An Augmented Lagrangian Aproach to Constrained MAP Inference
Supplementary Material

A. Derivation of QUAD for Binary Pairwise Factors

In this section, we derive in detail the closed form solution of problem (12) for binary pairwise factors (Sect. 4.1). Recall that the marginal polytope $\mathcal{M}(\mathcal{G}_a)$ is given by:

$$\mathcal{M}(\mathcal{G}_a) = \left\{ (\boldsymbol{\nu}^a_{\mathcal{N}(a)}, \boldsymbol{\nu}_a) \;\middle|\; \begin{array}{ll} \sum_{x_i \in \mathcal{X}_i} \nu^a_i(x_i) = 1, & \forall i \in \mathcal{N}(a) \\ \nu^a_i(x_i) = \sum_{\boldsymbol{x}_a \sim x_i} \nu_a(\boldsymbol{x}_a), & \forall i \in \mathcal{N}(a), x_i \in \mathcal{X}_i \\ \nu_a(\boldsymbol{x}_a) \geq 0, & \forall \boldsymbol{x}_a \in \mathcal{X}_a \end{array} \right\}. \tag{13}$$

If factor a is binary and pairwise ($|\mathcal{N}(a)| = 2$), we may reparameterize our problem by introducing new variables $z_1 \triangleq \nu^a_1(1)$, $z_2 \triangleq \nu^a_2(1)$, and $z_{12} \triangleq \nu_a(1,1)$. Noting that $\boldsymbol{\nu}^a_1 = (1-z_1, z_1)$, $\boldsymbol{\nu}^a_2 = (1-z_2, z_2)$, and $\boldsymbol{\nu}_a = (1 - z_1 - z_2 + z_{12}, z_1 - z_{12}, z_2 - z_{12}, z_{12})$, problem (12) becomes

$$\begin{array}{ll} \displaystyle\min_{z_1, z_2, z_{12}} & \frac{\eta_t}{2}[(1 - z_1 - \eta_t^{-1}\omega^a_1(0))^2 + (z_1 - \eta_t^{-1}\omega^a_1(1))^2 + (1 - z_2 - \eta_t^{-1}\omega^a_2(0))^2 + (z_2 - \eta_t^{-1}\omega^a_2(1))^2] \\ & -\phi_a(00)(1 - z_1 - z_2 + z_{12}) - \phi_a(10)(z_1 - z_{12}) - \phi_a(01)(z_2 - z_{12}) - \phi_a(11)z_{12} \\ \text{s.t.} & z_{12} \leq z_1, \quad z_{12} \leq z_2, \quad z_{12} \geq z_1 + z_2 - 1, \quad (z_1, z_2, z_{12}) \in [0,1]^3 \end{array} \tag{14}$$

or, multiplying the objective by the constant $1/(2\eta_t)$:

$$\begin{array}{ll} \displaystyle\min_{z_1, z_2, z_{12}} & \frac{1}{2}(z_1 - c_1)^2 + \frac{1}{2}(z_2 - c_2)^2 - c_{12}z_{12} \\ \text{s.t.} & z_{12} \leq z_1, \quad z_{12} \leq z_2, \quad z_{12} \geq z_1 + z_2 - 1, \quad (z_1, z_2, z_{12}) \in [0,1]^3, \end{array} \tag{15}$$

where we have substituted

$$c_1 = (\eta_t^{-1}\omega^a_1(1) + 1 - \eta_t^{-1}\omega^a_1(0) + \eta_t^{-1}\phi_a(00) - \eta_t^{-1}\phi_a(10))/2 \tag{16}$$

$$c_2 = (\eta_t^{-1}\omega^a_2(1) + 1 - \eta_t^{-1}\omega^a_2(0) + \eta_t^{-1}\phi_a(00) - \eta_t^{-1}\phi_a(01))/2 \tag{17}$$

$$c_{12} = (\eta_t^{-1}\phi_a(00) - \eta_t^{-1}\phi_a(10) - \eta_t^{-1}\phi_a(00) + \eta_t^{-1}\phi_a(11))/2. \tag{18}$$

Now, notice that in (15) we can assume $c_{12} \geq 0$ without loss of generality—indeed, if $c_{12} < 0$, we recover this case by redefining $c'_1 = c_1 + c_{12}$, $c'_2 = 1 - c_2$, $c'_{12} = -c_{12}$, $z'_2 = 1 - z_2$, $z'_{12} = z_1 - z_{12}$. Thus, assuming that $c_{12} \geq 0$, the lower bound constraints $z_{12} \geq z_1 + z_2 - 1$ and $z_{12} \geq 0$ are always innactive and can be ignored. Hence, (15) can be simplified to:

$$\begin{array}{ll} \displaystyle\min_{z_1, z_2, z_{12}} & \frac{1}{2}(z_1 - c_1)^2 + \frac{1}{2}(z_2 - c_2)^2 - c_{12}z_{12} \\ \text{s.t.} & z_{12} \leq z_1, \quad z_{12} \leq z_2, \quad z_1 \in [0,1], \quad z_2 \in [0,1]. \end{array} \tag{19}$$

Second, if $c_{12} = 0$, the problem becomes separable, and the solution is

$$z_1^* = [c_1]_{\mathbb{U}}, \quad z_2^* = [c_2]_{\mathbb{U}}, \quad z_{12}^* = \min\{z_1^*, z_2^*\}, \tag{20}$$

where $[x]_{\mathbb{U}} = \min\{\max\{x, 0\}, 1\}$ is the projection (clipping) onto the unit interval. We next analyze the case where $c_{12} > 0$. The Lagrangian function of (19) is:

$$\begin{array}{rcl} L(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\nu}) & = & \frac{1}{2}(z_1 - c_1)^2 + \frac{1}{2}(z_2 - c_2)^2 - c_{12}z_{12} + \mu_1(z_{12} - z_1) + \mu_2(z_{12} - z_2) \\ & & -\lambda_1 z_1 - \lambda_2 z_2 + \nu_1(z_1 - 1) + \nu_2(z_2 - 1). \end{array} \tag{21}$$

At optimality, the following KKT conditions need to be satisfied:

$$\nabla_{z_1} L(\boldsymbol{z}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0 \quad \Rightarrow \quad z_1^* = c_1 + \mu_1^* + \lambda_1^* - \nu_1^* \tag{22}$$

$$\nabla_{z_2} L(\boldsymbol{z}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0 \quad \Rightarrow \quad z_2^* = c_2 + \mu_2^* + \lambda_2^* - \nu_2^* \tag{23}$$

$$\nabla_{z_{12}} L(\boldsymbol{z}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0 \quad \Rightarrow \quad c_{12} = \mu_1^* + \mu_2^* \tag{24}$$

$$\lambda_1^* z_1^* = 0 \tag{25}$$

$$\lambda_2^* z_2^* = 0 \tag{26}$$

$$\mu_1^*(z_{12}^* - z_1^*) = 0 \tag{27}$$

$$\mu_2^*(z_{12}^* - z_2^*) = 0 \tag{28}$$

$$\nu_1^*(z_1^* - 1) = 0 \tag{29}$$

$$\nu_2^*(z_2^* - 1) = 0 \tag{30}$$

$$\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^* \geq 0 \tag{31}$$

$$z_{12}^* \leq z_1^*, \quad z_{12}^* \leq z_2^*, \quad z_1^* \in [0,1], \quad z_2^* \in [0,1] \tag{32}$$

We are going to consider three cases separately:

1. $\boxed{z_1^* > z_2^*}$

   From the primal feasibility conditions (32), this implies $z_1^* > 0$, $z_2^* < 1$, and $z_{12}^* < z_1^*$. Complementary slackness (25,30,27) implies in turn $\lambda_1^* = 0$, $\nu_2^* = 0$, and $\mu_1^* = 0$. From (24) we have $\mu_2^* = c_{12}$. Since we are assuming $c_{12} > 0$, we then have $\mu_2^* > 0$, and complementary slackness (28) implies $z_{12}^* = z_2^*$.

   Plugging the above into (22)–(23) we obtain

   $$z_1^* = c_1 - \nu_1^* \leq c_1, \qquad z_2^* = c_2 + \lambda_2^* + c_{12} \geq c_2 + c_{12}. \tag{33}$$

   Now we have the following:

   - Either $z_1^* = 1$ or $z_1^* < 1$. In the latter case, $\nu_1^* = 0$ by complementary slackness (29), hence $z_1^* = c_1$. Since in any case we must have $z_1^* \leq c_1$, we conclude that $z_1^* = \min\{c_1, 1\}$.
   - Either $z_2^* = 0$ or $z_2^* > 0$. In the latter case, $\lambda_2^* = 0$ by complementary slackness (26), hence $z_2^* = c_2 + c_{12}$. Since in any case we must have $z_2^* \geq \lambda_2$, we conclude that $z_2^* = \max\{0, c_2 + c_{12}\}$.

   In sum:

   $$z_1^* = \min\{c_1, 1\}, \quad z_{12}^* = z_2^* = \max\{0, c_2 + c_{12}\}, \tag{34}$$

   and our assumption $z_1^* > z_2^*$ can only be valid if $c_1 > c_2 + c_{12}$.

2. $\boxed{z_1^* < z_2^*}$

   By symmetry, we have

   $$z_2^* = \min\{c_2, 1\}, \quad z_{12}^* = z_1^* = \max\{0, c_1 + c_{12}\}, \tag{35}$$

   and our assumption $z_1^* < z_2^*$ can only be valid if $c_2 > c_1 + c_{12}$.

3. $\boxed{z_1^* = z_2^*}$

   In this case, it is easy to verify that we must have $z_{12}^* = z_1^* = z_2^*$, and we can rewrite our optimization problem in terms of one variable only (call it z). The problem becomes that of minimizing $\frac{1}{2}(z-c_1)^2 + \frac{1}{2}(z-c_2)^2 - c_{12}z$, which equals a constant plus $(z - \frac{c_1+c_2+c_{12}}{2})^2$, subject to $z \in \mathbb{U} \triangleq [0,1]$. Hence:

   $$z_{12}^* = z_1^* = z_2^* = \left[\frac{c_1 + c_2 + c_{12}}{2}\right]_{\mathbb{U}}. \tag{36}$$

Putting all the pieces together, we have the following solution assuming $c_{12} \geq 0$:

$$z_{12}^* = \min\{z_1^*, z_2^*\}, \qquad (z_1^*, z_2^*) = \begin{cases} ([c_1]_{\mathbb{U}}, & [c_2 + c_{12}]_{\mathbb{U}}) & \text{if } c_1 > c_2 + c_{12} \\ ([c_1 + c_{12}]_{\mathbb{U}}, & [c_2]_{\mathbb{U}}) & \text{if } c_2 > c_1 + c_{12} \\ ([(c_1 + c_2 + c_{12})/2]_{\mathbb{U}}, & [(c_1 + c_2 + c_{12})/2]_{\mathbb{U}}) & \text{otherwise.} \end{cases} \tag{37}$$

B. Derivation of QUAD for Several Hard Constraint Factors

In this section, we consider hard constraint factors with binary variables (Sect. 4.2). These are factors whose log-potentials are indicator functions, *i.e.*, they can be written as $\phi_a : \{0,1\}^m \to \bar{\mathbb{R}}$ with

$$\phi_a(\boldsymbol{x}_a) = \begin{cases} 0 & \text{if } \boldsymbol{x}_a \in \mathcal{S}_a \\ -\infty & \text{otherwise,} \end{cases} \tag{38}$$

where $\mathcal{S}_a \subseteq \{0,1\}^m$ is the acceptance set. Since any probability distribution over $\mathcal{S}_a$ has to assign zero mass to points not in $\mathcal{S}_a$, this choice of potential will always lead to $\nu_a(\boldsymbol{x}_a) = 0, \forall \boldsymbol{x}_a \notin \mathcal{S}_a$. Also, because the variables are binary, we always have $\nu_i^a(0) + \nu_i^a(1) = 1$. In fact, if we introduce the set

$$\mathcal{Z}_a \triangleq \{(\nu_1^a(1), \ldots, \nu_m^a(1)) \mid (\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a) \text{ for some } \boldsymbol{\nu}_a \text{ s.t. } \nu_a(\boldsymbol{x}_a) = 0, \forall \boldsymbol{x}_a \notin \mathcal{S}_a\} \tag{39}$$

we have that the two following optimization problems are equivalent for any function f:

$$\min_{\substack{(\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a)\\ \in \mathcal{M}(\mathcal{G}_a)}} f(\boldsymbol{\nu}^a_{N(a)}) + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \quad = \quad \min_{\boldsymbol{z} \in \mathcal{Z}_a} \tilde{f}(\boldsymbol{z}), \tag{40}$$

where $\tilde{f}(z_1, \ldots, z_m) \triangleq f(1 - z_1, z_1, \ldots, 1 - z_m, z_m)$. Hence the set $\mathcal{Z}_a$ can be used as a "replacement" of the marginal polytope $\mathcal{M}(\mathcal{G}_a)$. By abuse of language, we will sometimes refer to $\mathcal{Z}_a$ (which is also a polytope) as "the marginal polytope of $\mathcal{G}_a$." As a particularization of (40), we have that the quadratic problem (12) becomes that of computing a projection onto $\mathcal{Z}_a$. Of particular interest is the following result.

Proposition 3 *We have* $\mathcal{Z}_a = \operatorname{conv} \mathcal{S}_a$.

Proof: From the definition of $\mathcal{M}(\mathcal{G}_a)$ and the fact that we are constraining $\nu_a(\boldsymbol{x}_a) = 0, \forall \boldsymbol{x}_a \notin \mathcal{S}_a$, it follows:

$$\begin{aligned} \mathcal{Z}_a &= \left\{ \boldsymbol{z} \geq 0 \;\middle|\; \exists \boldsymbol{\nu}_a \geq 0 \text{ s.t. } \forall i \in N(a), z_i = \sum_{\substack{\boldsymbol{x}_a \in \mathcal{S}_a \\ \boldsymbol{x}_i = 1}} \nu_a(\boldsymbol{x}_a) = 1 - \sum_{\substack{\boldsymbol{x}_a \in \mathcal{S}_a \\ \boldsymbol{x}_i = 0}} \nu_a(\boldsymbol{x}_a) \right\} \\ &= \left\{ \boldsymbol{z} \geq 0 \;\middle|\; \exists \boldsymbol{\nu}_a \geq 0, \sum_{\boldsymbol{x}_a \in \mathcal{S}_a} \nu_a(\boldsymbol{x}_a) = 1 \text{ s.t. } \boldsymbol{z} = \sum_{\boldsymbol{x}_a \in \mathcal{S}_a} \nu_a(\boldsymbol{x}_a) \boldsymbol{x}_a \right\} = \operatorname{conv} \mathcal{S}_a. \end{aligned} \tag{41}$$

■

Note also that $\|\boldsymbol{\nu}_i^a - \eta_t^{-1} \boldsymbol{\omega}_i^a\|^2 = (\nu_i^a(1) - \eta_t^{-1} \omega_i^a(1))^2 + (1 - \nu_i^a(1) - \eta_t^{-1} \omega_i^a(0))^2$ equals a constant plus $2(\nu_i^a(1) - \eta_t^{-1}(\omega_i^a(1) + 1 - \omega_i^a(0))/2)^2$. Hence, (12) reduces to computing the following projection:

$$\operatorname{proj}_a(\boldsymbol{z}_0) \triangleq \operatorname*{argmin}_{\boldsymbol{z} \in \operatorname{conv} \mathcal{S}_a} \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{z}_0\|^2, \qquad \text{where } \boldsymbol{z}_{0i} \triangleq (\omega_i^a(1) + 1 - \omega_i^a(0))/2, \forall i. \tag{42}$$

Another important fact has to do with *negated inputs*. Let factor a' be constructed from a by negating one of the inputs (without loss of generality, the first one, x_1)—*i.e.*, $\mathcal{S}_{a'} = \{\boldsymbol{x}_{a'} \mid (1 - x_1, x_2, \ldots, x_m) \in \mathcal{S}_a\}$. Then, if we have a procedure for evaluating the operator proj_a, we can use it for evaluating $\operatorname{proj}_{a'}$ through the change of variable $z_1' \triangleq 1 - z_1$, which turns the objective function into $(1 - z_1' - z_{01})^2 = (z_1' - (1 - z_{01}))^2$. Naturally, the same idea holds when there is more than one negated input. The overall procedure computes $\boldsymbol{z} = \operatorname{proj}_{a'}(\boldsymbol{z}_0)$:

1. For each input i, set $z_{0i}' = z_{0i}$ if it is not negated and $z_{0i}' = 1 - z_{0i}$ otherwise.

2. Obtain $\boldsymbol{z}'$ as the solution of $\operatorname{proj}_a(\boldsymbol{z}_0')$.

3. For each input i, set $z_i = z_i'$ if it is not negated and $z_i = 1 - z_i'$ otherwise.

Below, we will also use the following

Algorithm 3 Projection onto simplex (Duchi et al., 2008)

Input: $\boldsymbol{z}_0$
Sort $\mathbf{z}_0$ into $\mathbf{y}_0$: $y_1 \geq \ldots \geq y_m$
Find $\rho = \max\left\{ j \in [m] \mid y_{0j} - \frac{1}{j}\left(\sum_{r=1}^{j} y_{0r} - 1\right) > 0 \right\}$
Define $\tau = \frac{1}{\rho}\left(\sum_{r=1}^{\rho} y_{0r} - 1\right)$
Output: $\boldsymbol{z}$ s.t. $z_i = \max\{z_{0i} - \tau, 0\}$.

Lemma 4 *Consider a problem of the form*

$$P: \qquad \min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \quad \text{s.t.} \quad g(\boldsymbol{x}) \leq 0, \tag{43}$$

where $\mathcal{X}$ is nonempty convex subset of $\mathbb{R}^d$ and $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{X} \to \mathbb{R}$ are convex functions. Suppose that the problem (43) is feasible and bounded below, and let $\mathcal{A}$ be the set of solutions of the relaxed problem $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$, *i.e.* $\mathcal{A} = \operatorname{Argmin}_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$. *Then:*

1. *if for some $\tilde{\boldsymbol{x}} \in \mathcal{A}$ we have $g(\tilde{\boldsymbol{x}}) \leq 0$, then $\tilde{\boldsymbol{x}}$ is also a solution of the original problem P;*
2. *otherwise (if for all $\tilde{\boldsymbol{x}} \in \mathcal{A}$ we have $g(\tilde{\boldsymbol{x}}) > 0$), the inequality constraint is necessarily active in P, i.e., problem P is equivalent to $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ s.t. $g(\boldsymbol{x}) = 0$.*

Proof: Let f^* be the optimal value of P. The first statement is obvious: since $\tilde{\boldsymbol{x}}$ is a solution of a relaxed problem we have $f(\tilde{\boldsymbol{x}}) \leq f^*$; hence if $\tilde{\boldsymbol{x}}$ is feasible this becomes an equality. For the second statement, assume that $\exists \boldsymbol{x} \in \mathcal{X}$ s.t. $g(\boldsymbol{x}) < 0$ (otherwise, the statement holds trivially). The nonlinear Farkas' lemma (Bertsekas et al., 2003, Prop. 3.5.4, p. 204) implies that there exists some $\lambda^* \geq 0$ s.t. $f(\boldsymbol{x}) - f^* + \lambda^* g(\boldsymbol{x}) \geq 0$ holds for all $\boldsymbol{x} \in \mathcal{X}$. In particular, this also holds for an optimal $\boldsymbol{x}^*$ (i.e., such that $f^* = f(\boldsymbol{x}^*)$), which implies that $\lambda^* g(\boldsymbol{x}^*) \geq 0$. However, since $\lambda^* \geq 0$ and $g(\boldsymbol{x}^*) \leq 0$ (since $\boldsymbol{x}^*$ has to be feasible), we also have $\lambda^* g(\boldsymbol{x}^*) \leq 0$, *i.e.*, $\lambda^* g(\boldsymbol{x}^*) = 0$. Now suppose that $\lambda^* = 0$. Then we have $f(\boldsymbol{x}) - f^* \geq 0$, $\forall \boldsymbol{x} \in \mathcal{X}$, which implies that $\boldsymbol{x}^* \in \mathcal{A}$ and contradicts the assumption that $g(\tilde{\boldsymbol{x}}) > 0, \forall \tilde{\boldsymbol{x}} \in \mathcal{A}$. Hence we must have $g(\boldsymbol{x}^*) = 0$. ■

B.1. Derivation of QUAD for the XOR Factor

The XOR factor is defined as:

$$\phi_{\text{XOR}}(x_1, \ldots, x_m) = \begin{cases} 0 & \text{if } \sum_{i=1}^{m} x_i = 1 \\ -\infty & \text{otherwise,} \end{cases} \tag{44}$$

where each $x_i \in \{0, 1\}$. When $m = 2$, $\exp(\phi_{\text{XOR}}(x_1, x_2)) = x_1 \oplus x_2$ (the Boolean XOR function), hence the name. When $m > 2$, it is a "one-hot" generalization of $\oplus$.[3] For this case, we have that $\operatorname{conv} \mathcal{S}_{\text{XOR}} = \{\boldsymbol{z} \geq 0 \mid \mathbf{1}^\top \boldsymbol{z} = 1\}$ is the probability simplex. Hence the quadratic problem (12) reduces to that of projecting onto the simplex, which can be done efficiently by Alg. 3. This algorithm requires a sort operation and its cost is $O(m \log m)$.[4]

B.2. Derivation of QUAD for the OR Factor

The OR factor is defined as:

$$\phi_{\text{OR}}(x_1, \ldots, x_m) = \begin{cases} 0 & \text{if } \sum_{i=1}^{m} x_i \geq 1 \\ -\infty & \text{otherwise,} \end{cases} \tag{45}$$

where each $x_i \in \{0, 1\}$. This factor indicates whether any of its input variables is 1, hence the name. For this case, we have that $\operatorname{conv} \mathcal{S}_{\text{OR}} = \{\boldsymbol{z} \in [0, 1]^m \mid \mathbf{1}^\top \boldsymbol{z} \geq 1\}$ is a "faulty" unit hypercube where one of the vertices (the origin) is missing. From Lemma 4, we have that the following procedure solves (42) for $\mathcal{S}_{\text{OR}}$:

[3] There is another generalization of $\oplus$ which returns the parity of $x_1, \ldots, x_m$; ours should not be confused with that one.

[4] In the high-dimensional case, a red-black tree can be used to reduce this cost to $O(m)$ (Duchi et al., 2008). In later iterations of DD-ADMM, great speed ups can be achieved in practice since this procedure is repeatedly invoked with only small changes on the coefficients.

1. Set $\tilde{\boldsymbol{z}}$ as the projection of $\boldsymbol{z}_0$ onto the unit cube. This can be done by clipping each coordinate to the unit interval $\mathbb{U} = [0, 1]$, i.e., by setting $\tilde{z}_i = [z_{0i}]_{\mathbb{U}} = \min\{1, \max\{0, z_{0i}\}\}$. If $\mathbf{1}^\top \tilde{\boldsymbol{z}} \geq 1$, then return $\tilde{\boldsymbol{z}}$. Else go to step 2.

2. Return the projection of $\boldsymbol{z}_0$ onto the simplex (use Alg. 3).

The validity of the second step stems from the fact that, if the relaxed problem in the first step does not return a feasible point, then the constraint $\mathbf{1}^\top \boldsymbol{z} \geq 1$ has to be active, i.e., we must have $\mathbf{1}^\top \boldsymbol{z} = 1$. This, in turn, implies that $\boldsymbol{z} \leq 1$ hence the problem becomes equivalent to the XOR case.

B.3. Derivation of QUAD for the OR-WITH-OUTPUT Factor

The OR-WITH-OUTPUT factor is defined as:

$$\phi_{\text{OR-OUT}}(x_1, \ldots, x_m) = \begin{cases} 0 & \text{if } x_m = \bigvee_{i=1}^{m-1} x_i \\ -\infty & \text{otherwise,} \end{cases} \tag{46}$$

where each $x_i \in \{0, 1\}$. In other words, this factor indicates (via the "output" variable x_m) if any of variables x_1 to x_{m-1} (the "input" variables) is active. Solving the quadratic problem for this factor is slightly more complicated than in the previous two cases; however, we next see that it can also be addressed in $O(m \log m)$ with a sort operation. For this case, we have that

$$\operatorname{conv} \mathcal{S}_{\text{OR-OUT}} = \left\{ \boldsymbol{z} \in [0,1]^m \;\middle|\; z_m \leq \sum_{i=1}^{m-1} z_i \text{ and } z_m \geq z_i, i = 1, \ldots, m-1 \right\}.$$

It is instructive to write this polytope as the intersection of the three sets $[0, 1]^m$, $\mathcal{A}_1 \triangleq \{\boldsymbol{z} \mid z_m \geq z_i, i = 1, \ldots, m-1\}$, and $\mathcal{A}_2 \triangleq \{\boldsymbol{z} \in [0, 1]^m \mid z_m \leq \mathbf{1}^\top \boldsymbol{z}_1^{m-1}\}$. We further define $\mathcal{A}_0 \triangleq [0, 1]^m \cap \mathcal{A}_1$. From Lemma 4, we have that the following procedure is correct:

1. Set $\tilde{\boldsymbol{z}}$ as the projection of $\boldsymbol{z}_0$ onto the unit cube. If $\tilde{\boldsymbol{z}} \in \mathcal{A}_1 \cap \mathcal{A}_2$, then we are done: just return $\tilde{\boldsymbol{z}}$. Else, if $\tilde{\boldsymbol{z}} \in \mathcal{A}_1$ but $\tilde{\boldsymbol{z}} \notin \mathcal{A}_2$, go to step 3. Otherwise, go to step 2.

2. Set $\tilde{\boldsymbol{z}}$ as the projection of $\boldsymbol{z}_0$ onto $\mathcal{A}_0$ (we will describe how to do this later). If $\tilde{\boldsymbol{z}} \in \mathcal{A}_2$, return $\tilde{\boldsymbol{z}}$. Otherwise, go to step 3.

3. Return the projection of $\boldsymbol{z}_0$ onto the set $\{\boldsymbol{z} \in [0, 1]^m \mid z_m = \mathbf{1}^\top z_1^{m-1}\}$. This corresponds to the QUAD problem of a XOR factor with the mth input negated (we call such factor XOR-WITH-OUTPUT because it is analogous to OR-WITH-FACTOR but with the role of OR replaced by that of XOR). As explained above, it can be solved by projecting onto the simplex (use Alg. 3).

Note that the first step above can be ommited; however, it avoids performing step 2 (which requires a sort) unless it is really necessary. To completely specify the algorithm, we only need to explain how to compute the projection onto $\mathcal{A}_0$ (step 2). Recall that $\mathcal{A}_0 = [0, 1]^m \cap \mathcal{A}_1$. Fortunatelly, it turns out that the following sequential projection is correct:

Procedure 5 *To project onto $\mathcal{A}_0 = [0, 1]^m \cap \mathcal{A}_1$:*

2a. Set $\tilde{\tilde{\boldsymbol{z}}}$ as the projection of $\boldsymbol{z}_0$ onto $\mathcal{A}_1$. Alg. 4 shows how to do this.

2b. Set $\tilde{\boldsymbol{z}}$ as the projection of $\tilde{\tilde{\boldsymbol{z}}}$ onto the unit cube (with the usual clipping procedure).

The proof that the composition of these two projections yields the desired projection onto $\mathcal{A}_0$ is a bit involved, so we defer it to Prop. 6.[5] We only need to describe how to project onto $\mathcal{A}_1$ (step 2a), which is written as the following problem:

$$\min_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{z}_0\|^2 \quad \text{s.t.} \quad z_m \geq z_i, \; i = 1, \ldots, m-1. \tag{47}$$

[5] Note that in general, the composition of individual projections is not equivalent to projecting onto the intersection. In particular, commuting steps 2a and 2b would make our procedure incorrect.

Algorithm 4 Projection onto $\mathcal{A}_1$

Input: $\boldsymbol{z}_0$
Sort $z_{01}, \ldots, z_{0(m-1)}$ into $y_1 \geq \ldots \geq y_{m-1}$
Find $\rho = \min\left\{j \in [m] \mid \frac{1}{j}\left(z_{0m} + \sum_{r=1}^{j-1} y_r\right) > y_j\right\}$
Define $\tau = \frac{1}{\rho}\left(z_{0m} + \sum_{r=1}^{\rho-1} y_r\right)$
Output: $\boldsymbol{z}$ s.t. $z_m = \tau$ and $z_i = \min\{z_{0i}, \tau\}$, $i = 1, \ldots, m-1$.

Algorithm 5 Dijkstra's algorithm for projecting onto $\bigcap_{j=1}^{J} \mathcal{C}_j$

Input: Point $\boldsymbol{x}_0 \in \mathbb{R}^d$, convex sets $\mathcal{C}_1, \ldots, \mathcal{C}_J$
Initialize $\boldsymbol{x}^{(0)} = \boldsymbol{x}_0$, $\boldsymbol{u}_j^{(0)} = \mathbf{0}$ for all $j = 1, \ldots, J$
for $t = 1, 2, \ldots$ **do**
  for $j = 1$ **to** J **do**
    Set $s = j + (t-1)J$
    Set $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}^{(s-1)} - \boldsymbol{u}_j^{(t-1)}$
    Set $\boldsymbol{x}^{(s)} = \mathrm{proj}_{\mathcal{C}_j}(\tilde{\boldsymbol{x}}_0)$, and $\boldsymbol{u}_j^{(t)} = \boldsymbol{x}^{(s)} - \tilde{\boldsymbol{x}}_0$
  end for
end for
Output: $\boldsymbol{x}$

It can be successively rewritten as:

$$\begin{aligned}
& \min_{z_m} \frac{1}{2}(z_m - z_{0m})^2 + \sum_{i=1}^{m-1} \min_{z_i \leq z_m} \frac{1}{2}(z_i - z_{0i})^2 \\
= \;& \min_{z_m} \frac{1}{2}(z_m - z_{0m})^2 + \sum_{i=1}^{m-1} \frac{1}{2}(\min\{z_m, z_{0i}\} - z_{0i})^2 \\
= \;& \min_{z_m} \frac{1}{2}(z_m - z_{0m})^2 + \frac{1}{2} \sum_{i \in \mathcal{I}(z_m)} (z_m - z_{0i})^2. \qquad (48)
\end{aligned}$$

where $\mathcal{I}(z_m) \triangleq \{i : z_{0i} \geq z_m\}$. Assuming that the set $\mathcal{I}(z_m)$ is given, the previous is a sum-of-squares problem whose solution is $z_m^* = \frac{z_{0m} + \sum_{i \in \mathcal{I}(z_m)} z_{0i}}{1 + |\mathcal{I}(z_m)|}$. The set $\mathcal{I}(z_m)$ can be determined by inspection after sorting $z_{01}, \ldots, z_{0(m-1)}$. The procedure is shown in Alg. 4.

Proposition 6 *Procedure 5 is correct.*

Proof: The proof is divided into the following parts:

1. We show that Procedure 5 corresponds to the first iteration of Dijkstra's projection algorithm (Boyle & Dykstra, 1986) applied to sets $\mathcal{A}_1$ and $[0, 1]^m$;
2. We show that Dijkstra's converges in one iteration if a specific condition is met;
3. We show that with the two sets above that condition is met.

The first part is rather trivial. Dijkstra's algorithm is shown as Alg. 5; when $J = 2$, $\mathcal{C}_1 = \mathcal{A}_1$ and $\mathcal{C}_2 = [0, 1]^m$, and noting that $\boldsymbol{u}_1^{(1)} = \boldsymbol{u}_2^{(1)} = 0$, we have that the first iteration becomes Procedure 5.

We turn to the second part, to show that, when $J = 2$, the fact that $\boldsymbol{x}^{(3)} = \boldsymbol{x}^{(2)}$ implies that $\boldsymbol{x}^{(s)} = \boldsymbol{x}^{(2)}$, $\forall s > 3$. In words, if at the second iteration t of Dijkstra's, the value of $\boldsymbol{x}$ does not change after computing the first projection, then it will never change, so the algorithm has converged and $\boldsymbol{x}$ is the desired projection. To see that, consider the moment in Alg. 5 when $t = 2$ and $j = 1$. After the projection, we update $\boldsymbol{u}_1^{(2)} = \boldsymbol{x}^{(3)} - (\boldsymbol{x}^{(2)} - \boldsymbol{u}_1^{(1)})$, which when $\boldsymbol{x}^{(3)} = \boldsymbol{x}^{(2)}$ equals $\boldsymbol{u}_1^{(1)}$, *i.e.*, $\boldsymbol{u}_1$ keeps unchanged. Then, when $t = 2$ and $j = 2$, one first computes $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}^{(3)} - \boldsymbol{u}_2^{(1)} = \boldsymbol{x}^{(3)} - (\boldsymbol{x}^{(2)} - \boldsymbol{x}_0) = \boldsymbol{x}_0$, *i.e.*, the projection is the same as the one already computed at $t = 1$, $j = 2$. Hence the result is the same, *i.e.*, $\boldsymbol{x}^{(4)} = \boldsymbol{x}^{(2)}$, and similarly $\boldsymbol{u}_2^{(2)} = \boldsymbol{u}_2^{(1)}$. Since neither $\boldsymbol{x}$, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ changed in the second iteration, and subsequent iterations only depend on these values, we have that $\boldsymbol{x}$ will never change afterwards.

Finally, we are going to see that, regardless of the choice of $\boldsymbol{z}_0$ in Procedure 5 ($\boldsymbol{x}_0$ in Alg. 5) we will always have $\boldsymbol{x}^{(3)} = \boldsymbol{x}^{(2)}$. Looking at Alg. 4, we see that after $t = 1$:

$$x_i^{(1)} = \begin{cases} \tau, & \text{if } i = m \text{ or } x_{0i} \geq \tau \\ x_{0i}, & \text{otherwise,} \end{cases} \qquad u_{1i}^{(1)} = \begin{cases} \tau - x_{0i}, & \text{if } i = m \text{ or } x_{0i} \geq \tau \\ 0, & \text{otherwise,} \end{cases}$$
$$x_i^{(2)} = [x_i^{(1)}]_{\mathbb{U}} = \begin{cases} [\tau]_{\mathbb{U}}, & \text{if } i = m \text{ or } x_{0i} \geq \tau \\ [x_{0i}]_{\mathbb{U}}\,, & \text{otherwise.} \end{cases} \tag{49}$$

Hence in the beginning of the second iteration ($t = 2$, $j = 1$), we have

$$\tilde{x}_{0i} = x_i^{(2)} - u_{1i}^{(1)} = \begin{cases} [\tau]_{\mathbb{U}} - \tau + x_{0i}, & \text{if } i = m \text{ or } x_{0i} \geq \tau \\ [x_{0i}]_{\mathbb{U}}\,, & \text{otherwise.} \end{cases} \tag{50}$$

Now two things should be noted about Alg. 4:

- If a constant is added to all entries in $\boldsymbol{z}_0$, the set $\mathcal{I}(z_m)$ remains the same, and τ and $\boldsymbol{z}$ are affected by the same constant;
- Let $\boldsymbol{z}_0'$ be such that $z_{0i}' = z_{0i}$ if $i = m$ or $z_{0i} \geq \tau$, and $z_{0i}' \leq \tau$ otherwise. Let $\boldsymbol{z}'$ be the projected point when such $\boldsymbol{z}_0'$ is given as input. Then $\mathcal{I}(z_m') = \mathcal{I}(z_m)$, $\tau' = \tau$, $z_i' = z_i$ if $i = m$ or $z_{0i} \geq \tau$, and $z_i' = z_{0i}'$ otherwise.

The two facts above allow to relate the projection of $\tilde{\boldsymbol{x}}_0$ (in the second iteration) with that of $\boldsymbol{x}_0$ (in the first iteration). Using $[\tau]_{\mathbb{U}} - \tau$ as the constant, and noting that, for $i \neq m$ and $x_{0i} < \tau$, we have $[x_{0i}]_{\mathbb{U}} - [\tau]_{\mathbb{U}} + \tau \geq \tau$ if $x_{0i} < \tau$, the two facts imply that:

$$x_i^{(3)} = \begin{cases} x_i^{(1)} + [\tau]_{\mathbb{U}} - \tau = [\tau]_{\mathbb{U}}, & \text{if } i = m \text{ or } x_{0i} \geq \tau \\ [x_{0i}]_{\mathbb{U}}\,, & \text{otherwise;} \end{cases} \tag{51}$$

hence $\boldsymbol{x}^{(3)} = \boldsymbol{x}^{(2)}$, which concludes the proof. ∎