Temporal Ordering Notes

Spring 2014

1 Structured SVM

We have a generic structured SVM implemented according to the following objective function which minimizes the hinge loss over a set of N structured inputs and outputs $\{(\mathbf{x}_i, \mathbf{y}_i) | 1 \leq i \leq N\}$:

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{b}} \lambda_2 \|\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 \\ &+ \sum_{i=1}^{N} \left(- \mathbf{w}^\top \mathbf{g}(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{b}^\top \mathbf{l}(\mathbf{y}_i) + \max_{\mathbf{y} \in \mathcal{Y}_{x_i}} \left(\mathbf{w}^\top \mathbf{g}(\mathbf{x}_i, \mathbf{y}) + \mathbf{b}^\top \mathbf{l}(\mathbf{y}) + c(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}) \right) \right) \end{aligned} \tag{1}$$

The components of $\mathbf{w}$ are the regularized weights on features $\mathbf{g}$, the components of $\mathbf{b}$ are biases on label counts $\mathbf{l}$ and $c(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y})$ is the cost of predicting $\mathbf{y}$ for example i. The minimization of objective 1 results in feature weights $\mathbf{w}$ and biases $\mathbf{b}$ which make a prediction $\mathbf{y}$ for input $\mathbf{x}$ according to:

$$\mathbf{y} = \operatorname*{argmax}_{\mathbf{y}' \in \mathcal{Y}_x} \left(\mathbf{w}^\top \mathbf{g}(\mathbf{x}, \mathbf{y}') + \mathbf{b}^\top \mathbf{l}(\mathbf{y}') \right) \tag{2}$$

The model assumes that each structured input $\mathbf{x}_i$ factors into components $(x_{i1}, \ldots, x_{ij}, \ldots, x_{ik(\mathbf{x}_i)})$, and each structured output $\mathbf{y}_i$ factors into corresponding labels $(y_{i1}, \ldots, y_{ij}, \ldots, y_{ik(\mathbf{x}_i)})$. Each label y_{ij} comes from a set of factor labels $\mathcal{L}$ (and so $\mathcal{Y}_x = \mathcal{L}^{k(x_i)}$). Furthermore, the model is constructed so that $\mathbf{g}$ and $\mathbf{w}$ have components g_{ml} and w_{ml} where m indexes a feature type and $l \in \mathcal{L}$ is a factor label, with g_{ml} expressible as:

$$g_{ml}(\mathbf{x}_i, \mathbf{y}_i) = \sum_{j=1}^{k(\mathbf{x}_i)} \mathbf{1}(y_{ij} = l) f_m(x_{ij}) \tag{3}$$

Where each f_m is a percept that only depends on a single factor of the input structure (Nathan's terminology).

Similarly, the model assumes that the cost function $\mathbf{c}$ is factorable as:

$$c(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_h) = \sum_{j=1}^{k(\mathbf{x}_i)} \mathbf{1}(y_{ij} \neq y_{hj}) \tag{4}$$

When computing the loss for each example, the model must search a potentially large $\mathcal{Y}_x$ to find the value of the following expression:

$$\max_{\mathbf{y} \in \mathcal{Y}_{x_i}} \left(\mathbf{w}^\top \mathbf{g}(\mathbf{x}_i, \mathbf{y}) + \mathbf{b}^\top \mathbf{l}(\mathbf{y}) + c(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}) \right) \tag{5}$$

To perform this search, the model relies on the fact that this expression is equal to:

$$\max_{\mathbf{y}_h \in \mathcal{Y}_{x_i}} \left(\sum_{j=1}^{k(x_i)} \sum_{l \in \mathcal{L}} \left(\mathbf{1}(l = y_{hj}) \Big(\mathbf{1}(y_{ij} \neq y_{hj}) + \sum_m w_{ml} f_m(x_{ij}) \Big) \right) \right) \tag{6}$$

In order to compute the value of expression 6, the model computes the following for each i, j, and l:

$$u_{ijl} = \mathbf{1}(y_{ij} \neq l) + \sum_m w_{ml} f_m(x_{ij}) \tag{7}$$

Notice that these values are used in expression 6. They are passed to a domain-specific optimization function that computes $\mathbf{y}_h$ in expression 6 through:

$$\mathbf{e}_i = \underset{\mathbf{e}'_i \in \{0,1\}^{k(x_i)}}{\operatorname{argmax}} \mathbf{e}_i'^{\top} \mathbf{u} \tag{8}$$

Where each component e_{ijl} of $\mathbf{e}_i$ indicates whether $y_{hj} = l$, and these components are constrained so that the resulting $\mathbf{y}_h \in \mathcal{Y}_{x_i}$. The resulting $\mathbf{y}_h$ can be used to determine the value of expression 6.

The prediction for an example according to to equation 2 is computed similarly, just without the cost term in u_{ijl}

The implementation allows the model to remain abstract while referring to domain specific features and domain specific definitions for $\mathcal{Y}_x$.

1.1 Temporal Relation Classification Structured Model

The temporal relation classification task can use the model defined above with an implementation of equation 8 which has an ILP to enforce several constraints. The temporal relations can be partitioned into separate graph structures in several ways for this task, but we currently have them partitioned by sentence and sentence pairs. So, we have a set of within-sentence temporal relation graphs $\mathcal{X}_s$ and a set of between-sentence temporal relation graphs $\mathcal{X}_b$. Each $\mathbf{x}_s \in \mathcal{X}_s$ contains relations between the document creation time and all events and all times in a single sentences. Each $\mathbf{x}_b \in \mathcal{X}_b$ contains relations between all events and all times in two consecutive sentences. $\mathcal{X}_s$ is passed to a structured within-sentence model $\mathcal{M}_s$, and $\mathcal{X}_b$ is passed to a structured between-sentence model $\mathcal{M}_b$. The temporal relation classifications from $\mathcal{M}_s$ are used to constrain the output space of $\mathcal{M}_b$.

For this task, each e_{ijl} from equation 8 indicates whether event-event or event-time pair j has temporal relation l in graph i. There are several constraints on the value of each e_{ijl} enforced through an ILP that is run for each graph. To make the constraints easier to understand, we'll rewrite each e_{ijl} as t_{jkl} to indicate that there is a temporal relation of type l between vertices j and k (implicitly in the same example graph i). We have the following constraints:

1. **Label On-Off**: $\forall j \neq k, l : t_{jkl} \in \{0, 1\}$
2. **Single Label**: $\forall j \neq k : \sum_{l \in \mathcal{L}} t_{jkl} = 1$
3. **Converse**: $\forall j < k, l : t_{jkl} - t_{kjl'} \leq 0$ where l' is the converse of l
4. **Grounded Time-Time**: $t_{jkl} \geq 1$ if j and k index time expression vertices that are given relation l according to their grounded intervals
5. **Transitivity**: $\forall \{j, k, m\} : t_{jkl} + t_{kml'} - t_{jml''} \leq 1$ for Allen's interval algebra composition relations that say $l(j, k) \wedge l'(k, m) \rightarrow l''(j, m)$

6. **Disjunctive Transitivity**: $\forall\{j,k,m\} : t_{jkl} + t_{kml'} - \sum_{l''} t_{jml''} \leq 1$ for Allen's interval algebra composition relations that say $l(j,k) \wedge l'(k,m) \rightarrow \bigvee_{l''} l''(j,m)$

In total there are $n(n-1)|\mathcal{L}|$ variables and $O(n^3)$ constraints in the ILP for a graph with n vertices. The current implementation has variables t_{jkl} and t_{kjl} representing forward and backward links between two vertices. It might be possible to eliminate the backward link variables along with the **Converse** constraint for improved efficiency, but we haven't had time to think through that yet.

The Allen's interval relations enforced by the **Transitivity** and **Disjunctive Transitivity** constraints are given at `http://www.ics.uci.edu/~alspaugh/cls/shr/allen.html` and in the other document we passed around through email. Our implementation has the ability to easily turn off and on each of these constraints to see how they contribute to the overall performance.