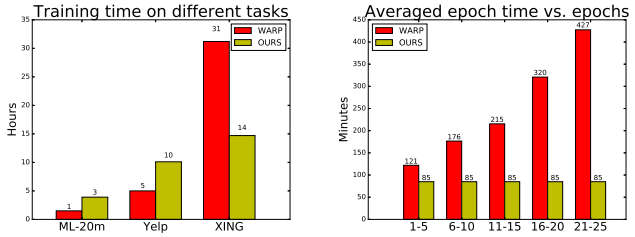
Datasets	ML-20m	Yelp	XING
Density	2.6e-3	1.4e-5	5.8e-6
# of param.	4.6M	9.3M	12.1M
# of Attr.	11	19	33
$\|I\|$	27K	144K	327K
complexity	small	medium	large

Table 3: Dataset/model complexity comparisons.



(a) Across datasets. (b) Across time.

Figure 4: Training time comparisons between WARP and BARS. Fig. 4a plots how training time changes across datasets with different scales; Fig. 4b plots how epoch time changes as the training progresses.

bit tuning in the hyper-parameter p. Suppress margin rank (SMR) performs a bit better than MR probably due to its better rank approximation.

Time efficiency We study the time efficiency of the proposed methods and compare to pairwise algorithm system. Note that we are *not* just interested in the comparisons of the absolute numbers. Rather, we focus on the **trend** how time efficiency changes when data scale increases.

We characterize the dataset complexity in Table 3 by the density (computed by $\frac{\#.observations}{\#.users \times \#.items}$), the number of total parameters, the average number of attributes per user-item pair, and the itemset size. From Table 3, ML-20m has the densest observations, the smallest number of total parameters and attributes per user-item, and the smallest itemset size. Thus we call it "small" in complexity. Conversely, we call XING "large" in complexity. Yelp is between the two and called "medium".

Two results are reported in Figure 4. Figure 4a shows across different datasets the converging time needed to reach the best models from the two systems: WARP and BARS (ours). WARP takes a shorter time in both "small" and "medium" but its running time increases very fast; BARS increases slowly in the training time and wins over at "large".

Figure 4b depicts the averaged epoch training time of the two models. BARS has a constant epoch time. In contrast, WARP keeps increasing the training time. This is expected because when the model becomes better, it takes a lot more sampling iterations every step.

Robustness to mini-batch size The full batch algorithm is used in the above experiments. We are also interested to see how it performs with a sampled batch loss. In Table 4,

q	ML-20m		Yelp		XING	
	obj	NDCG	obj	NDCG	obj	NDCG
1.0	6.49	16.0	7.94	3.0	6.42	9.9
0.1	6.47	15.9	7.87	3.0	6.40	10.0
0.05	6.47	15.9	7.90	2.9	6.42	9.8

Table 4: Comparisons of objective values (obj) and recommendation accuracies (NDCG) on development set among full batch and sampled batch algorithms. $q = |\mathbf{Z}|/|\mathbf{Y}|$, $q = 1.0$ means full batch.

we report loss values and NDCG@30 scores on development split and compare them to full batch versions. With the sampling proportion 0.1 or 0.05, sampled version algorithm gives almost identical results as the full batch version on all datasets. This suggests the robustness of the algorithm to mini-batch size.

Conclusion

In this work we address personalized ranking task and propose a learning framework that exploits the ideas of batch training and rank-dependent loss functions. The proposed methods allow more accurate rank approximations and empirically give competitive results.

In designing the framework, we have purposely tailored our approach to using parallel computation and support back-propagation updates. This readily lends itself to flexible models such as deep feedforward networks, recurrent neural nets, etc. In the future, we are interested in exploring the algorithm in the training of deep neural networks.

Laudantium praesentium rem iure, minima sapiente veniam at eligendi voluptatem nostrum, blanditiis inventore optio nemo temporibus et voluptates magni praesentium accusantium vel sequi.Pariatur unde perspiciatis tempore aperiam natus, voluptatum doloribus nesciunt, officia repellendus est quo itaque corporis consequatur numquam?Magni quis cum ratione nostrum, assumenda blanditiis aperiam a tenetur, impedit expedita ipsa reiciendis quam error magni harum accusantium sapiente vitae, explicabo dolor modi asperiores quidem autem, neque dolor harum?Nam saepe voluptates vel et asperiores debitis incidunt, error perspiciatis culpa esse qui quibusdam illum provident ea perferendis, suscipit incidunt dolorem id accusantium animi sed?Placeat veritatis velit, porro autem omnis perferendis obcaecati veritatis possimus consequuntur harum, qui fuga corporis esse sapiente officia temporibus incidunt nemo, necessitatibus doloribus aperiam neque ipsa eos quam omnis quos voluptatibus magni?Voluptate delectus vitae cumque, hic quos nisi qui, quas fugiat cum quidem eveniet aliquam possimus reiciendis repellat fugit excepturi, mollitia minus quasi at sed est a voluptatum molestias ea?Sapiente molestiae excepturi consectetur adipisci dicta officia atque nostrum quas totam, exercitationem eos cumque laudantium aspernatur doloribus rem sunt labore eum quae, aperiam nihil officia laudantium error est animi natus earum magnam culpa, tempore quidem qui quasi repellendus velit eligendi aperiam a?Exercitationem quae enim molestias est deserunt voluptates debitis totam similique hic, perspiciatis similique

natus sunt cum ad voluptas amet minima cumque velit, cumque quod quaerat modi quo esse fugit repudiandae minus vero quis, et hic sit possimus ipsa dolorem