Pseudo-boolean optimization and graph cuts

Markov Random Fields

Consider a set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ taking values in a set S (usually a discrete set of values). Assume a **neighbourhood structure**: for each variable X_i there is defined a set of variables $\mathcal{N}_i \subset \mathcal{X} - \{i\}$ such that

1. $i \in \mathcal{N}_j \Rightarrow j \in \mathcal{N}_j$.
2. $P(X_i = x_i \mid X_j = x_j; j \neq i) = P(X_i = x_i \mid X_j = x_j; j \in \mathcal{N}_i)$.

In other words the conditional probability distribution of a given variable X_i depends only the values of its neighbors.

This is called a *Markov Random Field* (MRF).

Notation. Denote the set (or vector) of all random variables by X and a set of values of the random variables by $\mathbf{x}$.

MRFs and graphs

Define an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that

1. The vertices $\mathcal{V}$ are in one-to-one correspondence with the random variables X_i. (In fact we will refer to the vertices as X_i.)
2. There is an edge from X_i to X_j if and only if $i \in \mathcal{N}_j$.

Example: Image graph (4-connected) has cliques of size 1 and 2.

- size 1 (vertices)
- size 2 (pairs of vertices joined by an edge).

6-connected and 8-connected graphs have cliques of size 3 and 4 respectively.

Gibbs distribution

Refer to paper Geman and Geman "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Relaxation Restoration of Images."

Theorem. The complete probability distribution of the MRF is given by

$$P(\mathrm{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \exp\left(-E_C(\mathbf{x})\right)$$

where $\mathcal{C}$ represents the set of all cliques in the graph of the MRF.

1. $E_C(\mathbf{x})$ is an **energy function** depending only on the values of the vertices $X_i \in C$.
2. Z is a normalizing constant

Definition of pseudo-boolean function.

Define $\mathcal{B} = \{0, 1\}$. A pseudo-boolean function is a mapping

$$f : \mathcal{B}^n \to \mathbb{R} \ .$$

Variables: x_i. The set of variables will be denoted $X = \{x_i; i = 1, \ldots, n\}$.

Literals: Literals are $x_i, \bar{x}_i$, where $\bar{x}_i = 1 - x_i$. Use u to represent a literal. The set of literals is denoted by $L = \{x_i, \bar{x}_i; i = 1, \ldots, n\}$.

Three ways of representing a pseudo-boolean function.

1. Tableau. This lists all 2^n values of the function.
2. Posiform.

$$a_0 + \sum_i a_i u_i + \sum_{i,j} a_{ij} u_i u_j + \ldots$$

   where all coefficients are positive, except perhaps a_0.

3. Polynomial.

$$c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{1 \le i < j \le n} c_{ij} x_i x_j + \ldots \ .$$

Maximizing the probability

Commonly, we want to find the assignment $\mathbf{x}$ that maximizes the probability (the most probable state of the MRF).

We see that

$$\log P(\mathrm{X} = \mathrm{x}) \ = \ \text{const} - \sum_{C \in \mathcal{C}} E_C(\mathbf{x})$$

hence

$$\text{argmax}_{\mathbf{x}} \ P(\mathrm{X} = \mathrm{x}) \ = \ \text{argmin}_{\mathbf{x}} \ \sum_{C \in \mathcal{C}} E_C(\mathbf{x})$$

Hence, finding the most probable state of the MRF is equivalent to minimizing the **energy function**

$$\sum_{C \in \mathcal{C}} E_C(\mathbf{x})$$

where each $E_C(\mathbf{x})$ is a function only of the variables in the clique C.

Transformation between different forms.

Example – Tableau to posiform

x_1	x_2	x_3	val	term
0	0	0	3	$\bar{x}_1\bar{x}_2\bar{x}_3$
0	0	1	2	$\bar{x}_1\bar{x}_2 x_3$
0	1	0	−5	$\bar{x}_1 x_2\bar{x}_3$
0	1	1	1	$\bar{x}_1 x_2 x_3$
1	0	0	−4	$x_1\bar{x}_2\bar{x}_3$
1	0	1	−2	$x_1\bar{x}_2 x_3$
1	1	0	3	$x_1 x_2\bar{x}_3$
1	1	1	−7	$x_1 x_2 x_3$

Corresponding posiform computed as follows.

$$3\,\bar{x}_1\bar{x}_2\bar{x}_3 + 2\,\bar{x}_1\bar{x}_2 x_3 - 5\,\bar{x}_1 x_2\bar{x}_3 + 1\,\bar{x}_1 x_2 x_3 - 4\,x_1\bar{x}_2\bar{x}_3 - 2\,x_1\bar{x}_2 x_3 + 3\,x_1 x_2\bar{x}_3 - 7\,x_1 x_2 x_3$$

Replace the terms with negative coefficients.

$$\begin{aligned} -x_1\bar{x}_2 x_3 &= -(1 - \bar{x}_1)\bar{x}_2 x_3 \\ &= \bar{x}_1\bar{x}_2 x_3 - \bar{x}_2 x_3 \\ &= \bar{x}_1\bar{x}_2 x_3 - (1 - x_2) x_3 \\ &= \bar{x}_1\bar{x}_2 x_3 + x_2 x_3 - x_3 \\ &= \bar{x}_1\bar{x}_2 x_3 + x_2 x_3 + \bar{x}_3 - 1 \end{aligned}$$

All terms with negative coefficients can be replaced in this way.

Transform to posiform

Replace the terms with negative coefficients.

$$\begin{aligned} -x_1\bar{x}_2x_3 &= -(1-\bar{x}_1)\bar{x}_2x_3 \\ &= \bar{x}_1\bar{x}_2x_3 - \bar{x}_2x_3 \\ &= \bar{x}_1\bar{x}_2x_3 - (1-x_2)x_3 \\ &= \bar{x}_1\bar{x}_2x_3 + x_2x_3 - x_3 \\ &= \bar{x}_1\bar{x}_2x_3 + x_2x_3 + \bar{x}_3 - 1 \end{aligned}$$

All terms with negative coefficients can be replaced in this way.

Transform to polynomial

Replace each $\bar{x}_i$ by $1 - x_i$ and multiply out. Example.

$$\begin{aligned} \bar{x}_1\bar{x}_2x_3 &= (1-x_1)(1-x_2)x_3 \\ &= x_3 - x_1x_3 - x_2x_3 + x_1x_2x_3 \end{aligned}$$

Ambiguity of posiform representation.

$$\begin{aligned} x_1\bar{x}_2 &= (1-\bar{x}_1)(1-x_2) \\ &= 1 - \bar{x}_1 - x_2 + \bar{x}_1x_2 \\ &= 1 - (1-x_1) - (1-\bar{x}_2) + \bar{x}_1x_2 \\ &= -1 + x_1 + \bar{x}_2 + \bar{x}_1x_2 \end{aligned}$$

Uniqueness of polynomial representation.

Polynomial representation is unique.

Proof: We can compute the coefficients of the polynomial representation by evaluating function.

Example: $f(\mathbf{x}) = a_0 + \sum_i a_ix_i + \sum_{ij} x_ix_j + \ldots$

1. Determination of the constant term: $f(0) = a_0$.
2. Determination of a_i. Evaluate at $x_i = 1$, $x_j = 0$ for $i \neq j$, gives $f(0,\ldots,1,\ldots,0) = a_0 + a_i$, allows us to determine a_i.
3. Determine the coefficients a_{ij} by evaluating by $x_i = x_j = 1$ and $x_k = 0$, otherwise. Gives $a_0 + a_i + a_j + a_{ij}$, allows us to determine a_{ij}.

Quadratic functions

For a quadratic pseudo-boolean function, we can formulate an alternative form.

$$f(\mathbf{x}) = a_0 + \sum_i a_iu_i + \sum_{1\le i<j\le n} a_{ij}\bar{x}_ix_j$$

where $u_i = x_i$ or $\bar{x}_i$, and $a_i \geq 0$ for all i.

This representation is unique.

Cost representation

Sometimes, we express the function to be minimized in terms of costs of the form $E_{i;p}$ or $E_{ij;pq}$, where i indexes the variable, and p, q are boolean values, 0 or 1.

Thus, cost $E_{i;p}$ is incurred if variable x_i takes value p, and not otherwise. Similarly, $E_{ij;pq}$ is incurred if $x_i = p$ and $x_j = q$. Thus, the costs associated with two variables x_i and x_j are made up of linear terms such as

$$E_{i;1}\ x_i + E_{i;0}\ \bar{x}_i$$

plus quadratic terms

$$E_{ij;00}\ \bar{x}_i\bar{x}_j + E_{ij;11}\ x_i x_j + E_{ij;01}\ \bar{x}_i x_j + E_{ij;10}\ x_i\bar{x}_j\ .$$

Higher-degree terms may also be expressed in a similar way.

Minimum of $f(x)$ and posiforms.

For a posiform ϕ, define $C(\phi)$ to be the constant term.

Theorem. $\min_{\mathbf{x}} f(\mathbf{x}) = \max_{\phi} C(\phi)$, where ϕ ranges over all posiform representations of f.

Proof. Consider a tableau of the function. For instance:

$$f(\mathbf{x}) \equiv \left[\begin{array}{cccrl} x_1 & x_2 & x_3 & \text{val} & \\ 0 & 0 & 0 & 3 & \bar{x}_1\bar{x}_2\bar{x}_3 \\ 0 & 0 & 1 & 2 & \bar{x}_1\bar{x}_2 x_3 \\ 0 & 1 & 0 & -5 & \bar{x}_1 x_2\bar{x}_3 \\ 0 & 1 & 1 & 1 & \bar{x}_1 x_2 x_3 \\ 1 & 0 & 0 & -4 & x_1\bar{x}_2\bar{x}_3 \\ 1 & 0 & 1 & -2 & x_1\bar{x}_2 x_3 \\ 1 & 1 & 0 & 3 & x_1 x_2\bar{x}_3 \\ 1 & 1 & 1 & -7 & x_1 x_2 x_3 \end{array}\right]$$

Now, subtract the minimum value of the function, to get a tableau representation of $f(\mathbf{x}) - \min_{\mathbf{x}} f(\mathbf{x})$. Namely

$$f(\mathbf{x}) - \min_{\mathbf{x}} f(\mathbf{x}) \equiv \left[\begin{array}{cccrl} x_1 & x_2 & x_3 & \text{val} & \\ 0 & 0 & 0 & 10 & \bar{x}_1\bar{x}_2\bar{x}_3 \\ 0 & 0 & 1 & 9 & \bar{x}_1\bar{x}_2 x_3 \\ 0 & 1 & 0 & 4 & \bar{x}_1 x_2\bar{x}_3 \\ 0 & 1 & 1 & 8 & \bar{x}_1 x_2 x_3 \\ 1 & 0 & 0 & 3 & x_1\bar{x}_2\bar{x}_3 \\ 1 & 0 & 1 & 2 & x_1\bar{x}_2 x_3 \\ 1 & 1 & 0 & 10 & x_1 x_2\bar{x}_3 \\ 1 & 1 & 1 & 0 & x_1 x_2 x_3 \end{array}\right]$$

Now, write this out as a posiform

$$f(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) \ + \ 10\,\bar{x}_1\bar{x}_2\bar{x}_3 + 9\,\bar{x}_1\bar{x}_2 x_3 + 4\,\bar{x}_1 x_2\bar{x}_3 + 8\,\bar{x}_1 x_2 x_3 + 3\,x_1\bar{x}_2\bar{x}_3 + 2\,x_1\bar{x}_2 x_3 + 10\,x_1 x_2\bar{x}_3 + 0\,x_1 x_2 x_3\ .$$

This is a posiform representation ϕ of the function, so we see $\min_{\mathbf{x}} f(\mathbf{x}) = C(\phi)$, so

$$\min_{\mathbf{x}} f(\mathbf{x}) \le \max_{\phi} C(\phi)\ .$$

However, for any posiform ϕ and value $\mathbf{x}$, we have $f(\mathbf{x}) \ge C(\phi)$, so

$$\min_{\mathbf{x}} f(\mathbf{x}) \ge \max_{\phi} C(\phi)$$

so equality holds.

NP hardness.

We can show that in general, minimizing a posiform is an NP hard problem (with respect to the number of variables n). Consider a problem in which all terms are of degree 3 and all coefficients are 1 in a posiform representation. This is

$$f(\mathbf{x}) = \sum_{i,j,k} u_i u_j u_k$$

where u_i, u_j and u_k are literals. For example:

$$f(x_1, x_2, x_3) = x_1\bar{x}_2 x_3 + \bar{x}_1 x_2\bar{x}_3 + x_1 x_2\bar{x}_3\ .$$

Clearly $f(\mathbf{x}) \ge 0$. To determine whether $f(\mathbf{x}) = 0$ for some $\mathbf{x}$ is the 3-SAT problem: Can we make a boolean assignment so that all terms $u_i u_j u_k$ are zero.

Another way to see this is to observe that a posiform or polynomial representation of a general pseudo-boolean function can have up to 2^n terms. Thus, given an arbitrary function over n variables, presented in polynomial format, it requires exponential time just to read the coefficients of the function.

An approach to minimizing a pseudo-boolean function.

Let $\mathcal{F}^n$ be the set of posiforms with terms of degree at most n. Define

$$C_2(f) = \max_{\phi \in \mathcal{F}^2} C(\phi) \ .$$

Then, since $\min_{\mathbf{x}} f(\mathbf{x}) = \max_{\phi \in \mathcal{F}} C(\phi)$, we see

$$C_2(f) \leq \min_{\mathbf{x}} f(\mathbf{x}) \ .$$

Thus, $C_2(f)$ gives a lower bound on $\min_{\mathbf{x}} f(\mathbf{x})$.

Solving using Linear Programming.

Let

$$\phi(\mathbf{x}) = a_0 + \sum_{i=1}^{n} (a_i x_i + a_{\bar{i}} \bar{x}_i) + \sum_{i,j=1}^{n} \left(a_{ij} x_i x_j + a_{\bar{i}j} \bar{x}_i x_j + a_{i\bar{j}} x_i \bar{x}_j + a_{\bar{i}\bar{j}} \bar{x}_i \bar{x}_j\right)$$

and

$$f(\mathbf{x}) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{1 \leq i,j \leq n} c_{ij} x_i x_j$$

represent the same function. Then we can compute the relationship between the coefficients a and c.

Thus

$$\begin{aligned} c_0 &= a_0 + \sum_{i=1}^{n} a_{\bar{i}} + \sum_{i,j=1}^{n} a_{\bar{i}\bar{j}} \\ c_0 + c_i &= a_0 + \sum_{k=1}^{n} a_{\bar{k}} + \sum_{k,j=1}^{n} a_{\bar{k}\bar{j}} - a_{\bar{i}} + a_i + \ldots \\ c_0 + c_i + c_j + c_{ij} &= L_{ij}(a_0, a_i, \ldots, a_{ij}) \end{aligned}$$

where L_{ij} is a linear expression in the coefficients a.

Thus, we can compute the lower bound as follows.

Minimize $\quad a_0$

Subject to $\quad c_0 = a_0 + \sum_{i=1}^{n} a_{\bar{i}} + \sum_{i,j=1}^{n} a_{\bar{i}\bar{j}}$

$$c_0 + c_i = L_i(a_0, \ldots, a_{ij}) \ \forall i$$

$$c_0 + c_i + c_j + c_{ij} = L_{ij}(a_0, \ldots, a_{ij}) \ \forall i, j$$

and $\quad a_i, \ldots, a_{ij} \geq 0$.

Note: This gives a lower bound for the solution, but it does not give the assignment.

Simple Graph Representability.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{W})$ be a directed weighted graph with vertices $\mathcal{V}$ and weights $\mathcal{W}$. The weights are real numbers (not necessarily positive).

We consider a pseudo-boolean function $f(x_1, \ldots, x_n)$ and consider graphs with $n + 2$ vertices. There are two vertices called 0 and 1 (or sometimes s and t) and other vertices that are labelled x_i, in one-to-one correspondence with the variables. There are certain edges with weights.

A **partition** of the graph is a division of the vertices into $\mathcal{V}_0$ and $\mathcal{V}_1$, where $0 \in \mathcal{V}_0$ and $1 \in \mathcal{V}_1$. If $\mathbf{x}$ is a particular value of the variable $\mathbf{x} = (x_1, \ldots, x_n)$, then $\mathcal{V}_0(\mathbf{x})$ is the set of vertices x_i with value 0, and $\mathcal{V}_1(\mathbf{x})$ is the set of vertices x_i with value 1. Thus $(\mathcal{V}_0(\mathbf{x}), \mathcal{V}_1(\mathbf{x}))$ is a particular partition of the graph.

Graph-representable functions

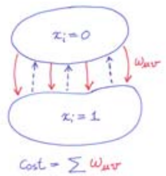
The **cost** of a partition $(\mathcal{V}_0, \mathcal{V}_1)$ is the sum of weights of all edges going from $\mathcal{V}_0$ to $\mathcal{V}_1$. *Formally,*

$$\text{Cost}(\mathcal{V}_0, \mathcal{V}_1) = \sum_{u \in \mathcal{V}_0, v \in \mathcal{V}_1} w(u, v)$$

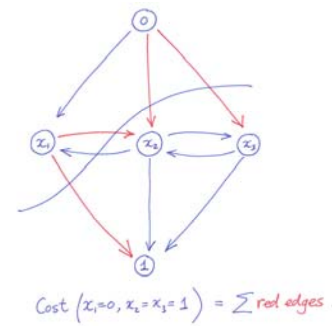
where $w(u, v)$ is the weight of the edge from vertex u to v.

Definition The function $f(\mathbf{x})$ is *simple graph representable* if there is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{W})$, such that for all x,

$$f(\mathbf{x}) = \text{Cost}(\mathcal{V}_0(\mathbf{x}), \mathcal{V}_1(\mathbf{x})) \ .$$



Cost of a partition is the sum of weights of edges passing from $\mathcal{V}_0$ to $\mathcal{V}_1$.



Representation of quadratic pseudo-boolean functions.

The easiest way to find a graph-representation of a quadratic pseudo-boolean function is as follows.

Step 1. Represent the function in the form

$$f(\mathbf{x}) = L + \sum_{i,j=1}^{n} a_{\bar{i}j} \bar{x}_i x_j$$

where L represents linear terms in literals x_i and $\bar{x}_i$.

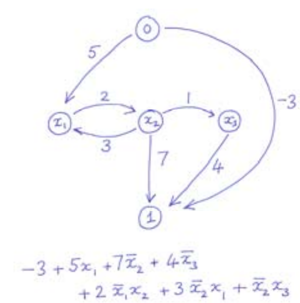
Step 2. Draw a graph with vertices labelled 0, 1 and x_i, and assign edges as follows:

1. For the constant term a_0, add an edge from 0 to 1 with weight a_0.
2. For a term $a_i x_i$, add an edge from 0 to x_i with weight a_i.
3. For a term $a_{\bar{i}} \bar{x}_i$, add an edge from x_i to 1 with weight $a_{\bar{i}}$.
4. For a term $a_{\bar{i}j} \bar{x}_i x_j$, add an edge from x_i to x_j with weight $a_{\bar{i}j}$.

Notes:

1. An edge from vertex u to vertex v with weight a represents the term $a\bar{u}v$, where $\bar{0} = 1$.
2. It is unnecessary to have both terms x_i and $\bar{x}_i$ occuring. Furthermore, non-zero linear terms can be of the form $a_i x_i$ with $a_i > 0$, or $a_{\bar{i}} \bar{x}_i$ with $a_{\bar{i}} > 0$.
3. It is unnecessary to have both $\bar{x}_i x_j$ and $\bar{x}_j x_i$ occuring. Thus, the function can be written as

$$f(\mathbf{x}) = L + \sum_{1 \le i < j \le n} a_{\bar{i}j} \bar{x}_i x_j \ .$$



Theorem. For a pseudo-boolean function represented as

$$f(\mathrm{x}) = L + \sum_{i,j=1}^{n} a_{\bar{i}j}\bar{x}_i x_j$$

the graph given by this construction is a simple graph-representation of f.

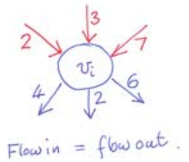
Proof: Consider a partition $(\mathcal{V}_0, \mathcal{V}_1)$ and an edge from vertex u to v (where $u, v \in X \cup \{0, 1\}$). Then the edge correpondes to a term $a_{uv}\bar{u}v$, and $a_{uv}\bar{u}v$ is non-zero if and only if $u \in \mathcal{V}_0$ and $v \in \mathcal{V}_1$.

Flow on a graph.

Given a graph with vertices v_i, a **flow** is a function $\phi : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ such that

1. $\phi(v_i, v_j) = -\phi(v_j, v_i)$
2. For all i, $\sum_j \phi(v_i, v_j) = 0$.

Think of Kirchoff's Current Law. Total flow into a vertex equals total flow out.



Permissible Flow

Later will will be interested in *permissible flow* on weighted graphs which will satisfy one other axiom.

1. If w_{ij} is the weight of an edge from vertex v_i to v_j, then $\phi(v_i, v_j) \leq w_{ij}$ for all i, j.

It is easily seen that there is no permissible flow on a weighted graph unless $w_{ij} + w_{ji} \geq 0$ for all i, j. In fact this is a necessary and sufficient condition.

Connection of flow and function representation.

Consider a weighted graph. If two edges are not connected, we add an edge with zero weight. Clearly, this makes no difference to the the function represented by the graph.

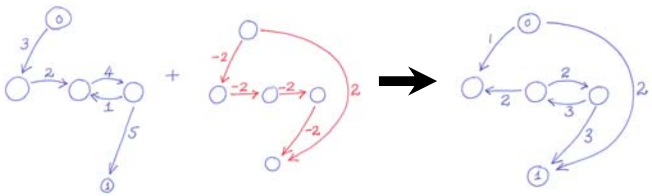
We now show the following theorem.

Theorem. Let w_{uv} and w'_{uv} be two different weights on a graph with vertices $\{x_i\} \cup \{0, 1\}$. Then the graphs $\mathcal{G} = (\mathcal{V}, \mathcal{W})$ and $\mathcal{G}' = (\mathcal{V}, \mathcal{W}')$ represent the same function $f(\mathrm{x})$ if and only if there is a flow ϕ_{uv} such that

$$w'_{uv} = w_{uv} + \phi_{uv}$$

for all vertices u and v.

Example of flow reweighting



$3x_1 + 2\bar{x}_1 x_2 + 4\bar{x}_2 x_3 + \bar{x}_3 x_2 + 5\bar{x}_3$

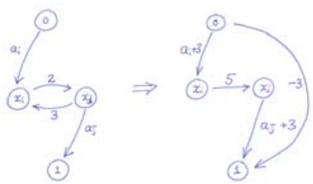
$= 2 + x_1 + 2x_1\bar{x}_2 + 2\bar{x}_2 x_3 + 3\bar{x}_3 x_2 + 3\bar{x}_3$

Proof. Suppose that w_{uv} and w'_{uv} represent the same function, and define $\phi_{uv} = w_{uv} - w'_{uv}$. We show that ϕ_{uv} is a flow on the graph.

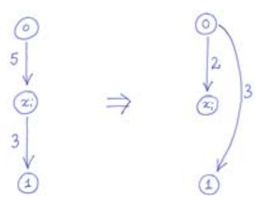
To do this, we observe that ϕ_{uv} is a set of weights representing a quadratic pseudo-boolean function representing the zero function. This function can be transformed via a flow transform (a *flow transform* is one that is induced by a flow) to the form

$$\phi(\mathbf{x}) = a_0 + \sum_i a_i u_i + \sum_{1 \le i < j \le n} \bar{x}_i x_j \ .$$

However, this form is unique, and hence all the coefficients are zero, and the function is identically zero.

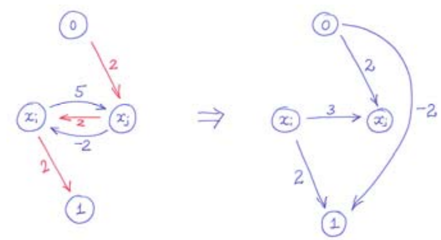


Simplification of quadratic terms



Simplification of linear terms

Example of rewriting formulas



$5\bar{x}_i x_j - 2\bar{x}_j x_i = -2 + 2x_j + 2\bar{x}_i + 3\bar{x}_i x_j$

Conversely, suppose that $\phi(u,v)$ is a flow, so that $w'(u,v) = w(u,v) + \phi(u,v)$ and let $f(\mathbf{x})$ and $f'(\mathbf{x})$ be the functions represented by these two weights. Then, we see that

$$f'(\mathbf{x}) = f(\mathbf{x}) + \sum_{u \in \mathcal{V}_0(\mathbf{x})} \sum_v \phi(u,v) \ ,$$

so $f(\mathbf{x}) = f(\mathbf{x}')$.

Otherwise stated: the function represented by a flow is identically zero.

This process is called reparametrization of the graph via the flow ϕ.

Maximum flow.

Drop the edge from 0 to 1. Flow need not be conservative at the nodes 0 and 1. Total flow is the amount of flow from 0 to 1. For a flow ϕ, define $\kappa(\phi)$ to be the *volume* of the flow – the flow out of node 0 and into node 1.

Theorem. Given a graph with weights w_{uv} and a maximum permissible flow ϕ^*_{uv}, let $w'_{uv} = w_{uv} - \phi^*_{uv}$. Then $w'(\mathbf{x}) = w(\mathbf{x}) - \kappa(\phi^*)$ for all $\mathbf{x}$, Hence

$$\min_{\mathbf{x}} f'(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) - \kappa(\phi^*)$$

$$\text{argmin}_{\mathbf{x}}\ f'(\mathbf{x}) = \text{argmin}_{\mathbf{x}}\ f(\mathbf{x})$$

Proof. By definition, $w'_{uv} \geq 0$ and there is no path from 0 to 1 with positive weights on all edges.

However, since all weights are positive in w', we see

$$\min_{\mathbf{x}} f'(\mathbf{x}) \geq 0 \ .$$

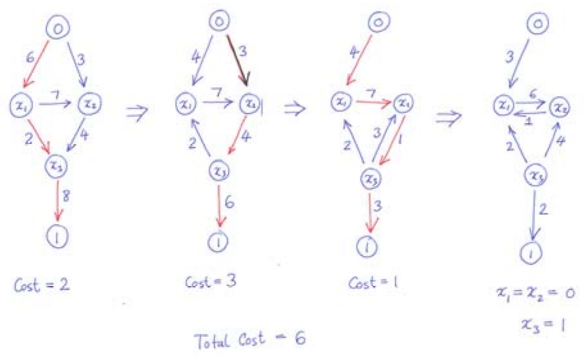
With weights w', let $\mathcal{V}_0$ be the set of nodes v_i reachable from 0 with positive edges and let $\mathcal{V}_1$ be the rest. Then there is no positive edge from $\mathcal{V}_0$ to $\mathcal{V}_1$, hence $\text{Cost}(\mathcal{V}_0, \mathcal{V}_1) = 0$. Let $\mathbf{x}^*$ be the variable assignment corresponding to this partition. Then, $f'(\mathbf{x}^*) = 0$. It follows that

$$\min_{\mathbf{x}} f'(\mathbf{x}) = 0,$$

and hence

$$\min_{\mathbf{x}} f(\mathbf{x}) = \kappa(\phi^*)$$

Max flow algorithm



Summary.

1. All quadratic pseudo-boolean functions $f(\mathbf{x})$ can be expressed as graphs $\mathcal{G} = (\mathcal{V}, \mathcal{W})$ such that the function value is equal to the cost of the corresponding partition:
$$f(\mathbf{x}) = \text{Cost}(\mathcal{V}_0(\mathbf{x}), \mathcal{V}_1(\mathbf{x}))$$
2. If all weights are positive, then the min-cut on the graph (minimum of the function) is equal to the maximal permissible flow.
3. A graph with weights w_{uv} can be reparametrized to a graph with non-negative weights if and only if
$$w(x_i, x_j) + w(x_j, x_i) \geq 0$$
for all pairs of variables x_i, x_j.
4. If all the weights are non-negative, then
$$\text{mincut} = \min_{\mathbf{x}} f(\mathbf{x}) = \text{maxflow}$$
and the minimization problem can be solved using a max-flow algorithm in polynomial time.

Regular functions.

A quadratic function is **regular** (or **submodular**) if it can be written in the form

$$f(x_1, \ldots, x_n) = a_0 + \sum_{i=0}^{n} a_i x_i \quad - \sum_{1 \leq i < j \leq n} a_{ij}\, x_i x_j \qquad (1)$$

with $a_{ij} \geq 0$ for all i, j.

A quadratic submodular function can be written as a posiform

$$a_0 + \sum_u a_u u + \sum_{i,j} a_{ij} \bar{x}_i x_j \qquad (2)$$

and hence is representable by a graph with non-negative weights.

Note. If the function is written in terms of $E_{ij;pq}$, then

$$\begin{aligned} & E_{ij;00}\, \bar{x}_i \bar{x}_j + E_{ij;11}\, x_i x_j + E_{ij;01}\, \bar{x}_i x_j + E_{ij;10}\, x_i \bar{x}_j \\ = \; & L + (E_{ij;00} + E_{ij;11} - E_{ij;01} - E_{ij;10})\, x_i x_j \end{aligned}$$

where L are linear terms.

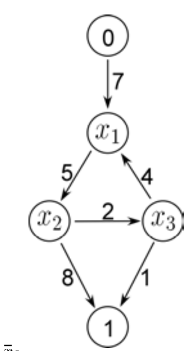
Hence,

$$a_{ij} = E_{ij;01} + E_{ij;10} - (E_{ij;00} + E_{ij;11})$$

Function is regular if

$$E_{ij;01} + E_{ij;10} - (E_{ij;00} + E_{ij;11}) \geq 0$$

Theorem. A quadratic pseudo-boolean function can be minimized using max-flow on its simple graph representation if and only if it is of the form (1) or equivalently (2).



Method.

1. Write the function in the form (2).
2. Coefficients in the posiform are exactly the edges on the graph.

Example.

$$\begin{aligned} & 7 + 7x_1 + 3\bar{x}_2 + x_3 - 5x_1x_2 - 2x_2x_3 + 4x_1\bar{x}_3 \\ = \; & 1 + 7x_1 + 8\bar{x}_2 + \bar{x}_3 + 5\bar{x}_1x_2 + 2\bar{x}_2x_3 + 4x_1\bar{x}_3 \end{aligned}$$

General Regular functions.

Definition. A pseudo-boolean function of any degree is called regular if any restriction to a two-variable function is regular, hence of the form $L + a_{ij}\bar{x}_i x_j$ with L linear and $a_{ij} \geq 0$.

Thus, set all variables but two to given values, you get a function of two variables, which must be regular.

Example.

$$-4x_1x_2 - 3x_1x_3 - 2x_2x_3 + 2\bar{x}_1x_2x_3 + 3x_1\bar{x}_2x_3 + 4x_1x_2\bar{x}_3$$

1. Set $x_3 = 1$ gives
$$\begin{aligned} & -4x_1x_2 + 2\bar{x}_1x_2 + 3x_1\bar{x}_2 + L \\ = \; & -9x_1x_2 + L \text{ which is submodular} \end{aligned}$$
2. Set $x_3 = 0$ gives $-8x_1x_2 + L$
3. Set $x_2 = 0, 1$, and $x_1 = 0, 1$ and test if submodular.

Cubic functions.

Observation A cubic pseudo-boolean function can be written uniquely in the form

$$L + \sum_{1 \leq i < j \leq n} a_{ij}x_ix_j + \sum_{1 \leq i < j < k \leq n} a_{ijk}u_iu_ju_k \qquad (1)$$

such that

1. L is a linear posiform without terms in both x_i and $\bar{x}_i$.
2. For each i, j, k, either $u_iu_ju_k = x_ix_jx_k$, or $u_iu_ju_k = \bar{x}_i\bar{x}_j\bar{x}_k$.
3. $a_{ijk} \leq 0$ for all i, j, k.
4. Coefficients a_{ij} may be positive or negative.
5. The coefficients are uniquely determined by these conditions.

Proof. First, get the cubic terms in the right form, then the quadratic terms, and finally the linear and constant terms.

Theorem. The cubic posiform in the form (??) is submodular if and only if the coefficients a_{ij} of the quadratic terms are all negative.

Proof. First, we show that this form is submodular. For two indices i, j, a restriction to x_i and x_j will be of the form

$$L + a_{ij}x_ix_j + \sum_{ij} b_{ij}x_ix_j + \sum_{ij} c_{ij}\bar{x}_i\bar{x}_j$$

where the b_{ij} and c_{ij} are sums of some of the coefficients a_{ijk} and hence are negative. However, we may write

$$\sum_{ij} c_{ij}\bar{x}_i\bar{x}_j = L + \sum_{ij} c_{ij}x_ix_j$$

and so the quadratic terms of the form $\bar{x}_ix_j$ have negative coefficients.

Conversely, suppose that the function is submodular. Consider two indices i, j. We wish to show that $a_{ij} \leq 0$. We restrict the function $f(\mathbf{x})$ by assigning values to each x_k except x_i and x_j. This gives $L + a_{ij}x_ix_j + C(\mathbf{x})$ where $C(\mathbf{x})$ represents cubic terms.

We wish to make an assignment to all other x_k so that the cubic terms vanish. There may exist a cubic term $a_{ijk}x_ix_jx_k$ or $a_{ijk}\bar{x}_i\bar{x}_j\bar{x}_k$ but not both.

Suppose a cubic term $a_{ijk}x_ix_jx_k$. Then setting $x_k = 0$ will cause this term to vanish.

If term $a_{ijk}\bar{x}_i\bar{x}_j\bar{x}_k$ occurs, then setting $x_k = 1$ will cause this term to vanish.

Thus, for this assignment of variables x_k, the function $f(\mathbf{x})$ reduces to $L + a_{ij}x_ix_j$. Since this must be submodular, $a_{ij} \leq 0$.

Graph representation of cubic functions.

Given a function of the form (**??**), we show how to represent the function as a graph. This is related closely to reducing the function in some sense to a quadratic function.

Handling the cubic terms: observe that for all values of x_i, x_j and x_k,

$$\begin{aligned} -x_i x_j x_k &= \min_y \, (\bar{x}_i + \bar{x}_j + \bar{x}_k - 1) y \\ &= \min_y \, (\bar{x}_i + \bar{x}_j + \bar{x}_k) y + \bar{y} - 1 \end{aligned}$$

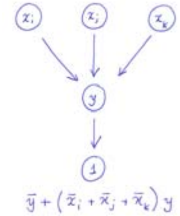
So, given a term $-a_{ijk} x_i x_j x_k$, with $a_{ijk} > 0$, replace by

$$a_{ijk} \, (\bar{x}_i y + \bar{x}_j y + \bar{x}_k y + \bar{y})$$

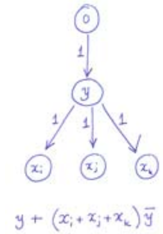
which differs by a constant.

Similarly, replace term $-a_{ijk} \bar{x}_i \bar{x}_j \bar{x}_k$, with $a_{ijk} > 0$ by

$$a_{ijk} \, (x_i \bar{y} + x_j \bar{y} + x_k \bar{y} + y) \ .$$



Graph for term $-x_i x_j x_k$



Graph for term $-\bar{x}_i \bar{x}_j \bar{x}_k$

Graph Representability.

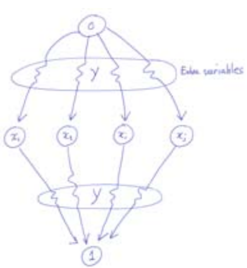
Consider $\mathcal{V} = \mathcal{V}_x + \mathcal{V}_y + \{0, 1\}$ where the vertices $\mathcal{V}_x$ correspond to vertices $x_1, \ldots, x_n$. The $\mathcal{V}_y$ vertices are extra vertices.

Definition The function $f(\mathbf{x})$ is *graph representable* if there is a graph $\mathcal{G} = (\mathcal{V} \cup \mathcal{Y}, \mathcal{W})$, such that for all x,

$$f(\mathbf{x}) = \min_{(\mathcal{Y}_0, \mathcal{Y}_1)} \mathrm{Cost}(\mathcal{Y}_0 \cup \mathcal{V}_0(\mathbf{x}), \mathcal{Y}_1 \cup \mathcal{V}_1(\mathbf{x}))$$

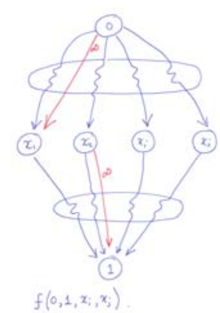
The minimum is taken over all partitions of $\mathcal{Y}$.

We have shown that quadratic and cubic pseudo-boolean functions are graph representable.

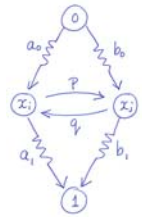


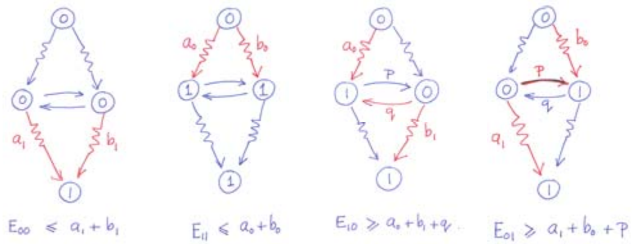
Graph Representable functions are submodular.

Step 1: Reduction to 2 variables.



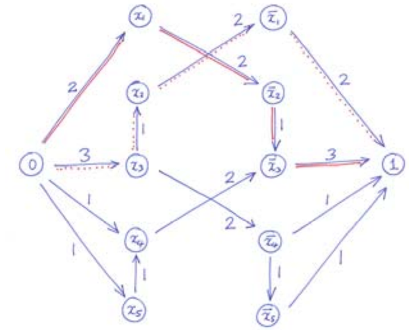
Proof for 2-variable functions





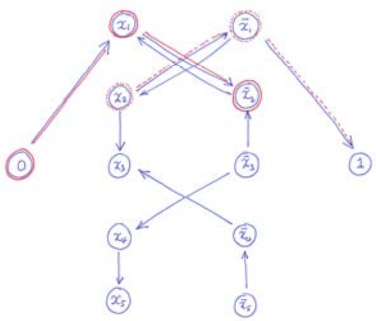
Graph representation of non-submodular functions.

First, we represent a general quadratic pseudo-boolean function.



$$2x_1 + 3x_3 + x_4 + x_5 + x_2\bar{x}_3 + x_4\bar{x}_5 + 2\bar{x}_1\bar{x}_2 + 2\bar{x}_3\bar{x}_4 + 2\bar{x}_3\bar{x}_4$$

After maximum-flow algorithm, we can reduce it to a form where there is no path from 0 to 1.

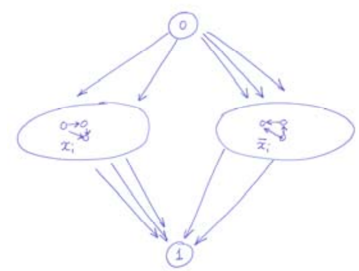


What happens for submodular functions?

For a submodular function

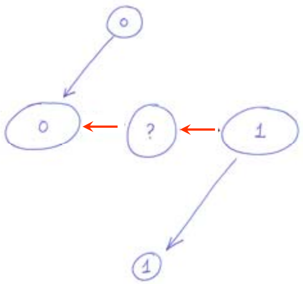
$$f(\mathbf{x}) = L + \sum_{1 \le i < j \le n} \bar{x}_i x_j$$

the graph separates into two parts, disjoint except at 0 and 1.



Max-flow provides a partial solution.

Nodes connected to the vertex 0 will have value 0 in any optimal solution.



Elimination.

Consider the problem of minimizing a pseudo-boolean function $f_0(x_1, x_2, \ldots, x_n)$ We may write

$$f_0(x_1, x_2, \ldots, x_n) = x_1 \,\Delta(x_2, \ldots, x_n) + h(x_2, \ldots, x_n) \ .$$

We call Δ the *derivative* of f_0 with respect to x_1.

Now, suppose values of $x_2, \ldots, x_n$ are given, and let

$$x_1^* = \text{argmin}_{x_1} f_0(x_1, x_2, \ldots, x_n) \ .$$

We see that

$$x_1^* = \begin{cases} 1 & \text{if } \Delta(x_2, \ldots, x_n) < 0 \\ 0 & \text{if } \Delta(x_2, \ldots, x_n) > 0 \end{cases}$$

and if $\Delta(x_2, \ldots, x_n) = 0$, then x_1^* can have either value.

Back substitution

Note, x_1^* is a function of $(x_2, \ldots, x_n)$. Substitute back x_1^* in f_C and define

$$\begin{aligned} f_1(x_2, \ldots, x_n) &= \min_{x_1} f_0(x_1, \ldots, x_n) \\ &= x_i^* \Delta(x_2, \ldots, x_n) + h(x_2, \ldots, x_n) \end{aligned}$$

and

$$\begin{aligned} \min_{x_1, \ldots, x_n} f_0(x_1, \ldots, x_n) &= \min_{x_2, \ldots, x_n} \min_{x_1} f_0(x_1, \ldots, x_n) \\ &= \min_{x_2, \ldots, x_n} f_1(x_2, \ldots, x_n) \ . \end{aligned}$$

Example.

$$\begin{aligned} f_0(x_1, x_2, x_3) &= 4x_1 + 3x_2 - 2x_3 + 2x_1x_2 - 5x_1x_3 + 7x_2x_3 + 3x_1x_2x_3 \\ &= x_1(4 + 2x_2 - 5x_3 + 3x_2x_3) + (3x_2 - 2x_3 + 7x_2x_3) \\ &= x_1\Delta(x_2, x_2) + h(x_2, x_3) \end{aligned}$$

Then, we may compute

x_2	x_3	Δ	x_1^*	$x_1^* \Delta$	term
0	0	4	0	0	$\bar{x}_2\bar{x}_3$
0	1	-1	1	0	$\bar{x}_2 x_3$
1	0	6	0	6	$x_2\bar{x}_3$
1	1	4	0	0	$x_2 x_3$

Hence,

$$x_i^* = 6x_2\bar{x}_3 = 6x_2 - 6x_2x_3 \ .$$

Substitute in $f_1(x_2, x_3)$ gives

$$f_1(x_2, x_3) = 9x_2 - 2x_3 + x_2x_3 = x_2(9 + x_3) - 2x_3 \ .$$

Example – continued

Continuing with

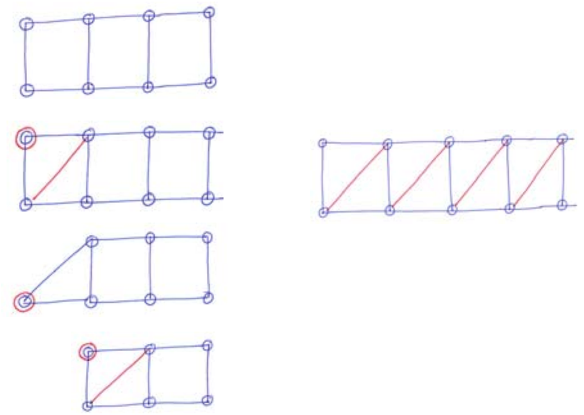
$$f_1(x_2.x_3) = x_2(9 + x_3) - 2x_3 \ .$$

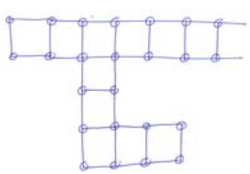
Then

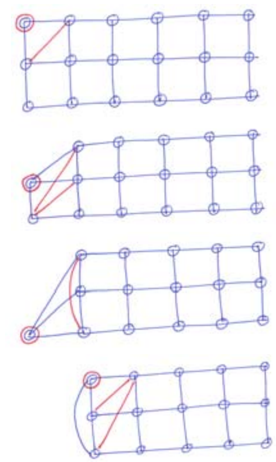
x_3	Δ	x_2^*	$x_2^* \Delta$	term
0	8	0	0	$\bar{x}_3$
1	9	0	0	x_3

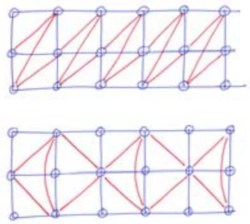
Hence, $f_2(x_3) = -2x_3$. This is minimized when $x_3^* = 1$, and the minimum is -2. Subsituting back gives $x_2^* = 0$, and $x_1^* = 1$.

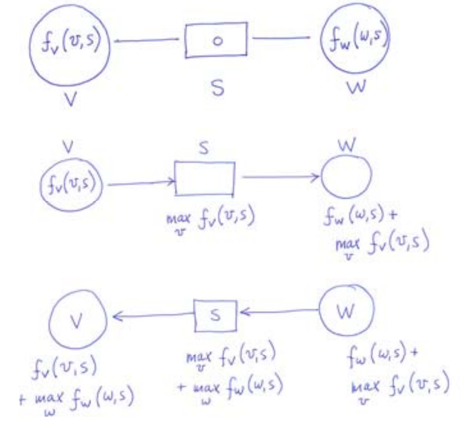
Elimination.











$$\max_{v,s}\left(f_v(v,s)+\max_w f_w(w,s)\right)$$

$$=\max_s\ \max_v\left(f_v(v,s)+\max_w f_w(w,s)\right)$$

$$=\max_s\left(\max_v f_v(v,s)+\max_w f_w(w,s)\right)$$

$$=\max_s\ \max_{v,w}\left(f_v(v,s)+f_w(w,s)\right)$$

$$=\max_{v,w,s}\left(f_v(v,s)+f_w(w,s)\right).$$