Learning Explainable Templated Graphical Models - Supplementary Material

Varun Embar *[*1] **Sriram Srinivasan** *[1] **Lise Getoor**[1]

[1]Dept. of Computer Science and Engineering , University of California, Santa Cruz , USA

1 STRONG CONVEXITY OF PSL ENERGY FUNCTION

We first reiterate the definition of strong convexity.

Definition 1. Strong Convexity: A function $E : (\mathcal{Y}, \mathcal{X}) \rightarrow \mathbb{R}$ is κ-strongly convex in $\mathcal{Y}$ (w.r.t the 1-norm) if $\mathcal{Y}$ is a convex set and, for $\mathbf{X} \in \mathcal{X}$ and any $\mathbf{Y}, \mathbf{Y'} \in \mathcal{Y}$, $\tau \in [0, 1]$,

$$\tau(1-\tau)\frac{\kappa}{2}\|\mathbf{Y}-\mathbf{Y'}\| + E(\tau\mathbf{Y} + (1-\tau)\mathbf{Y'}, \mathbf{X}) \\ \leq \tau E(\mathbf{Y}, \mathbf{X}) + (1-\tau)E(\mathbf{Y'}, \mathbf{X})$$

The energy function E is a summation of squared hinges and hence E is convex. Further, the prior template described in our approach acts a regularizer of $\mathbf{Y}$ and is κ-strongly convex. Hence E is at least κ-strongly convex in $\mathcal{Y}$.

2 PROOF OF LEMMA 1

Lemma. For a graphical model G with a set of potentials $\mathbf{\Phi}$, let Q_i denote the number of potentials that involve $\mathbf{X_i}$, and let $Q_G \triangleq \max_i Q_i$. Let $\|\mathbf{w}\| < R$. Let $\mathbf{X}, \mathbf{X'} \in \mathcal{X}$ differ at a single coordinate i by atmost ϵ. Then, for $\dot{\mathbf{Y}} \triangleq \text{argmin}_{\mathbf{Y}} E(\mathbf{Y}, \mathbf{X})$ and $\dot{\mathbf{Y}}' \triangleq \text{argmin}_{\mathbf{Y}} E(\mathbf{Y}, \mathbf{X'})$,

$$\left\|E(\dot{\mathbf{Y}}', \mathbf{X}) - E(\dot{\mathbf{Y}}', \mathbf{X'})\right\| \leq \epsilon R\sqrt{Q_G}$$

Proof.

$$\left\|E(\dot{\mathbf{Y}}', \mathbf{X}) - E(\dot{\mathbf{Y}}', \mathbf{X'})\right\| \\ = \left\|\mathbf{w}^T\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X}) - \mathbf{w}^T\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X'})\right\| \\ \leq \|\mathbf{w}\| \left\|\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X}) - \mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X'})\right\| \text{ [Form Cauchy-Schwarz]} \\ \leq R\left\|\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X}) - \mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X'})\right\|$$

because, by definition, $\|\mathbf{w}\|$ is upper bounded by R. Note that $\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X})$ and $\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X'})$ only differ at any grounding involving $\mathbf{X_i}$. The number of such groundings is Q_i, which is upper-bounded by Q_G, so at most Q_G potentials will change.

Further, the squared hinge loss potential has the from $max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}^2$ where $\mathbf{C_x}, \mathbf{C_y}$ co-efficient vectors consisting of $1, -1, 0$.

$$\left\|\mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X}) - \mathbf{\Phi}(\dot{\mathbf{Y}}', \mathbf{X'})\right\| \\ = \left(\sum_{\phi\in\mathbf{\Phi}} \mathbb{1}\{\mathbf{C_{X_i}} \neq 0\}((\phi(\dot{\mathbf{Y}}', \mathbf{X}) - \phi(\dot{\mathbf{Y}}', \mathbf{X'})))^2\right)^{1/2} \\ = \left(\sum_{\phi\in\mathbf{\Phi}} \mathbb{1}\{\mathbf{C_{X_i}} \neq 0\}(max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\dot{\mathbf{Y}}' - c, 0\} \right. \\ \left. -max\{\mathbf{C_x}^T\mathbf{X'} + \mathbf{C_y}^T\dot{\mathbf{Y}}' - c, 0\})^2\right)^{1/2} \\ \leq \left(\sum_{\phi\in\mathbf{\Phi}} \mathbb{1}\{\mathbf{C_{X_i}} \neq 0\}(max\{\mathbf{C_x}^T(\mathbf{X} - \mathbf{X'}) \right. \\ \left. +\mathbf{C_y}^T(\dot{\mathbf{Y}}' - \dot{\mathbf{Y}}'), 0\})^2\right)^{1/2} \\ \leq (Q_i)^{1/2}\epsilon \leq (Q_G)^{1/2}\epsilon$$

3 PROOF OF LEMMA 2

Lemma. Let $E : (\mathcal{Y}, \mathcal{X}) \rightarrow \mathbb{R}$ be κ-strongly convex, and let $\dot{\mathbf{Y}} \triangleq \text{argmin}_{\mathbf{Y}} E(\mathbf{Y}, \mathbf{X})$ and $\dot{\mathbf{Y}}' \triangleq \text{argmin}_{\mathbf{Y}} E(\mathbf{Y}, \mathbf{X'})$, where $\mathbf{X}, \mathbf{X'} \in \mathcal{X}$ differ at a single RV $\mathbf{X_i}$ by atmost ϵ. Then,

$$\left\|\dot{\mathbf{Y}}' - \dot{\mathbf{Y}}\right\|^2 \leq \frac{2}{\kappa}(E(\dot{\mathbf{Y}}, \mathbf{X}) - E(\dot{\mathbf{Y}}', \mathbf{X'})) \quad (1)$$

*Equal contribution

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

Proof. Without loss of generality, assume that $E(\dot{\mathbf{Y}}, \mathbf{X}) \geq E(\dot{\mathbf{Y}}', \mathbf{X}')$.(If $E(\dot{\mathbf{Y}}, \mathbf{X}) \leq E(\dot{\mathbf{Y}}', \mathbf{X}')$ we can state this in terms of $\dot{\mathbf{Y}}'$). Let $\Delta\mathbf{Y} \triangleq \dot{\mathbf{Y}}' - \dot{\mathbf{Y}}$. By Definition 1, for any $\tau \in [0, 1]$,

$$\tau(1-\tau)\frac{\kappa}{2}\left\|\dot{\mathbf{Y}}' - \dot{\mathbf{Y}}\right\| + E(\tau\dot{\mathbf{Y}}' + (1-\tau)\dot{\mathbf{Y}}, \mathbf{X}) \leq \tau E(\dot{\mathbf{Y}}', \mathbf{X}) + (1-\tau)E(\dot{\mathbf{Y}}, \mathbf{X})$$

Since $\dot{\mathbf{Y}}$ is, by definition, the unique minimizer of $E(\mathbf{Y}, \mathbf{X})$, it follows that $E(\dot{\mathbf{Y}} + \tau\Delta\mathbf{Y}, \mathbf{X}) - E(\dot{\mathbf{Y}}, \mathbf{X}) \geq 0$, so the above inequality is preserved when this term is dropped. This, dividing both sides by $\tau\kappa/2$, we have that

$$(1-\tau)\left\|\Delta\mathbf{Y}\right\|^2 \leq \frac{2}{\kappa}(E(\dot{\mathbf{Y}}', \mathbf{X}) - E(\dot{\mathbf{Y}}, \mathbf{X}))$$
$$\left\|\Delta\mathbf{Y}\right\|^2 \leq \frac{2}{\kappa}(E(\dot{\mathbf{Y}}', \mathbf{X}) - E(\dot{\mathbf{Y}}, \mathbf{X}))$$

where the last inequality follows from the fact that $(1-\tau)$ is maximized at $\tau = 0$.

Since $E(\dot{\mathbf{Y}}, \mathbf{X}) \geq E(\dot{\mathbf{Y}}', \mathbf{X}')$, the following inequality holds

$$\left\|\dot{\mathbf{Y}}' - \dot{\mathbf{Y}}\right\|^2 \leq \frac{2}{\kappa}(E(\dot{\mathbf{Y}}', \mathbf{X}) - E(\dot{\mathbf{Y}}', \mathbf{X}'))$$

4 PROOF OF LEMMA 3

Lemma. Let the explaining function f be defined as $f(\mathbf{X}, \mathbf{Y}, \phi) = \left\|\frac{w\partial\phi(\mathbf{X},\mathbf{Y})}{\partial\mathbf{Y_i}}|_y\right\|$. Let $\mathbf{X}, \mathbf{X}' \in \mathcal{X}$ differ at a single random variable $\mathbf{X_i}$ by at most ϵ. Let $\|\mathbf{Y} - \mathbf{Y}'\| < B$ for any two $\mathbf{Y}, \mathbf{Y}' \in \mathcal{Y}$ and $\|\mathbf{w}\| < R$. Then:

$$|f(\mathbf{X}, \mathbf{Y}, \phi) - f_k(\mathbf{X}', \mathbf{Y}', \phi)| \leq 2R(\epsilon + B) \quad (2)$$

Proof. The hinge loss function ϕ has the from $max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}^2$ where $\mathbf{C_x}, \mathbf{C_y}$ coefficient vectors consisting of $1, 0, -1$.

The partial derivative w.r.t to $\mathbf{Y_i}$

$$= \left\|\frac{\partial\phi(\mathbf{X},\mathbf{Y})}{\partial\mathbf{Y_i}}|_y\right\|$$
$$= 2 * max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\} * \left\|\frac{\partial max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}}{\partial\mathbf{Y_i}}|_y\right\|$$
$$= 2 * max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}$$

The last step comes from the fact that $\left\|\frac{\partial max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}}{\partial\mathbf{Y_i}}|_y\right\|$ can be $\{-1, 0, 1\}$. In all cases,

$$2 * max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\} * \left\|\frac{\partial max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}}{\partial\mathbf{Y_i}}|_y\right\|$$
$$= 2 * max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\}$$

Now consider $|f(\mathbf{X}, \mathbf{Y}, \phi) - f(\mathbf{X}', \mathbf{Y}', \phi)|$

$$= w\left\|\frac{\partial\phi(\mathbf{X},\mathbf{Y})}{\partial\mathbf{Y_i}}|_y - \frac{\partial\phi_j(\mathbf{X}',\mathbf{Y}')}{\partial\mathbf{Y_i}}|_{y'}\right\|$$
$$= 2w\|max\{\mathbf{C_x}^T\mathbf{X} + \mathbf{C_y}^T\mathbf{Y} - c, 0\} - max\{\mathbf{C_x}^T\mathbf{X}' + \mathbf{C_y}^T\mathbf{Y}' - c, 0\}\|$$
$$\leq 2w\left\|max\{\mathbf{C_x}^T(\mathbf{X} - \mathbf{X}') + \mathbf{C_y}^T(\mathbf{Y} - \mathbf{Y}'), 0\}\right\|$$
$$\leq 2w\left\|max\{\epsilon + B, 0\}\right\|$$
$$\leq 2R(\epsilon + B)$$

5 DATASETS

Enity Resolution Dataset: The Cora Citation entity resolution dataset is based on the citation references between scientific papers. The task is to identify papers that are the same. This is represented by the target predicate $SAME\_BIB$. The dataset contains 10 predicates. For each of the non-target predicates, we included the inverse predicates where the arguments are reversed. For example, for the predicate $SAME\_AUTHOR(A, B)$ we include the predicate $\_SAME\_AUTHOR(B, A)$. In total there are 19 predicates.

The set of predicates are $SAME\_AUTHOR$, $SAME\_BIB$, $SAME\_VENUE$, $SAME\_TITLE$, $AUTHOR$, $VENUE$, $TITLE$, $HASWORD\_AUTHOR$, $HASWORD\_TITLE$, $HASWORD\_VENUE$. Since all predicates are explainable, we do not classify them as explainable and non-explainable predicates. The dataset is split into 5 folds. We use the same splits as **?**.

Rcommendation Dataset: Both YELP and LASTFM datasets contain 21 predicates or relations. We categorize the predicates as explainable and non-explainable predicates based on how easy it is for an end-user to understand the predicates. There were 15 explainable predicates and 6 non-explainable predicates. The list of predicates are as follows:

Explainable predicates: $USERS\_ARE\_FRIENDS$, $SIM\_CONTENT\_ITEMS\_JACCARD$, $SIM\_PEARSON\_ITEMS$, SIM_COSINE_ITEMS, SIM_ADJCOS_ITEMS, $SIM\_MF\_COSINE\_ITEMS$, $SIM\_MF\_EUCLIDEAN\_ITEMS$, SIM_COSINE_USERS, $SIM\_PEARSON\_USERS$, $SIM\_MF\_COSINE\_USERS$, $SIM\_MF\_EUCLIDEAN\_USERS$, AVG_ITEM_RATING, $RATING\_PRIOR$, AVG_USER_RATING

Non-explainable predicates: $RATING$, $RATED$, SGD_RATING, $BPMF\_RATING$, $ITEM\_PEARSON\_RATING$, $USER$, $ITEM$

The **YELP** dataset is split in five folds. Each fold contains a train and a test split. The train splits contains 79240 observed ratings and 7924 ratings that need to be predicted. The test split contains 99049 observed ratings and 19809 ratings that need to be predicted.

Similarly, the **LASTFM** dataset is split in five folds. Each fold contains a train and a test split. The train splits contains 74267 observed ratings and 18567 ratings that need to be predicted. The test split contains 92834 observed ratings and 18567 ratings that need to be predicted.

6 MODE DECLARATIONS FOR BOOST

Modes are used to restrict/guide the search space and are a powerful tool in getting relational algorithms such as BoostSRL to work. Below we give the mode declarations used by BOOST for the recommendation and entity resolution datasets.

Entity resolution dataset:

$mode : author(+paper, -auth).$
$mode : haswordauthor(+auth, -word).$
$mode : haswordtitle(+title, -word).$
$mode : haswordvenue(+venue, -word).$
$mode : title(+paper, -title).$
$mode : venue(+paper, -venue).$
$mode : author(-paper, +auth).$
$mode : haswordauthor(-auth, +word).$
$mode : haswordtitle(-title, +word).$
$mode : haswordvenue(-venue, +word).$
$mode : title(-paper, +title).$
$mode : venue(-paper, +venue).$
$mode : samebib(+paper, +paper).$
$mode : sametitle(+title, +title).$
$mode : samevenue(+venue, +venue).$
$mode : sameauthor(+auth, +auth).$
$mode : recursive\_samebib(+paper, paper).$
$mode : recursive\_sametitle(+title, title).$
$mode : recursive\_samevenue(+venue, venue).$
$mode : recursive\_sameauthor(+auth, auth).$
$mode : recursive\_samebib(`paper, +paper).$
$mode : recursive\_sametitle(title, +title).$
$mode : recursive\_samevenue(venue, +venue).$
$mode : recursive\_sameauthor(auth, +auth).$
$mode : samebib(+paper, -paper).$
$mode : sametitle(+title, -title).$
$mode : samevenue(+venue, -venue).$
$mode : sameauthor(+auth, -auth).$
$mode : samebib(-paper, +paper).$
$mode : sametitle(-title, +title).$
$mode : samevenue(-venue, +venue).$
$mode : sameauthor(-auth, +auth).$
$usePrologVariables : true.$
$okIfUnknown : recursive\_sametitle/2.$
$okIfUnknown : recursive\_samebib/2.$
$okIfUnknown : recursive\_samevenue/2.$
$okIfUnknown : recursive\_sameauthor/2.$

Recommendation datasets:

$mode : avg\_item\_rating(+item).$
$mode : avg\_user\_rating(+user).$
$mode : bpmf\_rating(+user, +item).$
$mode : item\_pearson\_rating(+user, +item).$
$mode : sgd\_rating(+user, +item).$
$mode : users\_are\_friends(+user, -user).$
$mode : users\_are\_friends(-user, +user).$
$mode : sim\_adjcos\_items(+item, -item).$
$mode : sim\_adjcos\_items(-item, +item).$
$mode : sim\_content\_items_jaccard(-item, +item).$
$mode : sim\_content\_items_jaccard(-item, +item).$
$mode : sim\_cosine\_items(-item, +item).$
$mode : sim\_cosine\_items(+item, -item).$
$mode : sim\_cosine\_users(-user, +user).$
$mode : sim\_cosine\_users(+user, -user).$
$mode : sim\_mf\_cosine\_items(-item, +item).$
$mode : sim\_mf\_cosine\_items(+item, -item).$
$mode : sim\_mf\_cosine\_users(-user, +user).$
$mode : sim\_mf\_cosine\_users(+user, -user).$
$mode : sim\_mf\_euclidean\_items(-item, +item).$
$mode : sim\_mf\_euclidean\_items(+item, -item).$
$mode : sim\_mf\_euclidean\_users(-user, +user).$
$mode : sim\_mf\_euclidean\_users(+user, -user).$
$mode : sim\_pearson\_users(-user, +user).$
$mode : sim\_pearson\_users(+user, -user).$
$mode : sim\_pearson\_items(-item, +item).$
$mode : sim\_pearson\_items(+item, -item).$
$mode : rating(+user, +item).$

$bridger : friends/2.$
$bridger : sim\_adjcos\_items/2.$
$bridger : sim\_content\_items/2.$
$bridger : sim\_cosine\_items/2.$
$bridger : sim\_cosine\_users/2.$
$bridger : sim\_mf\_cosine\_items/2.$
$bridger : sim\_mf\_cosine\_users/2.$
$bridger : sim\_mf\_euclidean\_items/2.$
$bridger : sim\_mf\_euclidean\_users/2.$
$bridger : sim\_pearson\_users/2.$
$bridger : sim\_pearson\_items/2.$

7 LEARNED MODELS

Below we show the rules learned by our approach (ESMS), Path ranking algorithm (PRA) and Boost(BOOST) for one of the folds on the CORA and the YELP dataset. Note that model weights are relative.

7.1 CORA DATASET

Model learned by ESMS

$$0.07 : TARGETS(A0, A2) \wedge SAMEBIB(A0, A1) \wedge SAMEBIB(A1, A2) \rightarrow SAMEBIB(A0, A2)$$
$$0.018 : TARGETS(A0, A2) \wedge TITLE(A0, A1) \wedge \_TITLE(A1, A2) \rightarrow SAMEBIB(A0, A2)$$
$$0.018 : TARGETS(A0, A2) \wedge VENUE(A0, A1) \wedge \_VENUE(A1, A2) \rightarrow SAMEBIB(A0, A2)$$

Model learned by BOOST

$$0.35 : TARGETS(A0, A1) \wedge VENUE(A1, A2) \wedge VENUE(A0, A2) \rightarrow SAMEBIB(A0, A1)$$
$$0.35 : TARGETS(A0, A1) \wedge TITLE(A0, A2) \wedge TITLE(A1, A2) \rightarrow SAMEBIB(A0, A1)$$
$$0.37 : TARGETS(A0, A0) \wedge AUTHOR(A0, A1) \rightarrow SAMEBIB(A0, A0)$$

Model learned by PRA

$$0.07 : TARGETS(A0, A2) \wedge SAMEBIB(A0, A1) \wedge SAMEBIB(A1, A2) \rightarrow SAMEBIB(A0, A2)$$

7.2 YELP DATASET

Model learned by ESMS

$$0.01 : 1.00 * RATING(A0, A1) = 1.00$$
$$0.07 : AVG\_ITEM\_RATING(A0) \wedge ITEM(A0) \wedge RATED(A1, A0) \rightarrow RATING(A1, A0)$$
$$0.07 : AVG\_USER\_RATING(A0) \wedge USER(A0) \wedge RATED(A0, A1) \rightarrow RATING(A0, A1)$$
$$0.05 : RATING(A, B) \wedge SIM\_ADJCOS\_ITEMS(B, C) \wedge RATED(A, B) \wedge RATED(A, C) \rightarrow RATING(A, C)$$
$$0.07 : SGD\_RATING(A0, A1) \wedge RATED(A0, A1) \wedge RATED(A0, A1) \rightarrow RATING(A0, A1)$$
$$0.05 : SIM\_PEARSON\_USERS(A0, A1) \wedge RATING(A1, A2) \wedge RATED(A1, A2) \wedge RATED(A0, A2) \rightarrow RATING(A0, A2)$$
$$0.05 : RATING(A, B) \wedge SIM\_PEARSON\_USERS(A, C) \wedge RATED(A, B) \wedge RATED(C, B) \rightarrow RATING(C, B)$$
$$0.01 : 1.00 * RATING(A0, A1) = 0.00$$
$$0.07 : SIM\_MF\_EUCLIDEAN\_USERS(A0, A1) \wedge RATING(A1, A2) \wedge RATED(A1, A2) \wedge RATED(A0, A2) \rightarrow RATING(A0, A2)$$

Model learned by BOOST

$$0.12 : SIM\_PEARSON\_USERS(A, A1) \wedge RATED(A, A1) \rightarrow RATING(A, A1)$$
$$0.10 : AVG\_USER\_RATING(A) \wedge RATED(A, A1) \rightarrow RATING(A, A1)$$
$$0.15 : RATED(A1, A) \wedge SIM\_PEARSON\_ITEMS(A, A1) \rightarrow RATING(A1, A)$$
$$0.07 : AVG\_USER\_RATING(A) \wedge SIM\_PEARSON\_USERS(A, A1) \wedge RATED(A, A1) \rightarrow RATING(A, A1)$$
$$0.06 : AVG\_USER\_RATING(A) \wedge SIM\_PEARSON\_ITEMS(B, A1) \wedge RATED(A, B) \rightarrow RATING(A, B)$$
$$0.06 : AVG\_USER\_RATING(B) \wedge RATED(A, A1) \wedge SIM\_PEARSON\_USERS(A, B) \rightarrow RATING(A, A1)$$
$$0.06 : AVG\_USER\_RATING(A) \wedge SIM\_MF\_COSINE\_ITEMS(B, A1) \wedge RATED(A, B) \rightarrow RATING(A, B)$$
$$0.05 : RATED(A1, A) \wedge SIM\_PEARSON\_ITEMS(A, B) \wedge AVG\_ITEM\_RATING(B) \rightarrow RATING(A1, A)$$
$$0.09 : ITEM\_PEARSON\_RATING(A, B) \wedge RATED(A, B) \rightarrow RATING(A, B)$$
$$0.10 : RATED(A, B) \wedge SGD\_RATING(A, B) \rightarrow RATING(A, B)$$

Model learned by PRA

The PRA model had over 75 rules. Here, we shows some of the top weighted rules.

$$0.10 : ITEM\_PEARSON\_RATING(R2, R3) \wedge SIM\_COSINE\_USERS(R2, R1) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : BPMF\_RATING(R1, R2) \wedge SIM\_COSINE\_ITEMS(R3, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : SGD\_RATING(R1, R2) \wedge SIM\_COSINE\_ITEMS(R3, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : ITEM\_PEARSON\_RATING(R1, R2) \wedge SIM\_COSINE\_ITEMS(R3, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : SIM\_ADJCOS\_ITEMS(R3, R2) \wedge SGD\_RATING(R1, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : BPMF\_RATING(R1, R2) \wedge RATED(R1, R3) \wedge SIM\_ADJCOS\_ITEMS(R2, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : SIM\_ADJCOS\_ITEMS(R3, R2) \wedge BPMF\_RATING(R1, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : SIM\_ADJCOS\_ITEMS(R3, R2) \wedge ITEM\_PEARSON\_RATING(R1, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : SGD\_RATING(R1, R2) \wedge RATED(R1, R3) \wedge SIM\_ADJCOS\_ITEMS(R2, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : ITEM\_PEARSON\_RATING(R1, R2) \wedge RATED(R1, R3) \wedge SIM\_ADJCOS\_ITEMS(R2, R3) \rightarrow RATING(R1, R3)$$

$$0.11 : SGD\_RATING(R1, R2) \wedge SIM\_COSINE\_ITEMS(R2, R3) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.10 : SIM\_COSINE\_ITEMS(R2, R3) \wedge BPMF\_RATING(R1, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

$$0.11 : SIM\_COSINE\_ITEMS(R2, R3) \wedge ITEM\_PEARSON\_RATING(R1, R2) \wedge RATED(R1, R3) \rightarrow RATING(R1, R3)$$

8 TIMING EXPERIMENT

We evaluate the runtimes for the proposed PPLL weight learning approach and the standard Maximum Likelihood Estimate (MLE) approach. Given a set of rules, PPLL compute the weights only once for each rule. The presence of other rules in the model does not affect the weight of a rule.

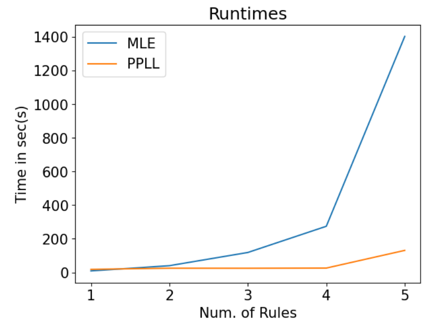

Figure 1: **Runtime for weight learning**: Runtime increases exponentially for MLE but increases linearly for PPLL.

However, since MLE couples all the rules, we need to compute the weights for each subset of the rules and select the best model. Fig. 1 shows the runtimes in seconds for PPLL and MLE as the number of rules in the model increases from 1 to 5. We observe that the runtimes increase exponentially for MLE but increases linearly for PPLL. The decoupling of the rules in weight learning help scale our approach to models with larger sets of rules.