Linear Programming Decoding

Lecture Notes for: Graphical Models and Inference
Henry D. Pfister

Ocobter 7th, 2014 (rev. 1)

1 Introduction

Linear Programming Decoding was introduced by Feldman, Wainwright, and Karger in [1]. A very similar approach based on graph covers was derived independently by Vontobel and Koetter [2]. The relationship between factor graphs and message passing also leads to iterative algorithms that solve these linear programs efficiently [3, 4, 5]. There is also a long history of using linear-programming relaxations to approximate the solutions of combinatorial problems. An introductory overview can be found in

`http://theory.stanford.edu/~trevisan/books/cs261.pdf`

2 Best Assignment Problem

Let $\mathcal{X}$ be a finite alphabet and $f : \mathcal{X}^n \to \mathbb{R}$ be a function. The best assignment problem consists of finding an assignment, x_1^n, in the set of best assignments

$$B = \arg \max_{x_1^n \in \mathcal{X}^n} f(x_1, \ldots, x_n).$$

In general, one has to check all $|\mathcal{X}|^n$ possibilities. If f factors though, then there is an approximation that is sometimes exact.

Let $(V \cup F, E)$ be a bipartite graph where the vertices in $V = \{1, 2, \ldots, n\}$ are associated with variable nodes, the vertices in F are associated with factor nodes, and the edges $E \subseteq V \times F$ defines which variables are involved in which factors. Abusing notation, we let $V(a)$ denote the set of variable nodes adjacent to the factor node $a \in F$ and $F(i)$ denote the set of factor nodes adjacent to variable node $i \in V$. Based on this factor graph we assume that f factors into the product

$$f(x_1, x_2, \ldots, x_n) = \prod_{a \in F} f_a(x_{V(a)}).$$

Example 2.1. Consider the binary linear code with parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Since this matrix has full rank over $\mathbb{F}_2$, the code contains only the all-zero vector.

Example 2.2. Consider the MAX-SAT version of the the boolean satisfiability problem where the goal is to maximize the number of satisfied clauses. In this case, there are n literals $(x_1, \ldots, x_n) \in \{0, 1\}^n$ and a factor for each clause. If clause $a \in F$ is satisfied, then the factor $f_a(x_{V(a)}) = 2$ and otherwise it equals 1. Thus, if x_1^n satisfies S clauses, then $f(x_1, \ldots, x_n) = 2^S$. Of couse, a the boolean system is satisfiable iff the value of MAX-SAT equals 2^n.

2.1 Integer Linear Program

Now, we will transform the best assignment problem into an integer linear program (ILP). First , we observe that

$$\begin{aligned} B &= \arg\max_{x_1^n \in \mathcal{X}^n} \ln f(x_1, \ldots, x_n) \\ &= \arg\max_{x_1^n \in \mathcal{X}^n} \ln \prod_{a \in F} f_a(x_{V(a)}) \\ &= \arg\max_{x_1^n \in \mathcal{X}^n} \sum_{a \in F} \ln f_a(x_{V(a)}). \end{aligned}$$

To make this objective function linear, we can change variables from x_1^n to a set of variable indicator functions that encode the value of x_1^n. First, consider the set of variable indicator functions $\underline{q} \triangleq \{q_i(\cdot)\}_{i\in[n]}$ where $q_i : \mathcal{X} \to \{0,1\}$ satisfies $\sum_{z\in\mathcal{X}} q_i(z) = 1$. One can map from x_1^n to $\{q_i(\cdot)\}_{i\in[n]}$ by choosing

$$q_i(z) = \begin{cases} 1 & \text{if } z = x_i \\ 0 & \text{if } z \neq x_i. \end{cases}$$

Since there are only $|\mathcal{X}|$ valid $q_i(\cdot)$ functions, this mapping is one-to-one and invertible. This allows one to rewrite the ath term in the objective function as

$$\ln f_a(x_{V(a)}) = \sum_{\underline{z} \in \mathcal{X}^{|V(a)|}} \left(\prod_{i \in V(a)} q_i(z_i) \right) \ln f_a(\underline{z})$$

but these terms are not a linear functions of $\underline{q}$.

To get linearity, we introduce the set of factor indicator functions $\underline{\hat{q}} \triangleq \{\hat{q}_a(\cdot)\}_{a\in F}$ where $\hat{q}_a : \mathcal{X}^{|V(a)|} \to \{0,1\}$ satisfies $\sum_{\underline{z}\in\mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) = 1$. One can map from x_1^n to $\{\hat{q}_a(\cdot)\}_{a\in F}$ by choosing

$$\hat{q}_a(\underline{z}) = \prod_{i \in V(a)} q_i(z_i) = \begin{cases} 1 & \text{if } \underline{z} = x_{V(a)} \\ 0 & \text{if } \underline{z} \neq x_{V(a)}. \end{cases}$$

This allows one to rewrite the ath term in the objective function as

$$\ln f_a(x_{V(a)}) = \sum_{\underline{z} \in \mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) \ln f_a(\underline{z}),$$

which a linear function of $\underline{\hat{q}}$. The mapping from x_1^n to $\underline{\hat{q}}$ is not surjective, however, because each $\hat{q}_a(\cdot)$ function automatically satisfies the consistency condition

$$\sum_{z_{V(a)} \in \mathcal{X}^{|V(a)|}} \delta_{z_i, z} \hat{q}_a(z_{V(a)}) = \begin{cases} 1 & \text{if } z = x_i \\ 0 & \text{if } z \neq x_i. \end{cases}$$

Therefore, we define the set of *consistent indicator functions,*

$$\mathcal{M}^* \triangleq \left\{ \{q_i(\cdot)\}_{i\in V}, \{\hat{q}_a(\cdot)\}_{a\in F} \,\middle|\, q_i : \mathcal{X} \to \{0,1\}, \hat{q}_a : \mathcal{X}^{|V(a)|} \to \{0,1\}, \right. \\ \left. \sum_{z\in\mathcal{X}} q_i(z) = 1, \sum_{\underline{z}\in\mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) = 1, \sum_{z_{V(a)}\in\mathcal{X}^{|V(a)|}} \delta_{z_i,z} \hat{q}_a(z_{V(a)}) = q_i(z), i \in V, a \in F \right\},$$

and note that any element of $\mathcal{M}^*$ can be represented (e.g., on a computer) by $n + \sum_{a\in F} \mathcal{X}^{|V(a)|}$ binary variables. The previously defined mappings from $\mathcal{X}^n$ to $\underline{q}$ and $\underline{\hat{q}}$ also show that there is a one-to-one correpondence between $\mathcal{X}^n$ and $\mathcal{M}^*$. Thus, we find that there is also a one-to-one corespondence between the set of best assignments,

$$B = \arg\max_{x_1^n \in \mathcal{X}^n} \sum_{a \in F} \ln f_a(x_{V(a)}),$$

and the set of $(\underline{q}, \underline{\hat{q}})$ indicator functions of best assignments,

$$B' = \arg \max_{(\underline{q},\underline{\hat{q}})\in\mathcal{M}^*} \sum_{a\in F} \sum_{\underline{z}\in\mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) \ln f_a(\underline{z}).$$

The second formulation is an ILP because the the objective function is linear in $\underline{\hat{q}}$ and the feasible set is a subset of integers generated by linear constraints. This ILP has $n + \sum_{a\in F} \mathcal{X}^{|V(a)|}$ binary variables and $n + |E|$ linear equality constraints.

Example 2.3. Consider the binary linear code from Example 2.1 where the factor node $a \in F = \{1, 2, 3\}$ is identified with row a of the matrix. In this case, points in $\mathcal{M}^*$ consist of the functions

$$q_1(z_1), q_2(z_2), q_3(z_3), \hat{q}_1(z_1, z_2), \hat{q}_2(z_1, z_2, z_3), \hat{q}_3(z_2, z_3)$$

and the only point in $\mathcal{M}^*$ is defined by $q_j(0) = 1$ for $j \in \{1, 2, 3\}$ and $\hat{q}_a(\underline{0}) = 1$ for $a \in \{1, 2, 3\}$.

2.2 Linear Program and the Marginal Polytope

Any linear integer program can be relaxed into a linear program (LP) by relaxing the integer constraint. Let $\mathcal{M}$ denote the *marginal polytope* defined by

$$\mathcal{M} \triangleq \Bigg\{ \{q_i(\cdot)\}_{i\in V]}, \{\hat{q}_a(\cdot)\}_{a\in F} \,\Bigg|\, q_i : \mathcal{X} \to [0, 1], \hat{q}_a : \mathcal{X}^{|V(a)|} \to \{[0, 1],$$
$$\sum_{z\in\mathcal{X}} q_i(z) = 1, \sum_{\underline{z}\in\mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) = 1, \sum_{z_{V(a)}\in\mathcal{X}^{|V(a)|}} \delta_{z_i,z}\hat{q}_a(z_{V(a)}) = q_i(z), i \in V, a \in F \Bigg\}.$$

One can think of $\hat{q}_a(\cdot)$ as a hypothetical joint distribution for the subset of variables attached to factor a. In this case, $q_i(\cdot)$ represents the marginal distribution of x_i and, for each $a \in V(i)$, the consistency constraint,

$$\sum_{z_{V(a)}\in\mathcal{X}^{|V(a)|}} \delta_{z_i,z}\hat{q}_a(z_{V(a)}) = q_i(z),$$

requires that these marginals agree.

The LP relaxation of the best assignment problem consists of finding a vector in the set

$$B_{LP} = \arg \max_{(\underline{q},\underline{\hat{q}})\in\mathcal{M}} \sum_{a\in F} \sum_{\underline{z}\in\mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) \ln f_a(\underline{z}).$$

This optimization can be written as an LP with $n + \sum_{a\in F} \mathcal{X}^{|V(a)|}$ variables on $[0, 1]$ and $n + |E|$ linear equality constraints. Since $\mathcal{M}^* = \big\{(\underline{q}, \underline{\hat{q}}) \in \mathcal{M} \,|\, q_i(z) \in \{0, 1\}, \hat{q}_a(\underline{z}) \in \{0, 1\}, i \in V, a \in F\big\}$ is the set of integer-valued functions inside of $\mathcal{M}$, any LP solution with integer coordinates must also be in B. Thus, we say that the LP relaxation provides optimality certificate. If one has an LP optimum with integer values, then it is a best assignment.

2.3 Reduced Marginal Polytope

It is worth noting that the marginal polytope only depends on the factor graph structure and not the factors themselves. If some of the factors can take the value zero, then one can reduce the polytope without affecting the LP. In particular, if $f_a(\underline{z}) = 0$ and $\hat{q}_a(\underline{z}) > 0$ for some $a \in F$ and $\underline{z} \in \mathcal{X}^{|V(a)|}$, then the objective function is equal to minus infinity. Therefore, we can assume wolog that $\hat{q}_a(\underline{z}) = 0$ in this case. This allows us to define the *reduced marginal polytope*

$$\mathcal{M}' \triangleq \Big\{(\underline{q}, \underline{\hat{q}}) \in \mathcal{M} \,|\, \forall(a, \underline{z}) \in F \times \mathcal{X}^{|V(a)|}, \hat{q}_a(\underline{z}) = 0 \text{ if } f_a(\underline{z}) = 0\Big\}$$

and say that

$$B_{LP} = \arg \max_{(\underline{q},\underline{\hat{q}})\in\mathcal{M}'} \sum_{a\in F} \sum_{\underline{z}\in\mathcal{X}^{|V(a)|}} \hat{q}_a(\underline{z}) \ln f_a(\underline{z})$$

as long as $f(x_1, \ldots, x_n) > 0$ for some $x_1^n \in \mathcal{X}^n$. This equality holds because the objective function equals $-\infty$ for all $(\underline{q}, \underline{\hat{q}}) \in \mathcal{M}\backslash\mathcal{M}'$.

Example 2.4. Consider the binary linear code from Example 2.1. The factors f_1, f_2, f_3 take the value 0 whenever their arguments contain an odd number of ones. Thus, the marginal polytope is reduced by the equality constraints

$$\begin{aligned} 0 =& \hat{q}_1(0,1) = \hat{q}_1(1,0) \\ 0 =& \hat{q}_2(1,0,0) = \hat{q}_2(0,1,0) = \hat{q}_2(0,0,1) = \hat{q}_2(1,1,1) \\ 0 =& \hat{q}_3(0,1) = \hat{q}_3(1,0). \end{aligned}$$

3 Zero-One Factors

3.1 Projected Reduced Marginal Polytope

Now, suppose that the function $f(x_1, \ldots, x_n)$ is an indicator function for valid assignments and that X_1^n is uniform over all valid assignments. Furthermore, let Y_1^n be a noisy observation of X_1^n where $\Pr(Y_i = y_i | X_i = x_i) = W(y_i|x_i)$. Let the set of variable marginals be $\mathcal{V} \triangleq \{q_i : \mathcal{X} \to [0,1] \,|\, q_j(z) = 1, j \in V\}$ and $\pi : \mathcal{M} \to \mathcal{V}$ be the projection defined by $(\underline{q}, \underline{\hat{q}}) \mapsto \underline{q}$ for $(\underline{q}, \underline{\hat{q}}) \in \mathcal{M}$. Then, the best assignment linear program reduces to

$$\begin{aligned} \overline{B}_{LP}(y_1^n) &= \pi\left(\arg\max_{(\underline{q},\underline{\hat{q}})\in\mathcal{M}'} \sum_{i=1}^{n} \sum_{z\in\mathcal{X}} q_i(z) \ln W(y_i|z)\right) \\ &= \arg\max_{\underline{q}\in\overline{\mathcal{M}}} \sum_{i=1}^{n} \sum_{z\in\mathcal{X}} q_i(z) \ln W(y_i|z), \end{aligned}$$

where the polytope $\overline{\mathcal{M}} = \pi(\mathcal{M}')$ is the projection of $\mathcal{M}'$ onto $\mathcal{V}$. A vertex $\underline{q} \in \overline{\mathcal{M}}$ is called a pseudo-codeword if $q_j(z) \notin \{0,1\}$ for any $j \in V$ and $z \in \mathcal{X}$. The LP relaxation is tight if the maximum of the objective fuction does not occur at a pseudo-codeword.

The polytope $\overline{\mathcal{M}}$ has a natural geometric interpretation in this case. Each factor constraint, f_a, defines a set of locally valid configurations $\mathcal{L}_a = \left\{z_{V(a)} \in \mathcal{X}^{|V(a)|} \,|\, f_a(z_{V(a)}) = 1\right\}$ and, for any $a \in F$ and $i \in V(a)$, the variable-node marginal function is given by

$$q_i(z) = \sum_{z_{V(a)}\in\mathcal{L}_a} \hat{q}_a(z_{V(a)}) \delta_{z_i,z}.$$

One can interpret this equation as first embedding each of the locally valid configurations into an $|\mathcal{X}|^{|V(a)|}$-dimensional space as unit vectors, then taking the convex hull of these vectors, and finally projecting down onto the ith coordinate marginal. Thus, the constraint imposed on the LP by the factor node f_a is that the $q_i(z)$ functions lie in the convex hull of the locally valid configurations.

Example 3.1. Consider the binary linear code from Example 2.1. Consider the point $(\underline{q}, \underline{\hat{q}}) \in \mathcal{M}$ defined by

$$\hat{q}_a(z_1, z_2) = \begin{cases} 2/3 & \text{if } z_1 = z_2 = 1 \\ 1/3 & \text{if } z_1 = z_2 = 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$\hat{q}_b(z_1, z_2, z_3) = \begin{cases} 1/3 & \text{if } (z_1, z_2, z_3) \in \{110, 101, 011\} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\hat{q}_c(z_2, z_3) = \begin{cases} 2/3 & \text{if } z_2 = z_3 = 1 \\ 1/3 & \text{if } z_2 = z_3 = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The variable node marginals are defined by $q_1(1) = q_2(1) = q_3(1) = 2/3$ and one can verify that this is pseudo-codeword.

3.2 Binary Variables

For binary variables (i.e., $\mathcal{X} = \{0,1\}$), the normalization constraint implies that $q_i(1) = 1 - q_i(0)$ and the LP can be formulated in terms of the vector $\underline{u} = (u_1, u_2, \ldots, u_n)$ defined by $u_i = q_i(1)$. Thus, we can write

$$\begin{aligned}
\overline{B}_{LP}(y_1^n) &= \arg\max_{\underline{q}\in\overline{\mathcal{M}}} \sum_{i=1}^{n} \sum_{z\in\mathcal{X}} q_i(z) \ln W(y_i|z) \\
&= \arg\max_{\underline{q}\in\overline{\mathcal{M}}} \sum_{i=1}^{n} \left(q_i(0)\ln W(y_i|0) + q_i(1)\ln W(y_i|1)\right) \\
&= \arg\max_{\underline{u}\in\overline{\mathcal{Q}}} \sum_{i=1}^{n} \left((1-u_i)\ln W(y_i|0) + u_i \ln W(y_i|1)\right) \\
&= \arg\max_{\underline{u}\in\overline{\mathcal{Q}}} \sum_{i=1}^{n} u_i \ln \frac{W(y_i|1)}{W(y_i|0)} \\
&= \arg\min_{\underline{u}\in\overline{\mathcal{Q}}} \sum_{i=1}^{n} u_i \ln \frac{W(y_i|0)}{W(y_i|1)},
\end{aligned}$$

where $\overline{\mathcal{Q}} = \left\{ u_1^n \in [0,1]^n \,\middle|\, \underline{q} \in \overline{\mathcal{M}}, u_i = q_i(1), i \in [n] \right\}$. The geometric interpretation of the polytope simplifies because the alphabet $\{0,1\}$ has a natural embedding into $[0,1]$. With this embedding, the constraint regions associated with each factor are defined by

$$\mathcal{Q}_a \triangleq \text{conv}\left(\left\{ u_1^n \in [0,1]^n \,\middle|\, u_{V(a)} \in \mathcal{L}_a \right\} \right)$$

and a vector $u_1^n \in \{0,1\}^n$ satisfies all of the constraints iff it lies in the intersection

$$\overline{\mathcal{Q}} = \cap_{a\in F} \mathcal{Q}_a = \left\{ u_1^n \in [0,1]^n \,\middle|\, u_{V(a)} \in \text{conv}(\mathcal{L}_a) \right\}$$

.

Example 3.2. Consider a zero-one factor $f_a(x_1, \ldots, x_d)$ with binary variables that disallows exactly one local configuration $z_1, \ldots, z_d$. Then, the convex hull of the valid configurations can be found by starting with the box constraints $0 \le u_i \le 1$ for $i = 1, 2 \ldots, d$ and then using a single hyperplane to slice off the invalid configuration. The hyperplane must contain all the vertices adjacent to $z_1, \ldots, z_d$ that will not be removed. Thus, the normal vector of the desired hyperplane equals the vector from the center of the Hamming cube to the undesired vertex, $(w_1, \ldots, w_d) = (z_1, \ldots, z_d) - (\frac{1}{2}, \ldots \frac{1}{2})$. The desired inequality, $\sum_{i=1}^{d} w_i u_i \le c$, can be seen to project u onto the normal and then test its distance from 0. The constant c can be computed by making sure that we don't slice off the desired confguration $(1 - z_1, z_2, \ldots, z_d)$ and this gives

$$\begin{aligned}
c &= w_1(1-z_1) + \sum_{i=2}^{d} w_i z_i \\
&= \left(z_1 - \frac{1}{2}\right)(1-z_1) + \sum_{i=2}^{d} \left(z_i - \frac{1}{2}\right) z_i \\
&= \frac{3}{2} z_1 - z_1^2 - \frac{1}{2} + \sum_{i=2}^{d} z_i^2 - \frac{1}{2}\sum_{i=2}^{d} z_i \\
&= \frac{3}{2} z_1 - z_1 - \frac{1}{2} + \sum_{i=2}^{d} z_i - \frac{1}{2}\sum_{i=2}^{d} z_i \\
&= -\frac{1}{2} + \frac{1}{2}\sum_{i=1}^{d} z_i.
\end{aligned}$$

The resulting inequality can be rewritten as $\sum_{i=1}^{d}(2z_i - 1)u_i \le -1 + \sum_{i=1}^{d} z_i$.

3.3 Decoding Binary Linear Codes

For a binary linear code, all factors represent even-parity constraints. So, the first step is to construct the local codeword polytope for the even-parity code. In the Hamming cube, every vertex of odd weight is adjacent only to vertices of even weight. Thus, we can use the method of Example 3.2 to slice off all the odd weight vertices because the cutting planes always pass through even weight vertices[1]. For a degree-d parity check, the local codeword polytope is given by

$$\left\{ u_1^d \in [0,1]^d \;\middle|\; \textstyle\sum_{i=1}^d (2z_i - 1)u_i \leq -1 + \sum_{i=1}^d z_i, (z_1, \ldots, z_d) \in \{0,1\}^d, \sum_{i=1}^d z_i \text{ is odd} \right\}.$$

Example 3.3. Consider the binary linear code from Example 2.1. Since this code contains the all-zero vector (e.g., it is linear), the vector $(0, \ldots, 0)$ is in $\overline{Q}$ and the value of the LP,

$$\arg\min_{\underline{u} \in \overline{Q}} \sum_{i=1}^{n} u_i \ln \frac{W(y_i|0)}{W(y_i|1)},$$

is upper bounded by 0. Thus, if the zero codeword is transmitted, then the LP decoder recovers the correct codeword if there are no points in $\overline{Q}$ with negative value. Under what condition does the previously noted pseudo-codeword $(\frac{2}{3}, \frac{2}{3}, \frac{2}{3}) \in \overline{Q}$ beat the all-zero codeword?

References

[1] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, pp. 954–972, March 2005.

[2] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," 2005. submitted to *IEEE Trans. on Inform. Theory.*

[3] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on trees: Message-passing and linear programming," *IEEE Trans. Inform. Theory*, vol. 51, pp. 3697–3717, Nov. 2005.

[4] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *Advances in Neural Information Processing Systems 20*, pp. 553–560, MIT Press, 2007.

[5] P. O. Vontobel and R. Koetter, "On low-complexity linear-programming decoding of LDPC codes," *Eur. Trans. Telecom.*, vol. 18, pp. 509–517, Aug. 2007.

[1] In general, the inequality used to slice off undesired vertices may depend on the previous slices.