^T \lambda \quad A^T\lambda = c \quad \lambda \ge 0 \quad x \in \mathbb{R}^n}$

Putting together the story:

Markov Net Inference LP

$$\left.\begin{aligned}\max_{\mathbf{z}} \quad & \sum_{j,m} z_j(m)\left[\mathbf{w}_N^T \mathbf{f}_N(\mathbf{x}_j, m) + \ell_j(m)\right] \\ + & \sum_{jk,m,n} z_{jk}(m,n)\left[\mathbf{w}_E^T \mathbf{f}_E(\mathbf{x}_{jk}, m, n) + \ell_{jk}(m,n)\right]\end{aligned}\right\} \begin{aligned}&\mathbf{q}^T\mathbf{z} \\ &\mathbf{q} = \mathbf{F}^T\mathbf{w} + \ell\end{aligned}$$

$z_k(n)$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$z_j(m)$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$z_{jk}(m,n)$

$$z_j(m) \ge 0; \qquad z_{jk}(m,n) \ge 0;$$

$$\left.\begin{aligned} &\text{normalization} \quad \sum_m z_j(m) = 1 \\ &\text{agreement} \quad \sum_k z_{jk}(m,n) = z_j(m) \end{aligned}\right\} A\mathbf{z} = \mathbf{b}$$

If graph is triangulated, A is totally unimodular and soln to relaxed prob is integral

Has integral solutions **z** for chains, (hyper)trees

Can be fractional for untriangulated networks

[Chekuri+al 01, Wainright+al 02]

For the case with binary labels the problem can be rewritten as

$$\max_Z \sum_j Z_j\left[w^T f_N(x_j)\right] + \sum_{i,j} Z_{ij}\left[w^T f_E(x_i, x_j)\right]$$

(Note: loss components $\ell_j(m)$ etc. not reqd)

$$Z_j \ge 0 \quad \& \quad \sum_k Z_{jk} = Z_j$$ [agreement]

[Associative Markov Networks]

CLAIM: If $w^T f_E(0,0) + w^T f_E(1,1) \geq w^T f_E(0,1) + w^T f_E(1,0)$,

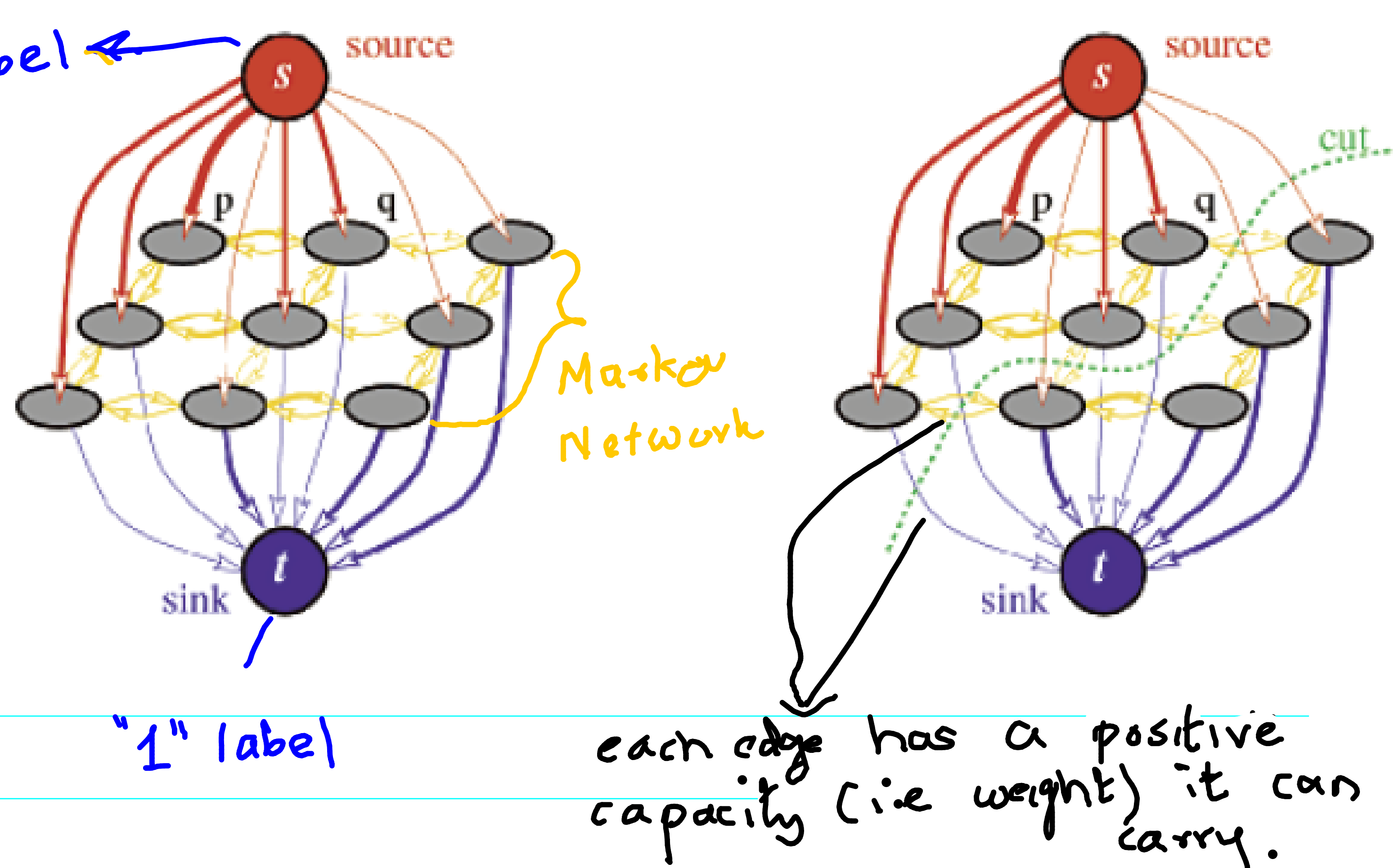
(x_i, x_j) are common arguments — also called regularity

then we can represent the MAP inference problem as a graph max-flow (or min-cut) problem

http://www.cs.cornell.edu/~ rdz/papers/kz-pami04.pdf

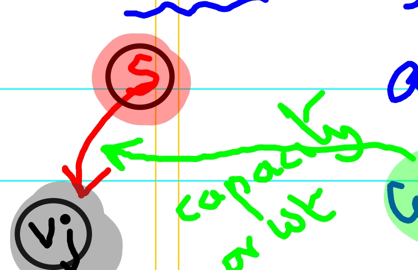
http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem

PROOF BY CONSTRUCTION [sufficiency part]



Case 1: If $w_N^T f_N(x_j, 0) < w_N^T f_N(x_j, 1)$ then add edge (s, v_j) with weight

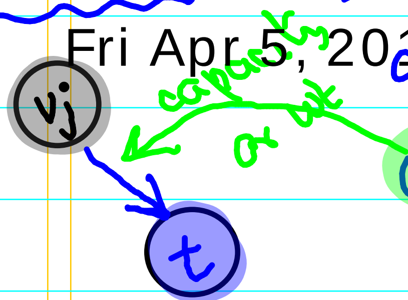
$w_N^T f_N(x_j, 1) - w_N^T f_N(x_j, 0) > 0$



Fri Apr 5, 2013 1:36 PM

Case 2: if $w_N^T f_N(x_j,0) \geq w_N^T f_N(x_j,1)$ then

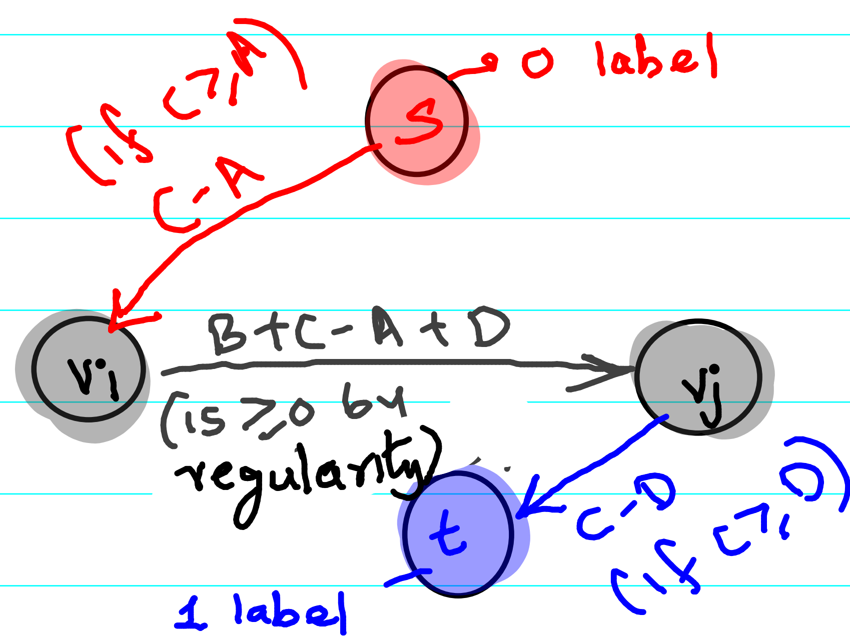
add edge (v_j,t) with weight $w_N^T f_N(x_j,0) - w_N^T f_N(x_j,1) \geq 0$



Now:

$w_E^T f_E(x_i,x_j,0,0) \to A$	$w_E^T f_E(x_i,x_j,0,1) \to C$
$w_E^T f_E(x_i,x_j,1,0) \to B$	$w_E^T f_E(x_i,x_j,1,1) \to D$

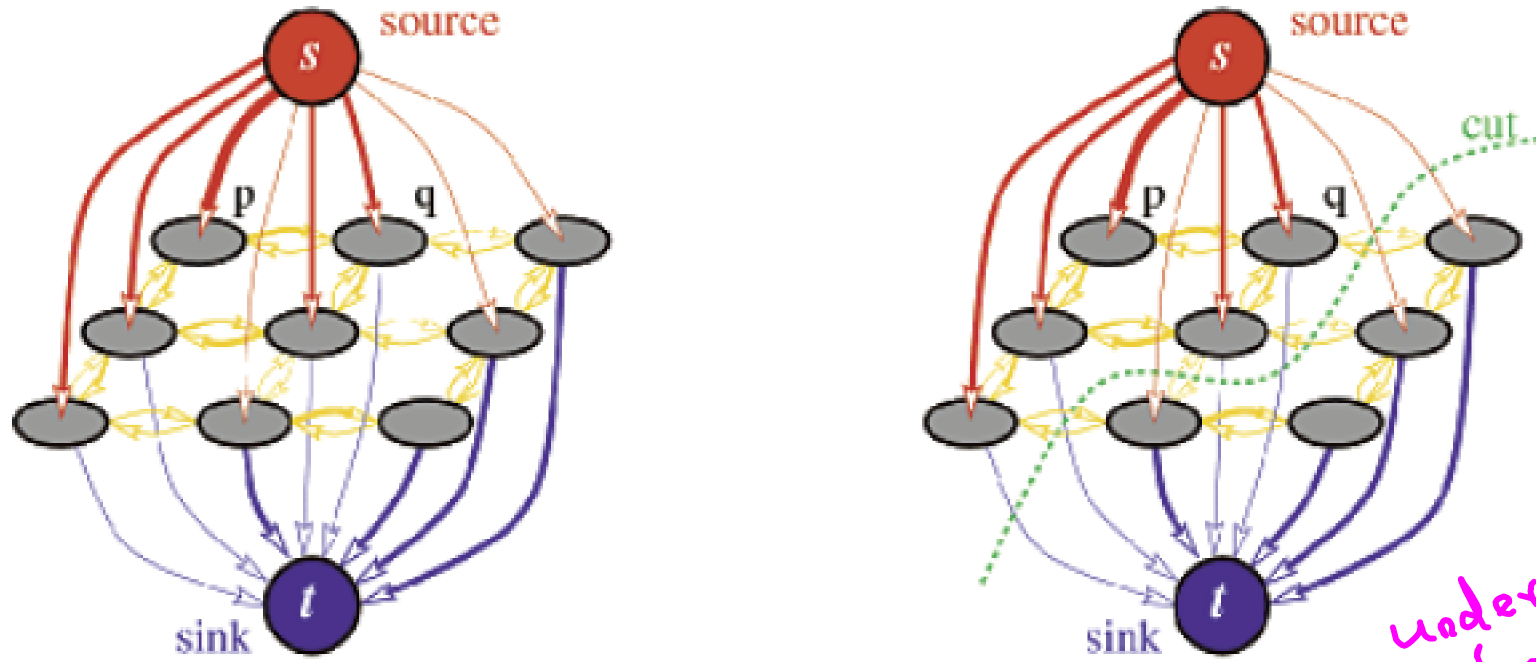
$$= A + \begin{bmatrix} 0 & 0 \\ C-A & C-A \end{bmatrix} + \begin{bmatrix} 0 & D-C \\ 0 & D-C \end{bmatrix} + \begin{bmatrix} 0 & B+C-A-D \\ 0 & 0 \end{bmatrix}$$



This split is to ensure (as per max flow view) that total incoming flow = total outgoing flow.

Claim (from course on algorithms)

See slide 25 & slides 7-10 of → www.usna.edu/Users/math/rmm/CS/DarbonSlides.pdf



under assumption of regularity

Max-flow solution = Min-cut solution = Soln to MAP inference for Markov N/W

The max-flow problem and min-cut problem can be formulated as two primal-dual linear programs.

Max-flow (Primal)	Min-cut (Dual)
maximize $\lvert f\rvert = \nabla_s$	minimize $\sum_{(i,j)\in E} c_{ij} d_{ij}$
subject to $$\begin{array}{rcll} f_{ij} & \leq & c_{ij} & (i,j)\in E \\ \sum_{j:(j,i)\in E} f_{ji} - \sum_{j:(i,j)\in E} f_{ij} & \leq & 0 & i\in V, i\neq s,t \\ \nabla_s + \sum_{j:(j,s)\in E} f_{js} - \sum_{j:(s,j)\in E} f_{sj} & \leq & 0 & \\ -\nabla_s + \sum_{j:(j,t)\in E} f_{jt} - \sum_{j:(t,j)\in E} f_{tj} & \leq & 0 & \\ f_{ij} & \geq & 0 & (i,j)\in E \end{array}$$	subject to $$\begin{array}{rcll} d_{ij} - p_i + p_j & \geq & 0 & (i,j)\in E \\ p_s & = & 1 & \\ p_t & = & 0 & \\ p_i & \geq & 0 & i\in V \\ d_{ij} & \geq & 0 & (i,j)\in E \end{array}$$

incoming flow

outgoing flow

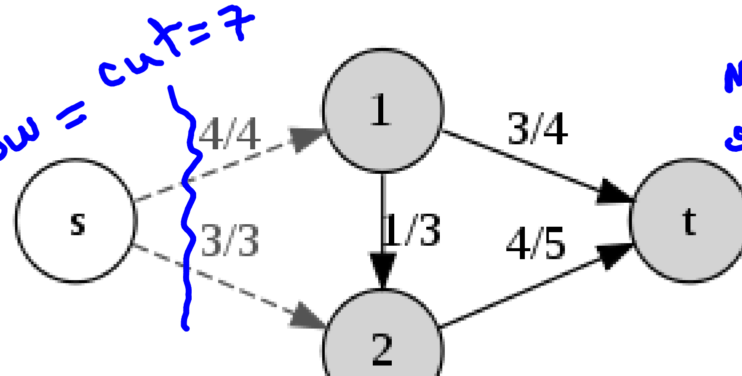
flow

capacity

$d_{ij} = 1$ if (i,j) is part of cut

$p_i = 1$ if (i) is on side of s

flow = cut = 7



max flow = min cut thm states that all edges on cut will be filled to their complete capacities

H/w: Prove that this max-flow/mincut formulation solves MAP inference for markov n/w

Algorithm for Maximum flows

- Assumption (recall): capacities are nonnegative
- Mainly two classes for computing maximum flows:
  - Augmenting Path class: augment flow along paths from source to sink while maintaining mass balance constraints.
  - Preflow-push class: flood the network so that some nodes have excesses. Send excess toward the sink or backward the source.
- Complexity (n=# nodes, m=# arcs):
  - Labelling: $O(nmC)$
  - Sucessive shortest path: $O(n^2 m)$
  - FIFO preflow-push: $O(n^3)$
  - Highest preflow-push: $O(n^2 \sqrt{m})$
  - Excess scaling: $O(nm + n^2 \log C)$

  where $C = \max_{ij} c_i j$

From "Regularity" to "submodularity"

Regularity $\Rightarrow$ $w_E^T f_E(x_i, x_j, 0, 0) + w_E^T f_E(x_i, x_j, 1, 1)$

$$\leq$$

$$w_E^T f_E(x_i, x_j, 0, 1) + w_E^T f_E(x_i, x_j, 1, 0)$$

Submodularity $\Rightarrow$ Let S be a finite set and $g: 2^S \rightarrow \mathbb{R}$ be a real valued function defined on the set of all subsets of S. g is called submodular if for any $X, Y \subset S$: $g(X)+g(Y) \geq g(X \cup Y)+g(X \cap Y)$

OR equivalently

if for any $X \subset S$ and $i, j \in S - X$

$$g(X \cup \{j\}) - g(X) \geq g(X \cup \{i,j\}) - g(X \cup \{i\})$$

Reflects the principle of DIMINISHING RETURNS

Connection with regularity:

g(set of nodes) = sum of capacities of edges connecting those nodes to the sink