Feature Selection for Facebook Feed Ranking System via a Group-Sparsity-Regularized Training Algorithm

Xiuyan Ni[1*], Yang Yu[2†], Peng Wu[2], Youlin Li[2‡], Shaoliang Nie[3*], Qichao Que[2], Chao Chen[4]
[1]The Graduate Center, City University of New York [2]Facebook Inc.
[3]North Carolina State University [4]Stony Brook University
{xiuyanni.xn, youlin.li, chao.chen.cchen}@gmail.com, {yuy,wupeng,qichao}@fb.com, snie@ncsu.edu

ABSTRACT

In modern production platforms, large scale online learning models are applied to data of very high dimension. To save computational resource, it is important to have an efficient algorithm to select the most significant features from an enormous feature pool. In this paper, we propose a novel neural-network-suitable feature selection algorithm, which selects important features from the input layer during training. Instead of directly regularizing the training loss, we inject group-sparsity regularization into the (stochastic) training algorithm. In particular, we introduce a group sparsity norm into the proximally regularized stochastical gradient descent algorithm. To fully evaluate the practical performance, we apply our method to Facebook News Feed dataset, and achieve favorable performance compared with state-of-the-arts using traditional regularizers.

CCS CONCEPTS

• **Information systems → Content analysis and feature selection**.

KEYWORDS

feature selection; deep neural networks; online learning; group sparsity; proximal regularization

ACM Reference Format:
Xiuyan Ni[1*], Yang Yu[2†], Peng Wu[2], Youlin Li[2‡], Shaoliang Nie[3*], Qichao Que[2], Chao Chen[4]. 2019. Feature Selection for Facebook Feed Ranking System via a Group-Sparsity-Regularized Training Algorithm. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358114

1 INTRODUCTION

In recent years, social networks have become important information resources for users [2, 9]. Everyday, billions of updates, called *social feeds*, appear in social networks. To deliver the most relevant feeds to users, one needs an online feed ranking model [1]. To build an efficient industrial feed ranking system, one major challenge is the extremely high feature dimension. Social feeds come from many different sources and can be of arbitrary formats, e.g., texts, images, and videos. To fully exploit these feeds, tens of thousands of features are extracted. The features are also very sparse, i.e., contain many zeros. An effective and efficient feature selection method is needed to develop a ranking system with desired scalability.

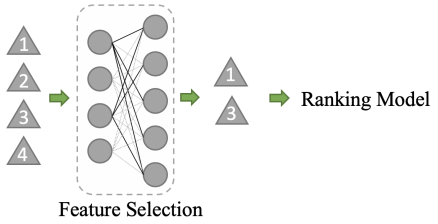


Figure 1: Feature selection of a 2-layer neural network using group sparsity. Out of four input features, two have nonzero weights (black edges) on neural links to the next layer.

For industrial feed ranking systems, a previous popular feature selection method is the gradient boosted decision tree (GBDT) method [4, 14]. In GBDT, the features are transformed depending on the types of features (categorical or continuous), then trained with regularization on weight coefficients. The most important features are selected for downstream prediction models, e.g., logistic regression. GBDT was well adopted in multiple industrial ranking systems due to its flexibility and scalability. However, while features selected by GBDT are effective for classic prediction models, they are not the most important features for a more complex neural network model. One needs a novel feature selection method that is more suitable for neural network based prediction models which have become dominant.

To better select features for a neural network ranking model, we propose a neural feature selection method, which directly selects important features through the training of a neural network. Our method enforces group sparsity on weights of the neural network. In particular, we consider neuron link weights associated with the same input feature as a group. During training, a sparse set of groups are selected, determining the features to use. Fig. 1 illustrates the idea on a two-layer neural network. A natural idea is to directly add group sparsity regularizer to the loss function [6, 11]. However, optimizing a group-sparsity regularized loss function cannot force the coefficients of insignificant features to be exactly zero. One needs a postprocessing procedure to aggregate weights for each feature and to select the important features accordingly [10, 11].

[*] This work is done when Xiuyan Ni and Shaoliang Nie were interns at Facebook.
[†] Current affiliation: Google. [‡] Current affiliation: SmartNews Inc.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358114

This postprocessing step becomes a technical barrier between the neural network for feature selection and the neural network for final prediction.

Instead of regularizing the loss, we inject the group sparsity penalty into the optimization algorithm. We use the proximally regularized online training algorithm, which updates the weights of a model by minimizing an auxiliary function measuring the similarity between the weight vector and the loss gradients. Meanwhile, the auxiliary function has an additional quadratic regularizer to improve stability. Proximally regularized optimization algorithms, e.g., FOBOS [5], Regularized Dual Averaging (RDA) [13] and Follow The Regularized Leader (FTRL) [7] have demonstrated their power in training online models. In particular, FTRL-Proximal has been applied to Ads-click prediction at Google [8]. Nonetheless, all these methods assume a non-neural-network model and use $L1$ penalty to select features. Thus, they cannot solve our problem.

In this paper, we propose a new algorithm for the neural feature selection models, called Group Follow The Regularized Leader (G-FTRL). Our method adds a sparse group lasso regularizer directly to the FTRL optimizer. By directly regularizing weight-updates, we are able to set the coefficients of insignificant features to exactly zero (Eq. 3). With a small number of significant features, we can train light-weight prediction models and apply them to the ranking system. For empirical evaluation, we compare our method with an existing GBDT feature selection method in large scale setting. Our method successfully improves feed ranking performance. In summary, we make the following contributions:

- We use group sparsity to select features by directly training a neural network. The method is compatible with a broad class of neural network based prediction models and can select the most important features for the networks.
- We add the group lasso regularizer to the optimizer instead of the loss. This ensures weights of insignificant features become exactly zeros, which avoids a postprocessing step to select features.
- Our method achieves better performance without using very deep neural networks, which significantly improves training efficiency, especially for industrial settings.

2 METHOD

We first briefly introduce the News Feed ranking system and establish the mathematical notations. Next, we discuss the FTRL algorithm and our proposed group lasso optimizer.

News feed ranking system. The ranking process of news feed posts can be divided into four stages: inventory, signals, prediction, and relevancy scores. The inventory collects the stories posted by users. Meanwhiles, the system collects signals that can be used for the ranking, e.g., the average time spent on a post and who posted the story. Each post is described by a high dimensional feature vector based on the signals. We train a model using the feature vectors to predict the likelihood of different actions a user may take on a post, e.g., commenting, liking or sharing. As a final step, we translate the likelihood of different actions into relevancy scores for feed ranking. The overall process can be found in Fig. 2.

As we have mentioned, the feature vectors tend to have very high dimension and be very sparse. Conventional practice further divides the prediction step into two sub-steps: 1) select significant features (signals) from the feature pool using a feature selection algorithm; 2) feed the selected features to the feed ranking prediction models (Fig. 1). In this paper, we focus on the feature selection step.

Algorithm 1 G-FTRL updating algorithm

Parameters: λ, α, β, γ_g

Let $\mathbf{z}_j = 0$, and $n_j = 0$, $j \in \{1, \ldots, D\}$
for $t = 1$ to T **do**
  calculate the gradient $\mathbf{h}_t = \Delta \ell_t(\mathbf{w}_t)$
  for each group $g \in \{1, \ldots, G\}$ **do**
    $z_g = 0$
    for each coordinate $i \in \{1, \ldots, d_g\}$ in group g **do**
      $n_{t,i} = n_{t-1,i} + (h_{t-1,i})^2$
      $\sigma_{t,i} = \frac{1}{\alpha}(\sqrt{n_{t,i}} - \sqrt{n_{t-1,i}})$
      $z_{t,i} = z_{t-1,i} + h_{t,i} - \sigma_{t,i} * w_{t,i}$
      $z_g = z_g + (z_{t,i})^2$
      update $\mathbf{w}_{t+1,i}$ based on Eq. 3
      $\eta_t = (\frac{\beta + \sqrt{n_i}}{\alpha} + \lambda \gamma_g)^{-1}$
    end for
  end for
end for

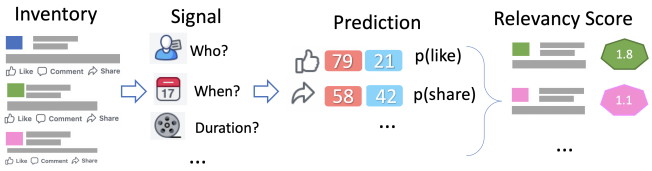


Figure 2: The ranking system at Facebook [3].

Training a neural network feature selection model. We train a neural network model to select significant features. The model is either a multi-label classifier or a regression model depending on the nature of the events. For a large scale time-varying real-world system, we need an online training setting [8] and optimize the expected empirical risk over time: $\mathcal{L}_T(\mathbf{w}_T) := \frac{1}{T}\sum_{t=1}^{T}(\ell_t(\mathbf{w}_T, \mathbf{x}_t) + \Omega_\lambda(\mathbf{w}_T))$ where T is the current time, x_t is the input feature at a previous time t, w_T is the weight vector and $\Omega_\lambda(\cdot)$ is a regularization term ($L1$ and/or group sparsity). However, simplying adding regularization to the loss function could not force the coefficients (or weights) of insignificant features to exactly zero [7].

The (Proximally) Regularized Leader algorithm, or FTRL-Proximal is proposed to train the online model to produce sparse models. Denote by h_t the gradient of the loss at time t, and $\mathbf{h}_{1:t} = \sum_{s=1}^{t} \mathbf{h}_s$ the aggregation of gradients till t. FTRL updates the weights as:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left(\mathbf{h}_{1:t}\mathbf{w} + \sum_{s=1}^{t} \frac{1}{2\eta_s}\|\mathbf{w} - \mathbf{w}_s\|_2^2 + \lambda\|\mathbf{w}_1\|_1\right), \quad (1)$$

where η_s is the learning rate at time s and $\|\cdot\|_p$ denotes Lp norm. This method has been used in Google's Ad click prediction [8] to select significant features. However, the $L1$ regularizer selects features only when each feature is associated with a single weight. For neural network models, each input feature is associated with multiple weights, corresponding to multiple links to the next layer. We need group-wise sparsity for feature selection.

G-FTRL: group FTRL algorithm. We proposed a new algorithm called Group Lasso Follow The Regularized Leader (G-FTRL) by

introducing sparse group lasso regularizer to the FTRL weight updating auxiliary function (Eq. (1)). Previous group-sparsity-based feature selection methods for online learning [10, 12, 15] directly enforce group sparsity on the loss. These methods tend to generate nonzero weights for all features. Instead, our method adds the group-sparsity regularizer to the optimizer and tends to set the weights of insignificant features to exactly zero. In particular, by adding the group sparsity norm into the auxiliary function in Eq. (1), with some derivation, we have

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left(\mathbf{h}_{1:t} - \sum_{s=1}^{t} \sigma_s \mathbf{w}_s\right)\mathbf{w} + \frac{1}{2\eta_t}\|\mathbf{w}\|_2^2 + \lambda \sum_{g=1}^{G} (\sqrt{d_g}\|\mathbf{w}^g\|_2) + \lambda\gamma_g\|\mathbf{w}^g\|_1 + \text{const} \tag{2}$$

Here we assume G groups of weights. Weights of group g is denoted by $\mathbf{w}^g$. d_g is the size of group g. λ is a positive tuning parameter that encourages sparsity. γ_g balances between the group sparsity and entry-wise sparsity. η_t is the learning rate, set to $\gamma_g/\sqrt{t}$. σ_s is the learning rate schedule satisfying $\sigma_{1:t} = 1/\eta_t$.

Let $\mathbf{z}_t = \mathbf{h}_{1:t} - \sum_{s=1}^{t} \sigma_s \mathbf{w}_s$. Denote by $\mathbf{z}_t^g$ its coordinates within group g. The coordinates of $\mathbf{w}_{t+1}$ in group g can be updated in closed form as follows:

$$\mathbf{w}_{t+1}^g = \begin{cases} 0, & \text{if } \| \mathbf{z}_t^g \|_2 \leq \lambda\sqrt{d_g} \\ -\frac{1}{\eta_t}\left[1 - \frac{\lambda\sqrt{d_g}}{\|\mathbf{z}_t^g\|_2}\right]\mathbf{z}_t^g, & \text{otherwise.} \end{cases} \tag{3}$$

We adopt the traditional per-coordinate learning rate schedule for online algorithm [7]. We update the weights at the group level. That is, we update a group of features at the same time with the same criteria. Notice that $\mathbf{z}_t$ can be updated incrementally to save time and space, namely, $\mathbf{z}_t = \mathbf{z}_{t-1} + \mathbf{h}_t + (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})\mathbf{w}_t$. The algorithm is provided in Algorithm 1.

3 EXPERIMENTS

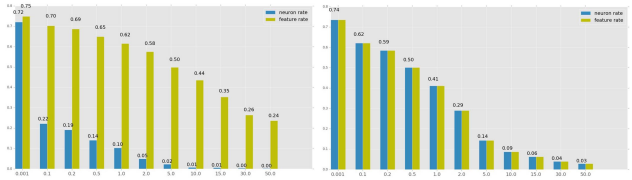
In this section, we empirically evaluate our new feature selection method. All the experiments are conducted on a Facebook News Feed dataset which contains hundreds of millions of feeds with thousands of features extracted for each feed.

We compare our method with the FTRL algorithm, and adding group regularization to loss (GL) method. We conduct a series of experiments to prove the stability of our method in feature selection. The model that we used for the comparisons is the "like" model, which predicts $p(like|\mathbf{w}, \mathbf{x})$, the probability of a feed being liked by the users. Then, we apply our method to all real feed ranking models to compare against the currently used GBDT method. We use 4-layer networks in the feature selection network. We use AUC-ROC and Normalized Cross-Entropy (NE) as quality metrics. In particular, for a given training data set of N examples with labels $y_i \in \{-1, +1\}, i \in \{1, \ldots, N\}$, and estimated probability of an event p_i, the average default mode probability is p. NE is calculated as:

$$\frac{-\frac{1}{N}\sum_{i=1}^{N}(\frac{1+y_i}{2}\log(p_i) + \frac{1-y_i}{2}\log(1-p_i))}{-p\log(p) + (1-p)log(1-p)}.$$

G-FTRL and FTRL. To compare the performance of FTRL optimizer and our G-FTRL in producing sparsity, we train two 4-layer fully connected neural network models, and calculate neuron rate (ratio of non-zero neurons among all neurons in the feature selection layer) and feature rate (ratio of non-zero features among the entire feature set) respectively over the input layer. We use fully connected networks in this work, so each feature in the input layer is connected to all the neurons in the hidden layer followed. A feature is counted as non-zero feature when at least one of the connections to the next layer is not zero.

As shown in Fig. 3, the neuron rate of FTRL reduces with higher regularization. However, the feature rate does not reduce. It is because FTRL treats each coefficient individually, and we consider a feature as non-zero only when all the coefficients connected are zero. For G-FTRL the feature rate and neuron rate reduce simultaneously.

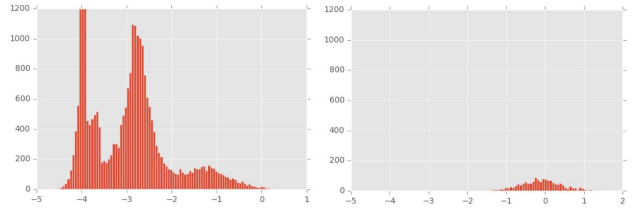


(a) (b)

Figure 3: The neuron rate (blue bars) and feature rate (yellow bars) of FTRL (Fig. (a)) and G-FTRL (Fig. (b)).

G-FTRL and Group Lasso on Loss. As in Section 2, G-FTRL introduces sparsity by adding group lasso regularizer to the optimizer. One of the straightforward method to enforce sparsity is to add the group lasso regularizer to the loss function (GL), which means the loss function becomes: $\ell'(\mathbf{w}, \mathbf{x}) = l(\mathbf{w}, \mathbf{x}) + \lambda \sum_{g=1}^{G}(\sqrt{d_g}\|\mathbf{w}^g\|_2)$.

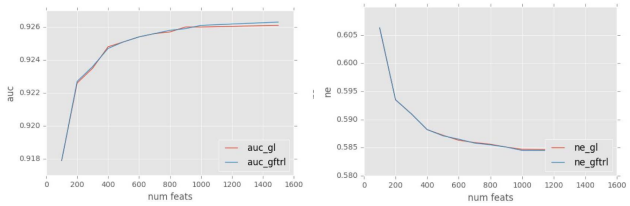
We compare the two methods by checking the distributions of the feature importance (Fig. 4). We use $L2$ norm on all the weights of a feature to represent its importance. We find that the feature rate of GL is 100%, *i.e.*, all candidate features have non-zero weights. While the feature rate of G-FTRL is only 5.12%. This observation is consistent with the theoretical fundamentals of both methods, *i.e.*, GL enforces the weights of unimportant features to arbitrarily small numbers, while G-FTRL directly enforces them to be zero.



(a) (b)

Figure 4: The comparison of GL (Fig. (a)) and G-FTRL (Fig. (b)) in fearture importance distributions. X-axis is the feature weights grouped into 100 bins, and y-axis is the number of features in the corresponding bin.

We also verify the quality of the selected features by selecting a range of top features using both methods and feeding them to the ranking models respectively. Fig. 5 shows the two methods have very close performance in term of AUC and NE.



(a) (b)

Figure 5: The AUC and NE across different regularization weights for GL and G-FTRL. The x-axis is the number of features used. The y-axis is AUC in (a), and NE in (b).

We further compare the feature sets selected by both methods by calculating the number of features overlapping with the GBDT baseline (Fig. 6). We can see that the features selected by both methods have very close overlapping rates.

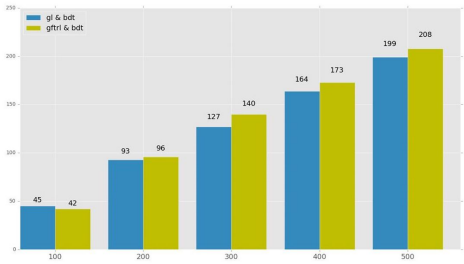
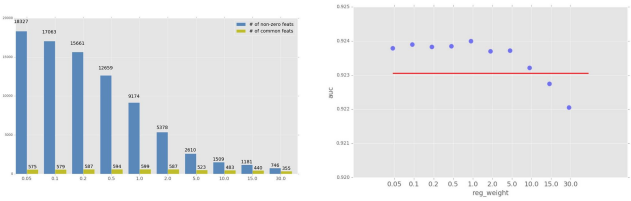


Figure 6: The number of overlapping features of GL (blue bars) and G-FTRL (yellow bars) with the GBDT baseline.

Features Selected and Performance across Different Regularization Weights. To test the stability of G-FTRL, we train a series of feature selection models under different regularization weights, and select the top 1500 features from each model. We calculate the number of overlapping features between G-FTRL and GBDT baseline (Fig. 7 (a)). We find that the number of common features is very stable with the increase of the regularization weight.

We further explore the relationship between the regularization weights and the performance of feature selection models (Fig 7 (b)), where we can find that our method generally outperforms the GBDT baselines with appropriate regularization weights.



(a) (b)

Figure 7: (a) The number of non-zero features under different regularization weights (blue bars), and the number of overlapping features between G-FTRL and GBDT (yellow bars). (b) The AUC across different regularization weights.

Evaluation on Real Models for Social Network. The performance of G-FTRL is evaluated on Facebook feed ranking system. The evaluation metrics are AUC loss reduction (AUC LR = $(a_1 - a_0)/(1 - a_0)$, where a_1 is the AUC for G-FTRL and a_0 is for GBDT), and the NE Reduction (NER = $(n_1 - n_0)/n_0$, where n_1 is NE for G-FTRL and n_0 is for GBDT). Negative NER indicates better performance. We use the results from "like" model as an illustration. The AUC LR of "like" model is 0.20%, and the NER of "like" model is -0.07%. The average AUC LR and NER of all the models is 0.68% and -0.23%, which indicate improved performance for both metrics.

4 CONCLUSIONS AND DISCUSSIONS

We designed a new optimizer algorithm called G-FTRL algorithm that applies group lasso regularizers directly to the optimizer to cut the insignificant features. Compared to adding the group lasso regularizer to the loss function and the original FTRL algorithm, G-FTRL sets the weights of the insignificant features to exactly zero instead of small numbers. We applied G-FTRL to feature selection models in a real feed ranking system and find that our algorithm successfully selects the significant features from candidate feature pools. Our method outperforms the existing GBDT feature selection method and can be integrated easily into the neural ranking system.

ACKNOWLEDGMENTS

The work of Chao Chen was partially supported by NSF IIS-1855759 and CCF-1855760.

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, Rupesh Gupta, Joshua Hartman, Qi He, Anand Iyer, Sumanth Kolar, Yiming Ma, Pannagadatta Shivaswamy, Ajit Singh, et al. 2014. Activity ranking in LinkedIn feed. In *Proceedings of the 20th ACM SIGKDD*. ACM, 1603–1612.
[2] Matthew Burgess, Alessandra Mazzia, Eytan Adar, and Michael J Cafarella. 2013. Leveraging Noisy Lists for Social Feed Ranking.. In *ICWSM*.
[3] Facebook. 2018. *News Feed Ranking in Three Minutes Flat.* https://newsroom.fb.com/news/2018/05/inside-feed-news-feed-ranking/
[4] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 1–9.
[5] John Langford, Lihong Li, and Tong Zhang. 2009. Sparse online learning via truncated gradient. *JMLR* 10, Mar (2009), 777–801.
[6] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2018. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* 50, 6 (2018), 94.
[7] Brendan McMahan. 2011. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization. In *Proceedings of the Fourteenth International Conference on AISTATS*. 525–533.
[8] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD*. ACM, 1222–1230.
[9] Ibrahim Uysal and W Bruce Croft. 2011. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2261–2264.
[10] Jing Wang, Meng Wang, Peipei Li, Luoqi Liu, Zhongqiu Zhao, Xuegang Hu, and Xindong Wu. 2015. Online feature selection with group structure analysis. *IEEE Transactions on Knowledge and Data Engineering* 27, 11 (2015), 3029–3041.
[11] Jing Wang, Zhong-Qiu Zhao, Xuegang Hu, Yiu-Ming Cheung, Meng Wang, and Xindong Wu. 2013. Online Group Feature Selection.. In *IJCAI*. 1757–1763.
[12] Xindong Wu, Kui Yu, Hao Wang, and Wei Ding. 2010. Online streaming feature selection. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. Citeseer, 1159–1166.
[13] Lin Xiao. 2010. Dual averaging methods for regularized stochastic learning and online optimization. *JMLR* 11, Oct (2010), 2543–2596.
[14] Zhixiang Xu, Gao Huang, Kilian Q Weinberger, and Alice X Zheng. 2014. Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD*. ACM, 522–531.
[15] Haiqin Yang, Zenglin Xu, Irwin King, and Michael R Lyu. 2010. Online learning for group lasso. In *Proceedings of the 27th ICML*. 1191–1198.