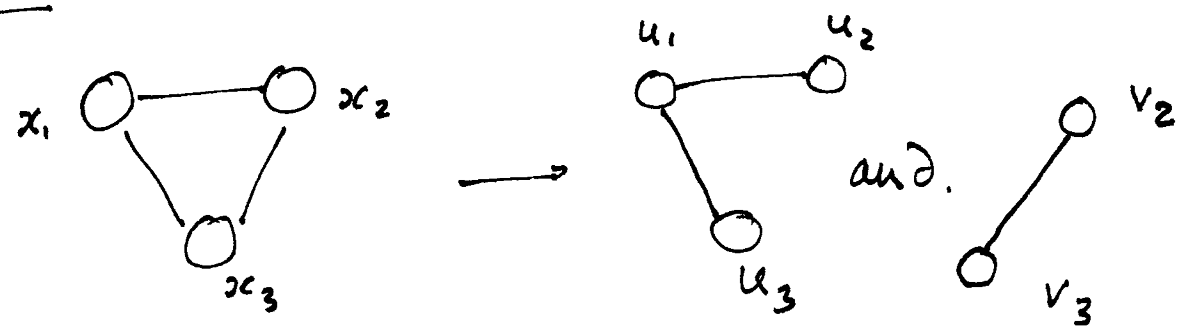
Domain decomposition (or dual decomposition) methods

1) Apply to more than MRF problems.
2) General idea
   - decompose problem into independent parts, by duplicating variables
   - impose constraints to make them duplicated variables agree
   - set up Lagrangian.
   - iterate:
     - go down (up) primal vars
     - go up (down) dual vars. using subgradient descent
   - Now break ties.

Example:



So $\min \sum_i m_i(x_i) + \sum_{ij} m_{ij}(x_i, x_j)$

becomes

$$\min m_1(u_1) + m_2(u_2) + m_3(u_3) + m_{12}(u_1, u_2) + m_{13}(u_1, u_3)$$
$$+ m_2(v_2) + m_3(v_3) + m_{23}(v_2, v_3)$$

st $u_2 = v_2$, $u_3 = v_3$

write $\min h_u(u_1, u_2, u_3) + h_v(v_2, v_3)$

st $u_2 = v_2$, $u_3 = v_3$

Lagrangian:

$$\mathcal{L}(u,v,\lambda) = h_u(u_1, u_2, u_3) + h_v(v_2, v_3) + \lambda_1(u_2 - v_2) + \lambda_2(u_3 - v_3)$$

Dual

$$G(\lambda) = \inf_{(u,v)} \mathcal{L}(u,v,\lambda) \longleftarrow \text{Lower bound}$$

So we'd like λ^* to max $G(\lambda)$.

Strategy:

- for fixed λ^n, evaluate $G(\lambda^n)$
- now get subgradient, update λ^{n+1}

Evaluating $G(\lambda)$

- find $\min_{u,v} \mathcal{L}(u, v, \lambda^n)$
- by choice of h_u, h_v, this is easy.

Update λ:

$$\partial G(\lambda^n) = \text{subgradient}$$

$$= \begin{pmatrix} u_2 - v_2 \\ u_3 - v_3 \end{pmatrix} \longrightarrow \text{now take a small step}$$

Notice this is good for integer problems,
because we have <u>integer points</u>

<u>But</u> they may not agree at overlap.

- random allocation
- voting.

<u>Thm</u>: if they do agree at overlap, at convergence, we have exact soln.
(Fairly obvious)

<u>Notice also</u>: this is rather good for other problems.

<u>Important cases</u>:

- big problem, parallelism (origins).
- reliable local experts.

reliable local experts eg

1) Parser, which is discriminative works by

parses.

$$\text{parse} = \arg\max_y w_p^T \phi(x, y)$$

sentence

2) Part of speech tagger, disc, same mechanism.

$$\text{tag} = \arg\max_z w_t^T \psi(x, z).$$

Now
- notice parsing implies some sort of tagging
- each problem individually has quite a good soln
- each allows exact inference

Idea:

a) $\hat{y} = (\hat{y}, z_p)$

expose implicit pos info in parse.

b) $\max \quad w_p^T \varphi(x, (\hat{y}, z_p)) + w_t^T \psi(x, z_t)$

s.t. $\quad z_p = z_t$

c) this fits into previous recipe; but

notice

- through constraints, expertise of tagger biases parser
- ditto, parser ... tagger.

d) improves performance.