Factorization Machines for Blog Feedback Prediction

Krisztian Buza[1] and Tomaš Horváth[1,2]

[1] Telekom Innovation Laboratories
Department of Data Science and Engineering
Faculty of Informatics
ELTE - Eötvös Loránd University, Budapest, Hungary
{buza,horvathtamas}@inf.elte.hu
http://t-labs.elte.hu
[2] Institute of Computer Science
Faculty of Science
Pavol Jozef Šafárik University, Košice, Slovakia

Abstract. Estimation of the attention that a blog post is expected to receive is an important text mining task with potential applications in various domains, such as online advertisement or early recognition of highly influential fake news. In the blog feedback prediction task, the number of comments is used as proxy for the attention. Although factorization machines are generally well-suited for sparse, high-dimensional data with correlated features, their performance has not been systematically examined in the context of the blog feedback prediction task yet. In this paper, we evaluate factorization machines on a publicly available blog feedback prediction dataset. Comparing the results with other results from the literature, we conclude that factorization machines are competitive with multilayer perceptron networks, linear regression and RBF network. Additionally, we analyze how parameters (feature weights and interaction weights) of factorization machine are learned.

Keywords: blog feedback prediction, factorization machine, machine learning

1 Introduction

Early recognition of highly influential posts in social media, such as blogs or tweets, is an essential task with various applications. For example, the identification of most visited blogs may be useful in online advertisement scenarios or in order to identify highly influential fake news in advance.

In the blog feedback prediction problem [2], the number of comments serves as a proxy for the attention, i.e., the task is to predict the number of comments that a blog will receive. In order to predict the number of comments, various features are used that refer to the textual content, source (blog or website) or other conditions, e.g. on which day of the week the post was published.

Recently, the blog feedback prediction task received considerable attention, see e.g. [4], [7], [8], [12], [10]. Various models have been used, such as state-of-the-art variants of nearest neighbour regression [4], fuzzy systems [7], ensemble techniques [12], neural networks and decision trees [10].

Motivated by movie recommendation tasks, factorization machines [9] have been developed for supervised machine learning on sparse and high-dimensional data containing correlated features. Factorization machines are especially well-suited to represent correlations efficiently, i.e., given n features, factorization machines are able to represent all the pairwise correlations with $\mathcal{O}(n)$ parameters. Additionally, factorization machines can estimate correlations between features even if the correlation is not expressed explicitly, but has to be inferred. Last, but not least, compared with deep learning, factorization machines require much less training data. For the aforementioned reasons, factorization machines have been recognized more and more in the machine learning community.

Due to the large number of textual features that correspond to the frequency of most predictive words, the data in the blog feedback prediction problem is high dimensional and sparse. Additionally, some of the features are expected to be correlated. Thus, factorization machines are promising candidates for the blog feedback prediction task.

Despite the aforementioned facts, factorization machines have not been systematically studied in context of the blog feedback prediction problem. For completeness, we note that an initial experiment with factorization machines on the blog feedback data have been documented in the first version of a manuscript available at arXiv [13], however, the results were not evaluated in terms of typical evaluation metrics of the blog feedback prediction task, such as AUC or the number of hits. More importantly, the manuscript does not focus on the blog feedback prediction tasks and the aforementioned experiment is not included in subsequent versions of the manuscript, including its current version.

For the above reasons, in this work, we aim to examine whether factorization machines are suitable blog feedback prediction.

The rest of this paper is organized as follows: in Section 2 we review factorization machines. In Section 3 we describe the blog feedback data and our experiments. Finally, we conclude in Section 4 and point out potential directions of future work.

2 Factorization Machines

Given an instance $\mathbf{x} = (x_1, \ldots, x_n)$, a factorization machine [9] of second degree with f factors predicts its label as follows:

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Big(\sum_{k=1}^{f} v_{i,k} v_{j,k} \Big) x_i x_j \tag{1}$$

where $w_0 \ldots w_n$ and $v_{1,1} \ldots v_{n,f}$ are parameters of the model. The later describe the interactions between features, while we refer to $w_1 \ldots w_n$ as *feature weights*.

Algorithm 1 Training the Factorization Machine

Require: Training data D, number of epochs e, learning rate η, standard deviation σ
Ensure: Weights $w_0, w_1, \dots w_k$ and $v_{1,1}, \dots, v_{n,f}$

Initialize $w_0, w_1, \dots w_k$ and $v_{1,1}, \dots, v_{n,f}$ from standard normal distribution with zero mean and standard deviation σ
for epoch in $1 \dots e$ **do**
  for each $(x, y) \in D$ in random order **do**
    $\hat{y} \leftarrow w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Big(\sum_{k=1}^{f} v_{i,k} v_{j,k} \Big) x_i x_j$
    $w_0 \leftarrow w_0 - \eta \, 2(\hat{y} - y)$
    for i in $1 \dots k$ **do**
      $w_i \leftarrow w_i - \eta \, 2(\hat{y} - y) x_i$
    end for
    for i in $1 \dots n$ **do**
      for j in $1 \dots f$ **do**
        $v_{i,j} \leftarrow v_{i,j} - \eta \, 2(\hat{y} - y) \Big(x_i \sum_{k=1}^{n} v_{k,j} x_k - v_{i,j} x_i^2 \Big)$
      end for
    end for
  end for
end for
return $w_0, w_1, \dots, w_k$ and $v_{1,1}, \dots, v_{n,f}$

Given a labelled training dataset, the parameters of the model can be learned with stochastic gradient descent so that the sum of squared errors (or another objective function) is optimized. For details, we refer to Algorithm 1 and [9].

3 Experimental Evaluation

Blog Feedback Data. We performed experiments on the publicly available Blog Feedback Data[3]. The data contains 60021 instances and 281 features, including the target. Each instance refers to a blog post which is described by various features referring to textual content, the source (blog) on which the post was published and other conditions, e.g. on which day of week the post was published. The target is the number of comments that the post received within the next 24 hours. For a more detailed description of the data and how it was created, see [2].

Experimental Protocol. In order to assist reproducibility and comparability, we used the predefined training and test sets associated with the data. We calculated the evaluation metrics (AUC@10 and Hits@10) for each test set and aggregated the results. This is exactly the same protocol as in [2], therefore, the results are directly comparable.

[3] `https://archive.ics.uci.edu/ml/datasets/BlogFeedback`

Table 1. The performance of factorization machine (FM) with $f = 0$ (linear regression) and $f = 3$ factors on the blog feedback prediction task

	Linear regression	FM, $f = 3$
AUC@10	0.864	0.869
Hits@10	4.733	5.117

Evaluation Metrics. In order to evaluate the predictions, we used AUC@10 and Hits@10 which are defined as follows.

For each test split, we consider 10 blog pages that were predicted to have the largest number of feedbacks. We count how many out of these pages are among the 10 pages that received the largest number of comments in reality. We call this evaluation measure Hits@10.

For the AUC, i.e., area under the receiver-operator curve, we considered as positive the 10 blog pages receiving the highest number of comments in the reality. Then, we ranked the pages according to their predicted number of comments and calculated AUC. We call this evaluation measure AUC@10. For both evaluation metrics, higher values indicate better performance.

Hyperparameters of the Factorization Machine. In our experiments, by default, we set the number of factors $f = 3$. In order to find appropriate model parameters, we minimized the sum of squared errors on the training data with stochastic gradient descent. We initialize the parameters (i.e., feature weights W_i and interaction weights $v_{i,k}$) from a standard normal distribution with zero mean and $\sigma = 10^{-8}$. We set the learning rate of stochastic gradient descent to $\eta = 10^{-12}$ and iterated over the training instances 1000 times, in other words: we learned the model in 1000 epochs.

Experimental Results. As can be seen in Tab. 3, factorization machine achieved AUC@10 of 0.869, while the average number of Hits@10 was 5.117. We note that factorization machine is a generalization of linear regression: in particular, a factorization machine with $f = 0$ is equivalent to linear regression, therefore, we show the results for linear regression in Tab. 3.

We also performed experiments with $f = 5$ factors. As the results were very similar to the results with $f = 3$ factors, for simplicity, we only report results with $f = 3$ factors.

Comparison with Results Reported in the Literature. According to the results reported in [2], factorization machine outperforms various multilayer perceptron networks, linear regression and RBF network. In particular, none of these models achieved Hits@10 greater than 5, while the AUC was between 0.8 and 0.85 for these models, see Fig. 1. in [2]. On the other hand, one of the regression trees, M5P, achieved AUC around 0.9, while its performance in terms of Hit@10 was comparable to that of factorization machine.

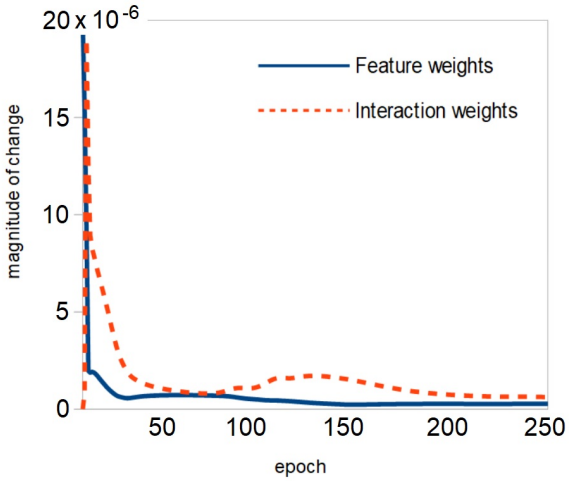

Fig. 1. Magnitude of change of feature weights and interaction weights as function of the number of epochs.

Learning the Parameters of the Factorization Machine. In order to understand how parameters of the factorization machine are learned during stochastic gradient descent, we calculated for each feature weight w_i and for each interaction weight $v_{i,k}$, how much that parameter changed in total during each epoch. Next, we calculated the absolute value of the change for each parameter. Subsequently, we calculated the mean of these absolute values separately for the feature weights and interaction weights. We refer to these mean values as the magnitude of change. We plot the magnitude of change for the first 250 epochs in Fig. 1.

As one can see, at the beginning, the change of both feature weights and interaction weights is relatively large. As a global trend, the magnitude of change of feature weights is decreasing. In contrast, the magnitude of change of interaction weights is decreasing at the beginning, then it begins to increase again around the 70th epoch. Roughly from the 140th epoch on, the magnitude of change of interaction weights is decreasing again.

This analysis indicates that interaction weights may be more difficult to learn than feature weights. The relatively high magnitude of change of the interaction weights between the 100th and 160th epochs may be explained by the assumption that interaction weights can be learned effectively, once feature weights have relatively good values.

4 Conclusions and Outlook

In this paper, we considered the blog feedback prediction task and systematically examined factorization machines for this task. As expected, factorization

machines are competitive models for blog feedback prediction. On the other hand, our analysis revealed that the development of efficient training algorithms may be challenging, because it seems to be the case that interaction weights can only be learned efficiently if feature weights already have reasonable values. Taking this observation into account, one may devise smart training algorithms in the future.

On the long term, it would also be interesting to adapt other techniques for blog feedback prediction, such as hybrid approaches, the incorporation of fuzzy-valued loss functions into factorization machines or model selection based on genetic algorithms, see e.g. [1], [5], [6]. Furthermore, we note that a special variant of the blog feedback prediction task, the so called *personalized blog feedback prediction* task has also been considered in the literature [3]. This motivates the idea of considering the situation as a *game* (in the sense of game theory) in which users have a limited budget of comments, i.e., each user is only able to comment on a limited number of blogs, while the users try to optimize the impact of their comments (payoff). Therefore, one could try approaches based on game theory, such as the one described in [11], for blog feedback prediction.

Acknowledgments. This work was supported by the project no. 20460-3/2018/FEKUTSTRAT within the Institutional Excellence Program in Higher Education of the Hungarian Ministry of Human Capacities.

References

1. Burduk, R., Kurzyński, M.: Two-stage binary classifier with fuzzy-valued loss function. Pattern Analysis and Applications 9(4), 353–358 (2006)
2. Buza, K.: Feedback prediction for blogs. In: Data analysis, machine learning and knowledge discovery, pp. 145–152. Springer (2014)
3. Buza, K., Galambos, I.: An application of link prediction in bipartite graphs: Personalized blog feedback prediction. In: 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications June. pp. 4–7. Citeseer (2013)
4. Buza, K., Nanopoulos, A., Nagy, G.: Nearest neighbor regression in the presence of bad hubs. Knowledge-Based Systems 86, 250–260 (2015)
5. Jackowski, K., Krawczyk, B., Woźniak, M.: Improved adaptive splitting and selection: the hybrid training method of a classifier based on a feature space partitioning. International journal of neural systems 24(03), 1430007 (2014)
6. Jackowski, K., Wozniak, M.: Method of classifier selection using the genetic approach. Expert Systems 27(2), 114–128 (2010)
7. Kaur, H., Pannu, H.S.: Blog response volume prediction using adaptive neuro fuzzy inference system. In: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). pp. 1–6. IEEE (2018)
8. Kaur, M., Verma, P.: Comment volume prediction using regression. International Journal of Computer Applications 151(1) (2016)
9. Rendle, S.: Factorization machines. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. pp. 995–1000. IEEE (2010)

10. Singh, K., Sandhu, R.K., Kumar, D.: Comment volume prediction using neural networks and decision trees. In: IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015), Cambridge, United Kingdom (2015)
11. Suciu, M., Lung, R.I., Gaskó, N., Dumitrescu, D.: Differential evolution for discrete-time large dynamic games. In: Evolutionary Computation (CEC), 2013 IEEE Congress on. pp. 2108–2113. IEEE (2013)
12. Uddin, M.T.: Automated blog feedback prediction with ada-boost classifier. In: Informatics, Electronics & Vision (ICIEV), 2015 International Conference on. pp. 1–5. IEEE (2015)
13. Yamada, M., Lian, W., Goyal, A., Chen, J., Wimalawarne, K., Khan, S.A., Kaski, S., Mamitsuka, H., Chang, Y.: Convex factorization machine for regression. arXiv preprint arXiv:1507.01073v1 (2015)