2015 IEEE International Conference on Big Data (Big Data)

Factorization Machines with Follow-The-Regularized-Leader for CTR prediction in Display Advertising

Anh-Phuong TA
Zebestof company – CCM Benchmark group
Paris, France
Email: anh-phuong.ta@zebestof.com

***Abstract*—Predicting ad click-through rates is the core problem in display advertising, which has received much attention from the machine learning community in recent years. In this paper, we present an online learning algorithm for click-though rate prediction, namely Follow-The-Regularized-Factorized-Leader (FTRFL), which incorporates the Follow-The-Regularized-Leader (FTRL-Proximal) algorithm with per-coordinate learning rates into Factorization machines. Experiments on a real-world advertising dataset show that the FTRFL method outperforms the baseline with stochastic gradient descent, and has a faster rate of convergence.**

***Keywords*-Online advertising; FTRL-Proximal; Factorization machines;**

I. Introduction

Internet advertising is a multi-billion dollar business and is growing rapidly. There are several major channels on the web for online advertising such as display advertising and search advertising. Display advertising is different from the search advertising in that it uses graphical banners placed on the publishers' web pages [1]. In online advertising, advertisers can choose between Cost per Click (CPC), Cost per Action (CPA) or Cost per Impression (CPM) pricing methods to purchase display ads. Among them, CPC is the most popular option, in which advertisers only pay when a user clicks on the ad. As a consequence, click through rate (CTR) prediction, which is defined as the problem of estimating the probability that a user clicks on an ad in a specific context, is crucial to online advertising.

Predicting CTR in display advertising has been widely studied in the literature. Logistic regression (LR) is commonly used in industry due to its ease of implementation and effective performance in large-scale systems [1][2][3][4]. Numerous optimization methods have been applied to train logistic regression models, including Stochastic (online) Gradient Descent (SGD) [5], Newton and Quasi-Newton methods (e.g. L-BFGS), Coordinate Descent [6]. SGD has proved to be effective in solving these kinds of problems, producing good prediction accuracy. Its resulting models, however, are not sparse enough, making them extremely expensive to store in production. Many efforts have been made to produce sparser models, e.g., the FOBOS algorithm [7], the Regularized Dual Averaging (RDA) algorithm [8], the Follow-The-Regularized Leader algorithm [9]. Among these algorithms, FTRL-Proximal has shown to be more effective at producing sparsity and better performance [9].

Despite their success, logistic regression-based methods can not capture higher order interactions (i.e., non-linear information) between features, which have proved to be important in CTR prediction [1]. One can manually select and construct conjunction features from the original ones as the input for LR models. This approach, however, will result in quadratic number of new features, making it really difficult to learn the model. A new line of research based on the use of feature engineering and matrix design, called Factorization Machines (FM), has recently emerged as very successful models for CTR prediction [10]. Indeed, FM combines the advantages of Support Vector Machines with factorization models, which are able to model all interactions between variables even with extreme sparsity of data. An implementation of FM, which supports some optimization algorithms including SGD, Alternating Least Squares (ALS), and Markov Chain Monto Carlo (MCMC), has been provided [11].

In this paper, we attempt to get both the sparsity provided by FTRL-Proximal and the ability of estimating higher order information of FM. To this end, we present the Follow-The-Regularized-Factorized-Leader (FTRFL) algorithm, which incorporates the FTRL-proximal with per-coordinate learning rates into FM.

This method has been used as part of our real-world deployments. The rest of this paper is organized as follows. An overview of our method is given in Section II. Experimental results are described in Section III, and conclusions are drawn in Section IV.

II. Proposed method

A. Model

Our model is based on the second-order FM model [10], which is defined as

$$\phi(w, x) = \sum_{i}^{n} \sum_{j>i}^{n} \langle v_i, v_j \rangle x_i x_j \tag{1}$$

where $\langle ., . \rangle$ is the dot product of two vectors of size k, and the model parameters $w = (v_i, v_j) \in \mathbb{R}^{n \times k}$. $k \in \mathbb{N}_0^+$ is a hyperparameter that defines the number of latent factors. $x \in \mathbb{R}^n$ is a real-value input vector. Note that, the linear and bias terms are excluded from our FM-based model, as they do not give any improvement to the model for the CTR prediction task, and make convergence slower[1]. Here we want to model only second-order interactions between variables.

B. Learning

Given an observed dataset $\{(x^i, y^i) | i = (1, .., L)\}$, in which x^i is of length n representing the input features, and $y^i \in \{1, 0\}$ indicating a click or non-click in an impression, the CTR prediction problem is to learn a function $h(x)$, which can be used to predict the probability of a user clicking on an ad. Similar to the likelihood of LR, $h(x) = Pr(y = 1|x, w) = \sigma(\phi(w, x))$, where $\sigma(a) = \frac{1}{1+\exp(-a)}$ is the sigmoid/logistic function. The model parameters are estimated by minimizing the following regularized loss function:

$$\underset{w}{\operatorname{argmin}} \sum_{i} l(\phi(w, x), y) + \lambda \times r(w) \tag{2}$$

where r is a regularization function, and l is the LogLoss (logistic loss) function, given as

$$l(w) = -y \log(\phi(w, x)) - (1-y) \log(1 - \phi(w, x)) \tag{3}$$

Several learning algorithms, e.g., SGD, have been proposed [11] to solve equation 2. In this work, the per-coordinate FTRL-Proximal algorithm was employed to induce sparsity and yield better performance. In particular, at each step, we update the weight vector on a per-coordinate basis, where the learning rate for each latent factor $v_{i,f}$ at iteration t is set to

$$\eta_{t,i,f} = \frac{\alpha}{\beta + \sqrt{\Sigma_{s=1}^{t} \nabla_{t,i,f}^2}} \tag{4}$$

where α, β are two tunable parameters, as proposed in [9]. Due to the limit of space, the details of the algorithm are omitted here. The interested readers are referred to [9] for similar formulations used with LR.

III. Experimental results

A. Experimental setup

We choose the DownPour SGD method introduced in [12] for parallel processing to train our model. The proposed method was implemented in c++, and run on 28 CPUs with 139G shared memory.

Evaluation metrics: the area under the ROC curve (auROC) was used as a test metric in our experiments.

B. Dataset

To evaluate our method, a portion of 11 days from our real-world dataset[2] was used to build the model parameters, and a subset of the data from the three consecutive days were used to test the trained model. There are 45 millions of impression instances in the training set. The test set contains 30M of impressions. The average CTR in the training set ($\frac{\#clicks}{\#impressions}$) is 0.0063. The early stopping technique was employed in the training to prevent overfitting. In particular, 20% samples from the last days in the training set were randomly selected as the validation set, and we stop the training process when the validation error begins to increase.

C. Features

We consider different feature families, including advertiser (ad ID, campaign ID, ...), publisher (url, publisher ID, ...), user (CRM data) and time features (serve time, click time, ...). The hashing trick, which was made popular by the Vowpal Wabbit learning software, was applied to reduce the dimensionality of the model. In our experiments, the number of bits used for hashing was 22, yielding a model with $4M \times k$ parameters, where k is the dimensionality of the factorization.

[1]This has been confirmed by our experiments. In fact, in our real-world setting we train a separate model for linear terms

[2]Log data from July 2015 at Zebestof: http://www.zebestof.com

D. Results

We compared our method against the standard FM with SGD [11]. For both methods, we started training on a small part of data to choose the tuning parameters, i.e., select the parameters that provide the smallest error on the validation data. Once the parameters are determined, the model is then learned from the entire training set. For the number of latent factors, $k = 20$ is used in all experiments. Table I shows a comparison of our results obtained from the test set with those of the baseline (FM) using SGD. It can be seen that the proposed method overcomes the baseline one by over 7%. In industry, this has significant impact to the overall system performance. In terms of convergence rate, we observed that the FTRFL method normally converges after 5 iterations, while the FM with SGD usually converges after 20 iterations.

Table I
Results for the proposed method and the FM model with SGD. The accuracy is measured by the area under the ROC curve.

Model / Test data	Our method (FTRFL)	FM with SGD
Day 1	**0.9836**	0.9128
Day 2	**0.9818**	0.9105
Day 3	**0.9809**	0.9021

IV. Conclusion

In this paper, we have applied the FTRL-Proximal algorithm with per-coordinate learning rate to FM. The proposed algorithm produces a sparse model, making it applicable to real-world scenarios (i.e., in production, one can store only the non-zero coefficients of the model). Experimental results show that the FTRFL method outperforms the standard FM with SGD, and has a much faster rate of convergence.

References

[1] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 61:1–61:34, Dec. 2014.

[2] K.-c. Lee, B. Orten, A. Dasdan, and W. Li, "Estimating conversion rate in display advertising from past erformance data," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, USA: ACM, 2012, pp. 768–776.

[3] A. Agarwal, O. Chapelle, J. Langford, and C. Cortes, "A reliable effective terascale linear learning system," Tech. Rep., 2011.

[4] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proceedings of the 27th International Conference on Machine Learning, Israel*, 2010.

[5] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Saad, Ed. Cambridge, UK: Cambridge University Press, 1998, revised, oct 2012. [Online]. Available: http://leon.bottou.org/papers/bottou-98x

[6] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, 2003, pp. 928–936.

[7] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, Dec. 2009.

[8] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, vol. 11, 2010.

[9] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica, "Ad click prediction: A view from the trenches," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[10] S. Rendle, "Factorization machines," in *The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pp. 995–1000.

[11] R. Steffen, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 57:1–57:22, May 2012.

[12] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *26th Annual Conference on Neural Information Processing Systems*, 2012.