Supplemental Material

Appendix A: The Proof of Lemma 1

We prove the cases of logistic log-loss and hinge loss in Lemma 1 respectively.

Proof. For the case with logistic log-loss, we directly follow the data-augmentation strategy from (Polson, Scott, and Windle 2013). Let X follow a Polya-Gamma distribution, denoted by $X \sim \mathcal{PG}(a, b)$, that is

$$X = \frac{1}{2\pi^2}\sum_{d=1}^{\infty}\frac{g_d}{(d-1/2)^2 + b^2/(4\pi^2)}, \tag{17}$$

where $a > 0$ and $b \in \mathcal{R}$ are parameters and each $g_d \sim \mathcal{G}(a, 1)$ is an independent Gamma random variable. The main result of (Polson, Scott, and Windle 2013) provides an alternative expression for the form of φ_1 in Eq. (5) by incorporating an augmented variable λ:

$$\varphi_1(\tilde{y}_{ij}|Z_i, Z_j, U) = \frac{1}{2^c}\int_0^\infty \exp\Big(\kappa_{ij}\omega_{ij} - \frac{\lambda_{ij}\omega_{ij}^2}{2}\Big)\phi(\lambda_{ij})\mathrm{d}\lambda_{ij}, \tag{18}$$

where $\kappa_{ij} = c(\tilde{y}_{ij} - \frac{1}{2})$ and $\phi(\lambda_{ij}) = \mathcal{PG}(\lambda_{ij}; c, 0)$.

For the case with hinge loss, we take the advantage of data augmentation for support vector machines (Polson and Scott 2011) and φ_2 in Eq. (6) can be represented as a scale mixture of Gaussian distributions:

$$\varphi_2(y_{ij}|Z_i, Z_j, U) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_{ij}}}\exp\Big(-\frac{(\lambda_{ij} + c\zeta_{ij})^2}{2\lambda_{ij}}\Big)\mathrm{d}\lambda_{ij}, \tag{19}$$

where $\zeta_{ij} = \ell - y_{ij}\omega_{ij}$ and λ_{ij} is the augmented variable. By reformulating similar terms in Eq. (19), we have:

$$\begin{aligned}\varphi_2(y_{ij}|Z_i, Z_j, U) \propto &\int_0^\infty \frac{1}{\sqrt{2\pi\lambda_{ij}}}\exp\Big(-\frac{1}{2}\Big(\frac{c^2\ell^2}{\lambda_{ij}} + \lambda_{ij}\Big)\Big)\\ &\exp\Big(cy_{ij}\Big(1 + \frac{c\ell}{\lambda_{ij}}\Big)\omega_{ij} - \frac{c^2\omega_{ij}^2}{2\lambda_{ij}}\Big)\mathrm{d}\lambda_{ij}\\ \propto &\int_0^\infty \exp\Big(\kappa_{ij}\omega_{ij} - \frac{\rho_{ij}\omega_{ij}^2}{2}\Big)\phi(\lambda_{ij})\mathrm{d}\lambda_{ij}\end{aligned} \tag{20}$$

where $\kappa_{ij} = cy_{ij}(1 + c\ell\lambda_{ij}^{-1})$, $\rho_{ij} = c^2\lambda_{ij}^{-1}$ and $\phi(\lambda_{ij}) = \mathcal{GIG}(\frac{1}{2}, 1, c^2\ell^2)$. Given the results of Eq. (18) and Eq. (20), Lemma 1 holds true. □

Appendix B: Closer Analysis on AstroPh dataset

Here, we provide more closer analysis on AstroPh dataset which is much larger than the NIPS dataset.

Sensitivity to Burn-In Fig. 4(a) shows the AUC scores on testing data with respect to the number of burn-in steps on AstroPh dataset. We can observe that all our variant models converge quickly to stable results, similar as on NIPS dataset. Our DLFRMs with full weight matrix (e.g., DLFRM^l, DLFRM^h, stoDLFRM^l and stoDLFRM^h) converge quickly within 10 steps. The diagDLFRMs need more steps to converge, but still within 40 steps to converge to stable results. These results demonstrate the stability of our Gibbs sampling algorithm.

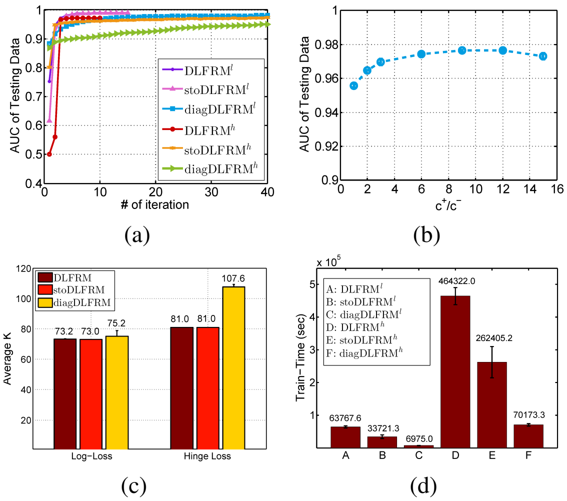

Figure 4: (a) Sensitivity of burn in iterations; (b) Sensitivity of c^+/c^- with diagDLFRM^l; (c) Average latent dimension K; (d) Training time of various models on AstroPh dataset.

Sensitivity to Parameter c We analyze how the regularization parameter c handles the imbalance in real networks using diagDLFRM^l, which is very efficient (see Fig. 4(d)). Following the settings on NIPS dataset, we change the ratio of c^+/c^- for diagDLFRM^l from 1 to 15 with all the parameters selected by the development set. As shown in Fig. 4(b), the AUC score increases when c^+/c^- becomes larger and the prediction performance is stable in a wide range (e.g., $6 < c^+/c^- < 12$). These observations again demonstrate that using a larger c^+ than c^- can effectively deal with the imbalance issue and our setting ($c^+ = 10c^-$) is reasonable.

Latent Dimensions Our variant models take the advantage of nonparametric technique to automatically learn the dimension of the latent features as shown in Fig. 4(c). We can see that diagDLFRMs generally need more features than DLFRMs because the simplified weight matrix U does not consider pairwise interactions between features. Moreover, DLFRM^h needs more features than DLFRM^l, possibly because of the non-smoothness nature of hinge loss. The small variance of each method suggests that the latent dimensions are stable in independent runs with random initializations.

Running Time The training time of our variant models on AstroPh dataset is shown in Fig. 4(d). We can see that for this relatively large network (with tens of thousands of entities and millions of links), the least time we need to obtain the good AUC score is only about 7×10^3 seconds. As on NIPS dataset, DLFRM^h takes more time for training than DLFRM^l and this phenomenon is more obvious here due to the scalability of the network. The reason is that DLFRM^h often converges slower (see Fig 4(a)) with a larger latent dimension K (see Fig. 4(c)). As discussed before, stoDLFRMs are more effective. When a full weight matrix U is used, training time per iteration increases exponentially with respect to K. Therefore, diagDLFRMs are much more efficient due to the linear increase of training time per iteration with respect to K.

Overall, DLFRMs are stable and improve prediction performance efficiently .