[45] S. Ana, J. Yoob, and S. Choi, "Manifold-respecting discriminant non-negative matrix factorization [j]," *Pattern Recognition Letters*, 2011.

APPENDIX

A. Message Interpolation (M-module)

The approximation, $\mathbf{X}^M \approx \mathbf{X}^G$, is developed as follows. First, if a source $\mathcal{S}_i$ posted, retweeted, or liked claim $\mathcal{C}_j$ in our data set (i.e., $x_{ij} = 1$ in matrix $\mathbf{X}$), then we know that the source endorses that claim (i.e., $x^M_{ij} = 1$ in matrix $\mathbf{X}^M$). The question is, what to do when $x_{ij} = 0$? In other words, we need to estimate the likelihood that the source endorses a claim, when no explicit observations of such endorsement were made. We do so by considering the claim similarity matrix $\mathbf{D}$. If source $\mathcal{S}_i$ was observed to endorse claims $\mathcal{C}_k$ similar to $\mathcal{C}_j$, then it will likely endorse $\mathcal{C}_j$ with a probability that depends on the degree of similarity between $\mathcal{C}_j$ and $\mathcal{C}_k$. Thus, when $x_{ij} = 0$, we can estimate x^M_{ij} by weighted sum interpolation:

$$x^M_{ij} = \sum_{k:\ x_{ik} \neq 0} d_{kj} \cdot x_{ik} \tag{5}$$

To compute matrix $\mathbf{D}$, in this work, we first compute a bag-of-words (BOW) vector w_j for each claim j. We then normalize it using vector L_2 norm, $\bar{w_j} = w_j / \|w_j\|_2$. We select non-zero entries x_{ij} in each row i of $\mathbf{X}$ as *medoids* $\{\bar{w_j} \mid x_{ij} \neq 0\}$. We assume that claims close to any of the *medoids* could also be endorsed by $\mathcal{S}_i$ as well. Based on that, we use:

$$d_{kj} = \phi(\|\bar{w_j} - \bar{w_k}\|) \tag{6}$$

in Equation (5). A Gaussian radial basis function (RBF) is used for $\phi(r) = e^{-(\epsilon r)^2}$. If the resulting value of x^M_{ij} is less than 0.2, we regard that it is far from all of the *medoids* and set it back to 0. In the experiment, ϵ is set 0.5 for sythetic datset and 0.05 for both Eurovision 2016 and Global Warming.

B. Social Graph Convolution (S-module)

To further improve our estimation of matrix $\mathbf{X}^G$, denoted by $\mathbf{X}^{MS}$, we consider the social dependency matrix $\mathbf{A}$.

The fundamental insight we would like to leverage is that users that are close in the social graph, $\mathbf{A}$, are likely to endorse the same claims, even if an explicit endorsement was not observed in the data set. Thus, we consider the social dependency matrix $\mathbf{A}$ (user-user retweet frequency) and compute the a degree matrix $\mathbf{F}$ by summing each row of $\mathbf{A}$ and the random walk normalized adjacency is denoted as $\tilde{\mathbf{A}_{rw}} = \mathbf{F}^{-1}\mathbf{A}$. We define our propagation operator based on $\tilde{\mathbf{A}_{rw}}$ with self-loop re-normalization, $\bar{\mathbf{A}_{rw}} \leftarrow \frac{1}{2}\tilde{\mathbf{F}_{rw}}^{-1}(\tilde{\mathbf{A}_{rw}} + I)$. Thus, the new source-claim network is given by,

$$\mathbf{X}^{MS} = \bar{\mathbf{A}_{rw}} X^M, \tag{7}$$

where each row of $\bar{\mathbf{A}_{rw}}$ adds up to 1. The effect of the propagation operator is to convolve the information from 1-hop neighbors, while preserving half of the information from itself. Note that, we deem dependency beyond 1-hop too weak to capture, so we do not consider $\mathbf{A}^n$, where $n > 1$. From a macroscopic perspective, this social graph convolution recovers some of the possible source-claim connections and also enforces the smoothness of matrix $\mathbf{X}^{MS}$.

C. Overall Loss and Optimization

Given a belief mixture matrix, $\mathbf{B}$, we now factorize $\mathbf{X}^{MS}$ to estimate matrices $\mathbf{U}$ and $\mathbf{M}$ that decide the belief regions associated with sources and claims, respectively. (e.g., the estimated belief for claim $\mathcal{C}_j$ is given by the index of maximum entry in the j_{th} row of $\mathbf{M}$).

Regularization. To avoid model overfitting, we include widely used L_2 regularization. Also, we enforce the sparsity of $\mathbf{U}$ and $\mathbf{M}$ by introducing L_1 norm. The overall objective function becomes (defined by the Forbenious-norm),

$$\begin{aligned} J = &\|\mathbf{X}^{MS} - \mathbf{U}\mathbf{B}\mathbf{M}^\top\|_F^2 + \lambda_1\|\mathbf{U}\|_F^2 + \lambda_1\|\mathbf{M}\|_F^2 \\ &+ \lambda_2\|\mathbf{U}\|_1 + \lambda_2\|\mathbf{M}\|_1. \end{aligned} \tag{8}$$

We rewrite J using matrix trace function $tr(\cdot)$,

$$\begin{aligned} J = &tr(\mathbf{X}^{MS^\top}\mathbf{X}^{MS}) - 2tr(\mathbf{X}^{MS^\top}\mathbf{U}\mathbf{B}\mathbf{M}^\top) \\ &+ tr(\mathbf{U}\mathbf{B}\mathbf{M}^\top\mathbf{M}\mathbf{B}^\top\mathbf{U}^\top) + \lambda_1 tr(\mathbf{U}^\top\mathbf{U}) + \lambda_1 tr(\mathbf{M}^\top\mathbf{M}) \\ &+ \lambda_2\|\mathbf{U}\|_1 + \lambda_2\|\mathbf{M}\|_1. \end{aligned} \tag{9}$$

We minimize J by gradient descent. Since only the non-negative region is of our interests, derivatives of L_1 norm is differentiable in this setting. By referring to gradient of traces of product with constant matrix A, $\nabla_{\mathbf{X}} tr(\mathbf{A}\mathbf{X}) = \mathbf{A}^\top$ and $\nabla_{\mathbf{X}} tr(\mathbf{X}\mathbf{A}\mathbf{X}^\top) = \mathbf{X}(\mathbf{A} + \mathbf{A}^\top)$, the partial derivative of J w.r.t. $\mathbf{U}$ and $\mathbf{M}$ are calculated as,

$$\nabla_{\mathbf{U}} = -2\mathbf{X}^{MS}\mathbf{M}\mathbf{B}^\top + 2\mathbf{U}\mathbf{B}\mathbf{M}^\top\mathbf{M}\mathbf{B}^\top + 2\lambda_1\mathbf{U} + \lambda_2\mathbf{1},$$

$$\nabla_{\mathbf{M}} = -2\mathbf{X}^{MS^\top}\mathbf{U}\mathbf{B} + 2\mathbf{M}\mathbf{B}^\top\mathbf{U}^\top\mathbf{U}\mathbf{B} + 2\lambda_1\mathbf{M} + \lambda_2\mathbf{1}.$$

The gradient matrix $\nabla_{\mathbf{U}}$ is of dimension $|\mathcal{S}| \times K$, and $\nabla_{\mathbf{M}}$ is of dimension $|\mathcal{C}| \times K$. Estimation step begins by updating $\mathbf{U} \leftarrow \mathbf{U} - \eta\nabla_{\mathbf{U}}$ and $\mathbf{M} \leftarrow \mathbf{M} - \eta\nabla_{\mathbf{M}}$, and η is the step size. Negative values might appear in the learning process, which are physically meaningless in this problem. Thus, we impose the non-negative constraints for $\mathbf{U}$ and $\mathbf{M}$ during the update. A *ReLU*-like strategy is utilized: when any entry of $\mathbf{U}$ or $\mathbf{M}$ becomes negative, it is set to be ξ. In the experiment, we set $\xi = 10^{-8}$, $\lambda_1 = \lambda_2 = 10^{-3}$. Note that the initial entry values of $\mathbf{U}$ and $\mathbf{M}$ are randomized uniformly from $(0, 1)$.

D. Complexity

After *M-module* and *S-module*, non-zero entries in the estimated matrix, $\mathbf{X}^{MS} \approx \mathbf{X}^G$, are still far fewer than $|\mathcal{S}| \times |\mathcal{C}|$. We consider to use sparse matrix multiplications and avoid dense intermediate matrices, which makes the computation efficient. Note that, K (number of beliefs) is picked according to the dataset, and it typically satisfies $K \ll min(|\mathcal{S}|, |\mathcal{C}|)$. During the estimation, we generalize standard NMF multiplicative update rules [32] for our tri-factorization,

$$\eta_{\mathbf{U}} = \frac{1}{2}\frac{\mathbf{U}}{\mathbf{U}\mathbf{B}\mathbf{M}^\top\mathbf{M}\mathbf{B}^\top}, \quad \eta_{\mathbf{M}} = \frac{1}{2}\frac{\mathbf{M}}{\mathbf{M}\mathbf{B}^\top\mathbf{U}^\top\mathbf{U}\mathbf{B}}. \tag{10}$$

Algorithmically, updating $\mathbf{U}$ and $\mathbf{M}$ takes $O(K|\mathcal{S}||\mathcal{C}|)$ per iteration. We could also take the advantages of the structure of $\mathbf{B}$, and reduce the complexity to $O(|\mathcal{S}||\mathcal{C}|)$, identical to typical NMF. The number of iterations before the empirical convergence is usually no more than 200 for random initialization, and thus we claim that our model is scalable and efficient.