Back to MRF's

recall that
\[
\max_x \frac{x^T A x}{2} + b^T x \quad \text{st } x \in \{0, 1\}
\]
is easy & if \( a_{ij} > 0 \)

for a 2 label MRF, write energy
\[
E = \sum_{i \in \text{sites}} E_i(l_i) + \sum_{i \neq j \in \text{sites}} E_{ij}(l_i, l_j)
\]

But 2 labels \( a, b \) \\
write \( l_i = a \) \( x_i = 1 \) \( l_i = b \)

then
\[
E = \sum_{i \in \text{sites}} \left[ E_i(a) \cdot (-x_i) + E_i(b) x_i \right] + \sum_{ij \in \text{sites}} \left[ -E_{ij}(a, a) (1-x_i) (1-x_j) \\
+ E_{ij}(a, b) (1-x_i) x_j \\
+ E_{ij}(b, a) x_i (1-x_j) \\
+ E_{ij}(b, b) x_i x_j \right]
\]
Back to MRF's

Recall that
\[
\max_x x^T Ax + b^T x
\]
\[\text{st } x \in \{0, 1\}^n\]
is easy LP if \( a_{ij} > 0 \)

for a 2 label MRF, write energy
\[
E = \sum_{i \in \text{sites}} E_i(l_i) + \sum_{ij \in \text{sites}} E_{ij}(l_i, l_j)
\]

But 2 labels, \( a, b \)
write \( x_i = \begin{cases} 0 & l_i = a \\ 1 & l_i = b \end{cases} \)

Then
\[
E = \sum_{i \in \text{sites}} \left[ E_i(a)(1-x_i) + E_i(b)x_i \right]
+ \sum_{ij \in \text{sites}} \begin{bmatrix}
- E_{ij}(a, a)(1-x_i)(1-x_j) \\
+ E_{ij}(a, b)(1-x_i)x_j \\
+ E_{ij}(b, a)x_i(1-x_j) \\
+ E_{ij}(b, b)x_i x_j
\end{bmatrix}
\]
we want to \( \min \) this

\[
\max_x x^\top A x + b^\top x \\
\text{ s.t. } x \in \{0, 1\}
\]

\[
\min_x x^\top A x - b^\top x \\
\text{ s.t. } x \in \{0, 1\}
\]

So easy energy \( \Rightarrow \)

\[
E_{ij}(6, 6) + E_{ij}(a, a) < E_{ij}(a, b) + E_{ij}(6, a)
\]

(better to agree than disagree).

And this is a cut problem

\( \Rightarrow \) dual to max-flow

\( \Rightarrow \) flow cut alg.
What if there are more than 2 labels?

\[
\begin{align*}
\min_x & \quad -x^T A x - b^T x \\
\text{subject to} & \quad x \in \{0, 1\}^n \\
\text{and} & \quad h^T x = b
\end{align*}
\]

(use the \(x\) to pick the label).

Two important strategies:

\[x - \beta\text{ swap.}\]

- take points w/ label \(x, \beta\) ONLY
- now either
  - leave
  - swap

\[x - \text{expansion}\]

- take points w/ label \(\text{NOT } x\)
- now either
  - leave
  - make \(x\).
$x - \beta$ swap:
- ignore
- $x_i \leftarrow$ one entry for each node labelled either $x$ or $\beta$

$$x_i = \begin{cases} 1 & \text{lemma } x \\ 0 & \text{swap } \beta \end{cases}$$

- notice $\equiv$ lemma, swap.

energy:
- unary terms for non $(x, \beta)$ nodes are fixed
- binary terms for $(x, \beta)$ nodes are fixed
- binary terms for $1$ non $(x, \beta)$, $1(x, \beta)$ become unary terms
- binary terms for $(x, \beta)$ terms are what matter.
Energy:

\[ E = \begin{cases} \text{quinary, binary terms} & \text{for non-} \alpha, \beta \\ \end{cases} \]

\[ + \sum_{i \in \alpha, \beta \text{ sites}} \left\{ E_i(\alpha) x_i + E_i(\beta)(1-x_i) \right\} \]

\[ + \sum_{j \in \alpha, \beta \text{ sites}} \left\{ \left[ \sum_k E_{kj}(\alpha, \alpha) \right] x_i + \left[ \sum_k E_{kj}(\alpha, \beta) \right] (1-x_i) \right\} \]

\[ + \sum_{i,j \in \alpha, \beta \text{ sites}} \left[ E_{ij}(\alpha, \alpha) x_i x_j + E_{ij}(\alpha, \beta) x_i (1-x_j) + E_{ij}(\beta, \alpha)(1-x_i) x_j + E_{ij}(\beta, \beta) (1-x_i)(1-x_j) \right] \]

\[ \text{condition} \]

want \( E_{ij}(\alpha, \beta) + E_{ij}(\beta, \alpha) \geq E_{ij}(\alpha, \alpha) + E_{ij}(\beta, \beta) \)
\( \alpha \)-expansion:

\[ x_i = \begin{cases} \text{D} & \text{leave} \\ \text{make label } \alpha. & \text{ } \end{cases} \]

Energy:

- Unary terms for non-\( \alpha \)-nodes
  and unary on \( x_i \), \( \alpha \)-nodes are fixed
- Binary non-\( \alpha \), \( \alpha \) terms become unary
- Binary \( \alpha \), \( \alpha \) terms become constant
- Binary non-\( \alpha \), non-\( \alpha \) become binary.
Energy = 

\[ \sum \text{ unary, binary terms w.r.t. } \alpha \text{ nodes } \]

\[ + \sum_{i \in \text{non-}\alpha} \left[ E_i(\alpha)(1-x_i) + E_i(x)x_i \right] \]

\[ + \sum_{i \in \text{non-}\alpha} \left[ \left( \sum_{j \in \alpha} E_{ij}(\alpha,\alpha) \right)(1-x_i) + \left( \sum_{j \in \alpha} E_{ij}(\alpha,\alpha) \right)x_i \right] \]

\[ + \sum_{i,j \in \text{non-}\alpha} \left[ E_{ij}(\alpha,\alpha)(1-x_i)(1-x_j) + E_{ij}(\alpha,\alpha)(1-x_i)x_j + E_{ij}(\alpha,\alpha)x_i(1-x_j) + E_{ij}(\alpha,\alpha)x_i x_j \right] \]

Condition: want

\[ E_{ij}(\alpha,\alpha) + E_{ij}(\alpha,\alpha) > E_{ij}(\alpha,\alpha) + E_{ij}(\alpha,\alpha). \]
Ways to meet conditions:

\( E \) is metric if

\( E(x, y) = 0 \iff x = y \)

\( E(x, y) = E(y, x) \geq 0 \)

\( E(x, y) \leq E(x, z) + E(z, y) \) (triangle inequality)

\( E \) is semimetric if \( a), b) \) true but not c)

Notice

\( E \) semimetric \( \Rightarrow \)

\( E(x, y) + E(y, x) \geq E(x, x) + E(y, y) \)

\( \therefore \ x - y \) swap OK

\( E \) metric

\( \Rightarrow \) \( E(x, y) + E(y, z) \geq E(x, z) + E(y, y) \)

\( = 0 \)

\( \therefore \) equality no big worry.

\( \therefore \) \( x - y \) swap OK
What graph should I cut?

I have

\[ \sum_{ij} \left[ \begin{array}{c} E_{ij}(0,0)(1-x_i)(1-x_j) + \\ E_{ij}(1,0)x_i(1-x_j) + \\ E_{ij}(0,1)(1-x_i)x_j + \\ E_{ij}(1,1)x_ix_j \end{array} \right] + \sum_i \left[ E_{i}(0)(1-x_i) + E_{i}(1)x_i \right] \]

\[ x \in \{0,1\}^2 \]

So I need to get it into a graph, and cut that — what graph do I cut? (earlier notes rather vague)

Strategy: 1. Describe a graph repn for component fns

2. Show how to add.
Graph rep H:

1. A graph has \( n+2 \) nodes \( s, t, x_i \).

2. Set up so:
   - \( x_i \in S \) side of cut \( \Rightarrow x_i = 0 \)
   - \( x_i \in t \) side \( \Rightarrow x_i = 1 \)

3. \( \text{Val(Cut)} = \text{Energy(vars)} + \text{const} \)

Reph linear functions:

\[
\sum_i \left[ E_i(0)(1-x_i) + E_i(1)x_i \right]
\]

- do one term (then sum).
Recall capacities are $>0$ for max flow - min cut.

So we want a graph where
1. tap, cuts rep'n cost
2. capacities $>0$

Let have $E_i(1) x_i + (1-x_i) E_i(0)$

Case 1: $\frac{1}{E_i(0)} > E_i(1)$.

Notice 2 cuts:
- $E_i(1) - E_i(0)$
- Value is $E_i(1) - E_i(0)$
- Value is $0$
So:

\[
\text{Value of cut} = \text{Energy represented by cut} - E_i(0)
\]
Case 2:

\[ E_i(1) \neq E_i(0) \leq \text{not greater than} \]

Again 2 cuts:

\[ E_i(0) - E_i(1) \]

So:

Value of cut = Energy repu by cut \( - E_i(1) \)
So now we can do any linear function.

\[ E_1(0) x (1-x_i) + E_1(1) x_i + E_2(0)(1-x_i) + E_2(1) x_i. \]

(\text{say})

\[ E_1(0) - E_1(1) \]

\[ E_2(0) - E_2(1) \]

etc.

Rep\'n Binary (\(= \text{Quadratic}\)) terms

Notice we can decompose

\[
\sum_{ij} \left[ E_{ij}(0,0)(1-x_i)(1-x_j) + E_{ij}(1,0)x_i(1-x_j) \\
+ E_{ij}(0,1)(1-x_i)x_j + E_{ij}(1,1)x_i x_j \right]
\]

as a sum of terms. — we need only show how to deal w/ 1 pair
Decompose as:

\[ E_{ij}(0,0) + \text{linear} \]

\[ + (1-x_i)O + x_i(E_{10}-E_{oo}) \]

\[ + (1-x_j)O + x_j(E_{11}-E_{10}) \]

\[ + (1-x_i)x_j(E_{10}+E_{01}-E_{oo}-E_{11}) \]

Cases depend on signs of linear coeffs:

1) \( E_{10}-E_{oo} > 0, \ E_{10}-E_{11} > 0 \)

(Notice 1,0 can't be soln.)

\[ \text{Val}(\text{cut}) = E_{10}-E_{oo} \]

\[ \text{Energy} = E_{11} \]

\[ = \text{Val}(\text{cut}) - E_{10} + E_{oo} + E_{11} \]
\[
\text{Val}(\text{Cut}) = E_{10} + E_{01} - E_{00} - E_{11}
\]

\[\text{Energy} = E_{01} = \text{Val}(\text{Cut}) - E_{10} + E_{00} + E_{11}\]

\[
\text{Val}(\text{Cut}) = E_{10} - E_{11}
\]

\[\text{Energy} = E_{10} - E_{00} = \text{Val}(\text{Cut}) - E_{10} + E_{00} + E_{11}\]

So:
\[
\text{Val}(\text{Cut}) = \text{Val}(\text{Cut}) + \text{const}
\]

\(\text{Case 2)}\)

\[E_{10} - E_{00} > 0, \quad E_{10} - E_{11} < 0\]

Notice this implies \(E_{00}\) is smallest.

\[E_{11} > E_{10} > E_{00}\]

and
\[E_{10} + E_{01} > E_{00} + E_{11}\]

\[\Rightarrow E_{01} > E_{00}\]
\( \text{Val}(\text{cut}) = \bar{E}_{oo} \overrightarrow{0} \) for all cuts.

\[
\text{Energy} = \bar{E}_{oo} = \text{Val}(\text{cut}) + E_{oo}
\]

**Case 3**

\( E_{oo} > E_{10} \)

\( E_{10} > E_{11} \)

**Case 4**
We can now do any OR.

QF such that

\[ E_{ij}(10) + E_{ij}(01) > E_{ij}(00) + E_{ij}(11) \]

- one node for each var
- for each linear term, insert edges, summing weights as needed
  - "quad"

\[ \text{Val} (\text{cut}) = \text{Energy} + \text{Const} \]

\[ \Rightarrow \text{Min} (\text{cut}) \text{ gives } \text{Min}(\text{Energy}) \]

This works both ways
(i.e. represent as cut \( \Rightarrow \) energy cond).