Predicting Future User Activities with Constrained Non-Linear Models

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

Prediction of future user activities from their history, all past activities, is a challenging problem. One reason is that the number of all potential histories grows exponentially with the length of the history. Recently, deep-learning models have been proposed for solving this problem [4, 8].

It is easy to learn a simple predictor of future user activities, by *averaging* all past activities of the user and then learning an activity classifier from this average representation, for instance by logistic regression [7]. This approach tends to have a high bias, due to using a simple feature representation and model. It is also easy to apply sequence models in deep learning [11] to learn a predictor of future user activities. This approach tends to have a high variance, because of the large number of parameters in the neural network.

To achieve the best of both worlds, low bias and variance, we investigate the value of constraining neural networks when learning user activity models. In particular, we hypothesize that more distant user activities should have a lower impact on future user activities than more recent activities. In a neural network, this would correspond to a constraint that the weights of the neurons that represent distant past activities are lower than those of the recent activities. We implemented this constraint in combination with stochastic gradient descent (SGD) [3, 6]. When the constraint is violated, we project the weights of the neurons back to satisfy the constraint using quadratic programming.

Our paper is organized as follows. In Section 2, we formally describe our models. In Section 3, we introduce our dataset and validate our hypothesis empirically. We observe minor improvements due to enforcing the constraint. Our subsequent analysis revealed that this because even the unconstrained neural network converges to the solution that satisfies our constraint.

Our work is related to [1], and a large body of work on attention modeling in deep learning [9, 2, 10, 12], which apply further constraints over the neural networks weights. In [1] the authors developed a new class of neural network layers that can solve a quadratic program. For example, they can be used to solve a Sudoku puzzle. In contrast, we enforce inequality constraints over the weights of a latent vector. Our work differs from attention models in deep learning in that the attention filter, which is the equivalent to our latent vector, does not impose the inequality constraints and bounds on the weights of the mask. The only constraint is an equality one, that the weights have to sum up to one.

2 Model

We consider the following model of user behavior. The *observation* at time t is denoted by $v_t \in [0,1]^d$. The observation is any vector that represents the activity of the user at time t. The time is discrete. Our goal is to predict some variable of interest at time t, y_t, from the history of the user up to time t, $(v_\ell)_{\ell=1}^{t-1}$.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

A naive approach is to predict y_t directly from $(v_\ell)_{\ell=1}^{t-1}$, the sequence of all past observations without any prepossessing. This approach is problematic because it does not generalize well. In particular, to perform well on unseen test users, it is necessary to generalize from similar users, similar sequences of past observations, in the training data. The number of such sequences is likely to be exponential in the length of the history.

An alternative is to compress the history of the user to gain statistical efficiency. In particular, we can build the *profile* of the user at time t, x_t, which is a function of $(v_\ell)_{\ell=1}^{t-1}$, and then train a predictor of y_t from x_t. One potential choice is the *weighted average profile*

$$x_t = \sum_{\ell=1}^{t-1} \alpha_{t-\ell} v_\ell \,, \tag{1}$$

which is the weighted sum of past observations. A reasonable constraint to enforce is that

$$1 \geq \alpha_1 \geq \cdots \geq \alpha_{t-1} \geq 0 \,, \tag{2}$$

which means that more recent observations are more influential than more distant ones. This profile is more statistically efficient than the complete history for multiple reasons. First, it is permutation invariant, in the sense that the order of observations with similar weights does not matter. Second, the observations are additive.

The weighted profile in (1), although natural, has two shortcomings. First, it is computed by averaging in the space of observations v_t, which may not be the best space for generalization. Second, the problem of predicting y_t is decoupled from that of choosing the profile.

In its most generality, the problem of learning y_t and x_t can be coupled as

$$y_t = f\left(\sum_{\ell=1}^{t-1} \alpha_{t-\ell}\, g(v_\ell)\right) \,, \tag{3}$$

where g is a function that transforms individual observations into the latent space, where their weighted sum forms a profile of the user; and f is a function that maps the profile to predictions. We represent f and g by neural networks. The weighted profile in (1) can be expressed as this profile when g is an identity.

In this work, we learn (3) under the constraint in (2), to study the benefits of constraining rich nonlinear representations. We learn our representation as follows. Let θ_i^f and θ_i^g be the i-th parameters of neural networks f and g, respectively. Fix $(v_1, \dots, v_{t-1}, y_t)$, the history of the user up to time t and the response at time t. We apply gradient descent on this data point and update all model parameters as follows. First, we run backpropagation on the network in (3). Let δ_i^f be the gradient of θ_i^f, $\delta_{i,\ell}^g$ be the gradient of θ_i^g with input v_ℓ, δ_ℓ^α be the gradient of α_ℓ. Then

$$\theta_i^f \leftarrow \theta_i^f - \gamma \delta_i^f \,, \quad \alpha_\ell \leftarrow \alpha_\ell - \gamma \delta_i^\alpha \,, \quad \theta_i^f \leftarrow \theta_i^f - \gamma \sum_{\ell=1}^{t-1} \delta_{i,\ell}^g \,,$$

where $\gamma > 0$ is a learning rate. The above update may violate the constraint in (2). To enforce it, we project α back by solving

$$\alpha \leftarrow \arg\max{}_{a \in [0,1]^{t-1}} \|a - \alpha\|_2 \text{ s.t. } 1 \geq a_1, \ a_1 \geq a_2, \ \dots, \ a_{t-2} \geq a_{t-1}, \ a_{t-1} \geq 0 \,.$$

Note that this is a quadratic program with $t - 1$ variables and t linear constraints, which can be solved efficiently.

3 Experiments

We experiment with the Movielens 20M dataset [5]. The task is to predict the movie genre watched by the user from previously watched movie genres. More precisely, y_t is the watched movie genre at time t and v_ℓ is the indicator vector of movie genres watched at time ℓ. We assume that function g in (3) is identity, and learn both f and α.

We compare four methods. The first one is logistic regression on the weighted user profile in (1). The discount factor is set as $\alpha_\ell = 0.75, 0.9, 1.0$. We choose three scalars and build the feature

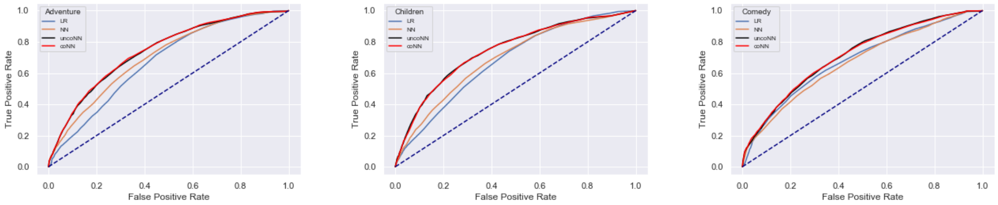

Figure 1: ROC curves showing the performance of LR , NN, uncoNN and coNN methods on the 'Adventure', 'Children' and 'Comedy' genres from the Movielens 20M dataset.

vector concatenating discounted (3) vectors before applying f. We did this to give the models data discounted at different rates, consequently richer features to use. We call this method LR. The second method is the same as the first one, except that we use deep learning instead of logistic regression. We call this method NN. The third method is learning of (3) under the constraint in (2). The method is implemented as described in Section 2 and we call it coNN. The last method is the same as the third one, except that we do not enforce the constraint in (2). We call this method uncoNN.

In Figure 1 we show the ROC curve on the prediction of three genres: Adventure, Comedy, and Children. We selected those genres given their highest presence in the dataset. We see that uncoNN and coNN outperform LR and NN. The methods uncoNN and coNN perform better because learning α, which constraint the past observations, is beneficial. The propose method coNN is on par with uncoNN in the results. After analyzing the values we encounter that uncoNN compute α that satisfy the expected constraints.

References

[1] Amos, B. and Kolter, J.Z.. (2017) *OptNet: Differentiable Optimization as a Layer in Neural Networks*. Proceedings of the 34th International Conference on Machine Learning in PMLR 70:136-145.

[2] Dzmitry Bahdanau and Kyunghyun Cho and Yoshua Bengio (2014) *Neural Machine Translation by Jointly Learning to Align and Translate*. CoRR, Volume 1409.0473.

[3] Bottou, L. (1991) *Stochastic gradient learning in neural networks*. In Proceedings of Neuro-Nimes 91.

[4] Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G.S., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., and Shah, H.(2016) *Wide & Deep Learning for Recommender Systems*. DLRS@RecSys.

[5] Harper, F.M., & Konstan, J.A. (2015). *The MovieLens Datasets: History and Context*. TiiS, 5, 19:1-19:19.

[6] LeCun, Y., Bottou, L., Orr, G., and Muller, K. Efficient backprop. (1998) *In Neural Networks: Tricks of the trade, pp. 5-50*. Springer.

[7] McCullagh, Peter, and John A. Nelder. (1989) *Generalized linear models*. Vol. 37. CRC press.

[8] Wang, R., Fu, B., Fu, G., and Wang, M.(2017) *Deep & Cross Network for Ad Click Predictions*. ADKDD@KDD.

[9] Kelvin Xu and Jimmy Ba and Ryan Kiros and Kyunghyun Cho and Aaron Courville and Ruslan Salakhudinov and Rich Zemel and Yoshua Bengio (2015) *Show, attend and tell: Neural image caption generation with visual attention*. Proceedings of the 32nd International Conference on Machine Learning in PMLR 37:2048-2057.

[10] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H. (2016). *Hierarchical Attention Networks for Document Classification*. HLT-NAACL.

[11] Zhai, S., Chang, K., Zhang, R., and Zhang, Z. (2016). *DeepIntent: Learning Attentions for Online Advertising with Recurrent Neural Networks*. KDD.

[12] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B. (2016). *Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification*. ACL.