Soft Constraints in Integer Linear Programs

Vivek Srikumar

February 5, 2013

1 Introduction

The goal of this note is to show how inference with *soft constraints* can be solved using an off-the-shelf ILP solver. We will introduce soft constraints and then show how they can be introduced into the inference problem using the same representation as hard constraints. The note ends with examples that works out the details of incorporating disjunctive and equality soft constraints.

2 Setting

We consider the problem of predicting structures. That is, given an input $\mathbf{x}$, a weight vector $\mathbf{w}$, and a feature function Φ, we predict a structure $\mathbf{y}$ as

$$\arg\max_{\mathbf{y}\in\mathcal{Y}} \mathbf{w}^T\Phi\left(\mathbf{x},\mathbf{y}\right) \tag{1}$$

Here $\mathcal{Y}$ is the feasible region for the structures. We use the standard approach to break the structure into a collection of *parts* and extract features from them. Let each $y_i \in \mathbf{y}$ correspond to a part in the structure. That is the *inference variable* y_i is an indicator for the existence of the i^{th} part in the structure. Let the feature function Φ decompose into a sum of features for each part. That is, we have

$$\Phi\left(\mathbf{x},\mathbf{y}\right) = \sum_i y_i \Phi_i\left(\mathbf{x}\right). \tag{2}$$

Here, each $\Phi_i\left(\mathbf{x}\right)$ is a function that generates features from the i^{th} part of $\mathbf{x}$.

Now, we can rewrite the inference problem as the following integer linear program:

$$\arg\max_{\mathbf{y}} \sum_i y_i \mathbf{w}^T \Phi_i\left(\mathbf{x}\right) \tag{3}$$

$$\text{s.t.}\, \mathbf{y}\in\mathcal{Y} \tag{4}$$

The feasible region $\mathcal{Y}$ can be defined in terms of linear inequalities, possibly with the introduction of additional inference variables that do not participate in the objective. They can be expressed in a logical representation and converted into linear inequalities. This is fairly well understood and the reader can refer to [Riz12], [Yih04] or [PRY08] for a primer on converting from a logical representation to inequalities.

3 Soft constraints

Each linear inequality that defines the feasible region prohibits certain assignments to the structure. We refer to such constraints as *hard constraints*. In contrast, a *soft* constraint merely imposes a penalty on certain assignments rather than prohibiting them.

We write soft constraints as logical formulas that involve the input $\mathbf{x}$ and the output $\mathbf{y}$. Structures that satisfy the condition are penalized by a score associated with that constraint. In this note, we will not deal with how these penalties are learned and assume that we wish to perform inference with such penalties. For example, a soft constraint in a part-of-speech tagging system could penalize POS tag assignments that do not have a verb in them. While most sentences do have verbs, it might be possible to think of sentences that do not have one. Writing this as a hard constraint could lead to an incorrect prediction.

Instead, we can have a constraint $C(\mathbf{x}, \mathbf{y})$ that expresses the logical formula and assign a penalty ρ_C to it. During inference, every prediction that violates the constraint will be forced to pay the penalty ρ_C. Yet, if the model strongly pushes the prediction towards a violating structure, the inference can result in predictions that violate the constraint.

Given soft constraints $C_1, C_2, \cdots$ and associated penalties $\rho_1, \rho_2, \cdots$, we can write inference as follows:

$$\begin{aligned} \arg\max_{\mathbf{y}} \quad & \sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) - \sum_j \rho_j \neg C_j(\mathbf{x}, \mathbf{y}) \\ \text{s.t.} \quad & \mathbf{y} \in \mathcal{Y} \end{aligned} \tag{5}$$

Here, we use the notation $\neg C$ indicate that the constraint is violated. Note that hard constraints can be seen as a special case of soft constraints where the penalty is infinite.

Equation 5 is a special case of the standard Constrained Conditional Model representation, where $\neg C$ is generalized into an arbitrary function d_C (See [CRR12] for a discussion about Constrained Conditional Models.):

$$\arg\max_{\mathbf{y}} \sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) - \sum_j \rho_j d_j^C(\mathbf{x}, \mathbf{y}). \tag{6}$$

Here, the constraints $\mathbf{y} \in \mathcal{Y}$ included into the constraints with an infinite penalty.

In this note, we will use two definitions of the function d_j^C. Equation 5 considers the case when d_j^C is defined as

$$d_j^C(\mathbf{x}, \mathbf{y}) = \begin{cases} 1, & \text{Constraint } C_j(\mathbf{x}, \mathbf{y}) \text{ is violated,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

We will refer to this as 0-1 soft constraints.

[CRR12] assume that the soft constraints can be decomposed over partial assignments to the structure. We will refer to such soft constraints as *partial assignment* soft constraints. Using the notation $\mathbf{y}_{1:i}$ to denote a partial assignment to the first i inference variables, we write this as

$$d_j^C(\mathbf{x}, \mathbf{y}) = \sum_i \neg \widehat{C}_j(\mathbf{x}, \mathbf{y}_{1:i}) \tag{8}$$

Here $\neg \widehat{C}_j$ refers to the violation of a logical constraint $\widehat{C}_j$ that applies to partial assignments to the structure. The function d_j^C is the number of such violations.

In the sequel, we will first see how 0-1 soft constraints can be solved using an off-the-shelf ILP solver. Then we will reduce the partial assignment soft constraints to the first case, leading to further generalizations.

4 Incorporating soft constraints into the ILP solver

0-1 soft constraints To write the 0-1 soft constraints expressed in Equations (6) and (7) as an integer linear program, we will introduce a Boolean variable z_j for each constraint C_j. This variable is an indicator that the constraint is satisfied and the corresponding penalty should *not* be accrued. This helps us to rewrite the objective of inference as

$$\arg\max_{\mathbf{y}} \sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) - \sum_j \rho_j (1 - z_j) \tag{9}$$

$$= \arg\max_{\mathbf{y}} \sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) + \sum_j \rho_j z_j \tag{10}$$

Note that this version of the objective has only 0-1 integer variables y_i and z_i. The only thing that remains is to connect the z_i to the constraints C_j. This is achieved by adding the following logical constraint:

$$z_j \leftrightarrow C_j(\mathbf{x}, \mathbf{y}) \tag{11}$$

Thus the final inference problem can be written as follows:

$$\begin{aligned} \arg\max_{\mathbf{y}} \quad & \sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) + \sum_j \rho_j z_j \\ \text{s.t.} \quad & \mathbf{y} \in \mathcal{Y} \\ \forall C_j, \quad & z_j \leftrightarrow C_j(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{12}$$

The final constraint that connects the z variables with the C expressions can be converted into logical representation in the standard way.

Partial assignment soft constraints To reduce the partial assignment soft constraints to the earlier case, we use the definition of the constraints in Equation (8) and rewrite the objective of inference from Equation (6) as

$$\sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) - \sum_{j,i} \rho_j \neg \widehat{C}_j(\mathbf{x}, \mathbf{y}_{1:i}). \tag{13}$$

Observe that this objective function is similar to the objective for the 0-1 soft constraints. As in the earlier case, we can introduce new inference variables $z_{i,j}$ that have coefficients ρ_j in the objective. By adding hard constraints $z_{i,j} \leftrightarrow \widehat{C}_j(\mathbf{x}, \mathbf{y}_{1:i})$, we can solve inference with partial assignment soft constraints using an off the shelf solver.

Finite discrete soft constraints We will now consider a generalization of the previous case. Suppose the range of d_j^C is a finite set of real numbers, say $\{d_{j,1}, d_{j,2}, \cdots, d_{j,m}\}$. That is, suppose d_j^C is defined using a collection of mutually exclusive constraints $\widehat{C}_{j,k}(\mathbf{x}, \mathbf{y})$ for $k = 1, 2, \cdots m$ as follows:

$$d_j^C(\mathbf{x}, \mathbf{y}) = \begin{cases} d_{j,k}, & \text{if constraint } \widehat{C}_{j,k}(\mathbf{x}, \mathbf{y}) \text{ is violated for any } i, \text{ or} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

This can be rewritten as

$$d_j^C(\mathbf{x}, \mathbf{y}) = \sum_k d_{j,k}(\neg \widehat{C}_{j,k}(\mathbf{x}, \mathbf{y})) \tag{15}$$

As before, the CCM objective function from Equation (6) can be written using this definition as

$$\sum_i y_i \mathbf{w}^T \Phi_i(\mathbf{x}) - \sum_{j,k} \rho_j d_{j,k} \neg \widehat{C}_{j,k}(\mathbf{x}, \mathbf{y}). \tag{16}$$

To convert this into an integer linear program, we introduce new inference variables $z_{j,k}$ with coefficients $\rho_j d_{j,k}$. Each $z_{j,k}$ corresponds to a new constraint $z_{j,k} \leftrightarrow \widehat{C}_{j,k}(\mathbf{x}, \mathbf{y})$. This completes the formulation.

5 Worked examples

In this section, we will present two worked examples that show how the conversion from $z \leftrightarrow C$ constraints to linear inequalities can be done in a systematic way. The first example deals with disjunctive soft constraints (that could express, for example, that at least one word in the sentence should be a verb). The second one deals with equality (which could express the preference that two variables should be equal.)

5.1 Disjunctive soft constraints

Suppose $C(\mathbf{x}, \mathbf{y}) = y_1 \vee y_2 \vee \cdots y_n$ for some n inference variables. Thus, we have the following constraint in our final ILP:

$$z \leftrightarrow y_1 \vee y_2 \vee \cdots y_n \tag{17}$$

We can convert the double implication into a collection of implications,

$$\begin{aligned} & z \rightarrow y_1 \vee y_2 \vee \cdots y_n \\ \forall 1 \leq i \leq n, \quad & y_i \rightarrow z. \end{aligned}$$

The first implication can be written as $\neg z \vee y_1 \vee y_2 \vee \cdots y_n$, which corresponds to the following inequality

$$\sum_{i=1}^{n} y_i \geq z$$

Similarly, each of the $y_i \rightarrow z$ constraints correspond to an inequality $z \geq y_i$. Thus we have the following set of inequalities that correspond to the soft constraint:

$$\begin{aligned} \sum_{i=1}^{n} y_i &\geq z, \\ \forall 1 \leq i \leq n, \qquad z &\geq y_i. \end{aligned}$$

5.2 Equality soft constraints

We now consider soft constraints that express the preference for two variables being equal. That is $C(\mathbf{x}, \mathbf{y}) \equiv y_1 \leftrightarrow y_2$. The corresponding constraint that we wish to convert is into linear inequalities is the following, which we will denote by the symbol f.

$$f(y_1, y_2, z) \equiv z \leftrightarrow (y_1 \leftrightarrow y_2) \tag{18}$$

This can be converted into a set of linear inequalities by methodically converting the right hand side into a conjunctive normal form (CNF). Here, instead, we look at a different way to convert the constraint into inequalities. We begin by observing that the function f defined above is the XOR function between three arguments and can be written as follows:

$$\begin{aligned} f(y_1, y_2, z) = & \qquad (y_1 \vee y_2 \vee z) \\ & \wedge \quad (\neg y_1 \vee \neg y_2 \vee z) \\ & \wedge \quad (\neg y_1 \vee y_2 \vee \neg z) \\ & \wedge \quad (y_1 \vee \neg y_w \vee \neg z) \end{aligned}$$

Each clause in this CNF can be converted into an inequality constraint for the ILP. Doing so, we get the following constraints:

$$\begin{aligned} y_1 + y_2 + z &\geq 1 \\ y_1 - y_2 - z &\geq -1 \\ -y_1 + y_2 - z &\geq -1 \\ -y_1 - y_2 + z &\geq -1 \end{aligned}$$

References

[CRR12] M. Chang, L. Ratinov, and D. Roth. Structured learning with constrained conditional models, June 2012.

[PRY08] V. Punyakanok, D. Roth, and W. Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2), 2008.

[Riz12] N. Rizzolo. *Learning Based Programming.* PhD thesis, University of Illinois, Urbana-Champaign, 2012.

[Yih04] W. Yih. Global inference using integer linear programming. 2004.