Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200680

Using the Generative Adversarial Network to Generate Recommendations

Prosvetov A.V.[a,b,c,1]
[a] *CleverDATA, Russia*
[b] *Space Research Institute, Russian Academy of Science, Russia*
[c] *LANIT, Russia*

Abstract. Widely used recommendation systems do not meet all industry requirements, so the search for more advanced methods for creating recommendations continues. The proposed new methods based on Generative Adversarial Networks (GAN) have a theoretical comparison with other recommendation algorithms; however, real-world comparisons are needed to introduce new methods in the industry. In our work, we compare recommendations from the Generative Adversarial Network with recommendation from the Deep Semantic Similarity Model (DSSM) on real-world case of airflight tickets. We found a way to train the GAN so that users receive appropriate recommendations, and during A/B testing, we noted that the GAN-based recommendation system can successfully compete with other neural networks in generating recommendations. One of the advantages of the proposed approach is that the GAN training process avoids a negative sampling, which causes a number of distortions in the final ratings of recommendations. Due to the ability of the GAN to generate new objects from the distribution of the training set, we assume that the Conditional GAN is able to solve the cold start problem.

Keywords. GAN, Recommendations System. Neural network

1. Introduction

Recommendations can be generated using several approaches. While some of the most popular methods are based on matrix decomposition (user-based and item-based collaborative filtering, Alternating Least Squares, SVD and etc), the alternative methods are based on neural networks (restricted Boltzmann machine, deep autoencoders and etc.). Approaches based on neural networks available to use additional information about the composition of items and the user's purchase sequence in embeddings [1], however, widely used matrix decomposition approaches show significant results either. Despite the successes achieved, several unresolved issues still exist, for example, Cold Start problem.

A recommendation system based on matrix decomposition can give each user a rating for each item, but the estimates obtained are in the form of floating point number, while in cases with high uncertainty level the distribution of rating is required. After training the recommendation system, the production implementation requires regular retraining, model monitoring and tuning, so reinforcement learning is very suitable.

[1] Prosvetov A.V.; E-mail: prosvetov@gmail.com.

The discriminative algorithms try to learn ***P(Y|X)*** directly or try to learn mappings from space of inputs X to the labels {0,1}. The generative algorithms try to learn ***P(X|Y)***. If ***Y*** indicates whether an item is interesting to user, then ***P(X|Y = 1)*** represents the distribution of item features, that are relevant to user. The probability ***P(X|Y = 0)*** in this case represents the distribution of product features for user, that are non-relevant. In the current work we suggest to solve the cold start problem using Generative Adversarial Network (GAN), that generative approach was proposed by I. Goodfellow et al. [2] in 2014. The GAN architecture combined from two neural networks: the first attempts to create new sample from the same distribution as the training set, while the second neural network trains to classify the real and the generated samples. When the training process is completed, a generative model is obtained. The resulting generative model is capable of generating new samples indistinguishable from objects from the training set. The GAN approach was proposed for unsupervised learning initially, however the reinforced learning is close to GAN architecture [3]. That fact gives additional potential for the implementation of a GAN recommendation system in production.

GAN architecture is commonly used to generate images. Currently GAN can achieve photorealistic results for photo [4], but also GAN was successfully used for prediction of gravitational lensing in a certain direction in the distribution of dark matter [5] . GANs were also proposed for the generation of simulated showers of particles in calorimeters of experiments in high-energy physics. [6,7]. Thus, the potential of GAN generalizations can be used by recommendation systems to offer more relevant products and to augment the data, which is in great demand for solving the cold start problem.

Previous attempts to use the GAN in the recommendation system showed a high potential of the approach with respect to typical datasets [8], however, measured offline metrics cannot guarantee the high quality of recommendations in real cases. In our work, we test the GAN architecture in flight recommendations. Our preliminary results has already published in short form [9], the current paper is present the detailed description of experiment, data preparation and network architecture.

2. GAN details

Assume that GAN consist of generator and discrimination subnetworks. The subnetwork architecture can be multi-layer perceptrons. Assume a prior on input noise variables $p_z(z)$ to learn the generator's distribution p_g over data x. The generator takes place a mapping to data space as $G(z;\ \theta_g)$, where ***G*** is multilayer perceptron with parameters θ_g. Assume that the discriminator is a multilayer perceptron $D(x,\ \theta_d)$ that outputs a single scalar. We are able to create train set where ***D(x)*** models the probability that object x came from the real data set rather than form the set of generated objects. The next step will be training ***D*** to maximize the probability of assigning the correct label to both training examples and samples from ***G***. Simultaneously one can train ***G*** to minimize ***log(1 - D(G(z)))***. Thus, ***D*** and ***G*** play the minimax game with value function ***V(G, D)***:

$$min_G max_D V(D,G) = E_{x\sim p_{data}(x)}\left[log\big(D(x)\big)\right] + E_{z\sim p_z(x)}\left[log\left(1 - D\big(G(z)\big)\right)\right] \quad (1)$$

During the train one can take ***k*** steps of optimizing ***D*** and one step of optimizing ***G*** in the loop to avoid overfitting. In this case results in ***D*** being maintained near its optimal solution, so long as G changes slowly enough. In more general case the discriminator and the generator can have convolutional layers [10] and recurrent cells [11]. If the discriminator and the generator are used conditioned information ***y***, than the architecture will be upgraded to a conditional model [12]. To get a recommendation system, we used the information about the historical choice of the user. The conditioning was carried out by feeding an additional input layer with historical information to the discriminator and the generator. Two-player minimax game objective function would be the following:

$$min_G max_D V(D,G) = E_{x\sim p_{data}(x)}\left[log\left(D(x \vee y)\right)\right] + E_{z\sim p_z(x)}\left[log\left(1 - D\left(G(z \vee y)\right)\right)\right] \quad (2)$$

The architecture of conditional adversarial network that was tested in our study is illustrated on Figure 1.

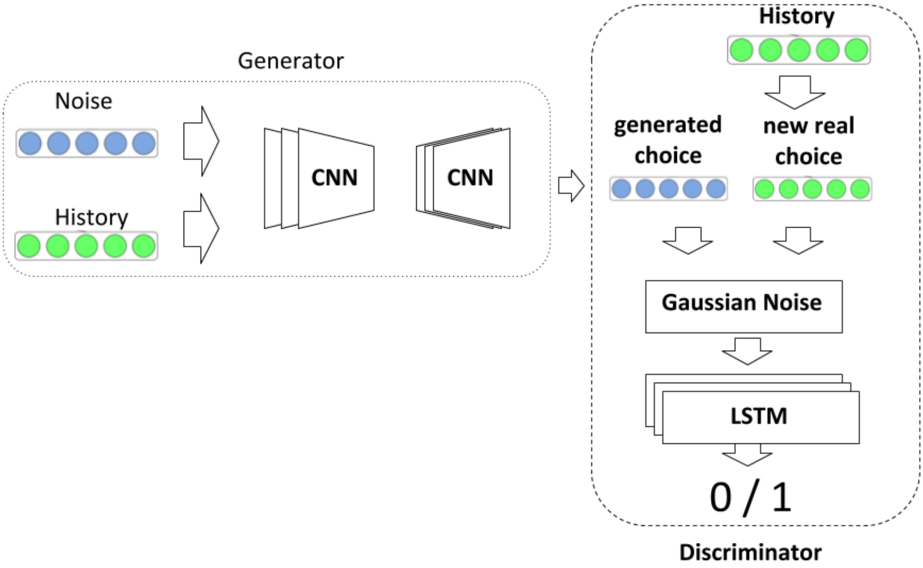

Figure 1. The architecture of recommendation system based on conditional GAN

3. Data preparation

In our experiment, we used a dataset with information about historical air flights provided by our partner. For each user, we used the sequence of airports visited in chronological order. We extracted the home airport from the sequence and coded the airports so that we could work with the sequence as with text. The last airport was extracted from the sequence, the generator was trained to predict the next flight airport, and the discriminator was trained to distinguish between the predicted airport and the real next airport for each user.

4. Recommendation system

Our recommendation system contains generator, discriminator and intermediate layer.

1) The aim of generator is to produce new user selection from previous selections and random noise. The generator network uses the following inputs:
    a) the historical choice of the user
    b) random noise.

   The embeddings for previous selections input are trainable; however one can use a pre-trained embeddings. The output of generator is a dense layer with the activation function of sigmoidal form. The number of user selection options is equal to the size of the output layer.
2) The goal of discriminator is to make a determination whether the input tensor is a real or imitative object. The discriminator uses the following inputs:
    a) historical customer selections
    b) next selection.

   The embeddings for historical selections are trainable; however one can use a pre-trained embeddings. The output of discriminator is a dense layer with one neuron with a sigmoidal activation function.
3) The intermediate layer is an additional level with Gaussian noise. Gaussian noise is required in order to help generator in producing realistic one-hot-encoded vector of user selection without the rounding operation. The rounding operation prevents the backpropagation, however without rounding the discriminator can easily distinct real objects from generated. The noise layer is intervened between the discriminator and generator. The shape of noise layer is equal to the number of user options to select. The Gaussian distribution density is the following:

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left(\frac{-1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{3}$$

where μ is a mean vector ($\mu \in R^n$), Σ is a covariance matrix ($\Sigma \in R^{n\times n}$). The noise level is ignored on stage of inference, it is required during the model training. As one can see from the mentioned above architecture description, for the training process the following data is required:

- the historical selections of the client;
- next selection of the client.

Thus, in the proposed architecture, we do not need to perform negative sampling to obtain positive and negative classes for binary classification. The generated objects will serve a purpose of negative objects for the discriminator training. A negative sampling causes a number of distortions in the final ratings of recommendations, since the model has tendency to use the frequency of item occurring in the training set as additional information. Due to the avoidance of negative sampling, the proposed approach is less prone to rating distortion.

5. Results

In order to compare the proposed approach with the current State-of-the-Art method, we trained the GAN model of the architecture described above on the flight dataset.

The data set contains flight information in 840 cities of 5.7 million anonymous customers.

A generator from a trained GAN was used to prepare recommendation ratings, and email offers with personalized recommendations were sent to customers in addition to newsletters.

The generator has encoder-decoder architecture: six sequent convolution layers of encoder using “relu” activation function with intermediate batch normalization, three convolution layers of decoder using “relu” activation function with intermediate batch normalization and upsampling layer.

The discriminator network was a recurrent type: we used three sequential LSTM layers and three fully connected dense layers, adding batch normalization and dropout after each layer.

In the role of competitor was Deep Semantic Similarity Model (DSSM) [13]:

- six convolution sequential layers with activation function “relu”, and max pooling layers in between,
- two fully connected dense layers with activation function “relu”, dropout and batch normalization,
- final fully connected dense layer with “sigmoid” activation function

The results of comparison between GAN and DSSM recommendation system is presented in table 1. A total of 162,978 clients were used to test the recommendations in the A / B test with the control group. In first group we used recommendations from DSSM, in the second group we used recommendations from GAN, in the baseline group we used the random direction. The conversion rate was the main metric for comparison, confidence interval was estimated using 95% significance level.

Table 1. The results of A/B testings of recommendations from DSSM and GAN.

	DSSM	**GAN**	**Baseline (random)**
Number of receivers	69688	59602	33688
Number of purchases	1627	1393	744
The rate of conversion	2.33 ± 0.11 %	**2.34 ± 0.12 %**	2.21 %

The GAN rate of conversion is higher than the DSSM, but the difference is not statistically significant. Thus, GAN can successfully compete with DSSM recommendation system in domain of airflight tickets.

As can be seen from the table 1, the conversion rate of DSSM and GAN recommendations is higher than the baseline, thus from both networks users received relevant offers. The tuning of DSSM network was based on the domain information, while GAN training wasn’t used additional domain information.

The proposed approach may encounter problems in cases where the client has a high number of choice options. We used simple architecture for GAN in the current case, however the improvement of recommendations based on GAN can be obtained using more complex loss functions, and more sophisticated network architecture, for example Transformer and additional domain information. The potential of StyleGAN for recommendation preparing require additional exploration and experiments.

6. Conclusion

The object of this paper is to consider possible application of GAN in the field of recommendation systems. The main novelty of the approach is to adapt the GAN to solve the practical problem of constructing recommendations, which is a step towards the main field of application of the GAN, such as image generation. Additional Gaussian Noise layer allow us to avoid the rounding operation, which prevents the back propagation. The advantage of a GAN-based recommendation system is that training does not require negative sampling. In addition, the GAN allows the researcher to obtain a rating distribution for each product for each item, which is very useful for understanding the confidence interval for rating recommendations. The recommender system based on the GAN was successively trained and showed results that are statistically close to the results of recommendations based on the DSSM neural network. Further study of GAN application is required on the field of recommender systems using the StyleGAN architecture, additional loss functions, and additional domain information.

References

[1] A Prosvetov, The comparison of autoencoder architectures in improving of prediction models, Journal of Physics: Conference Series, Volume 1117, conference 1 (2018)

[2] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua; Generative Adversarial Networks; Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680

[3] Ho, Jonathon; Ermon, Stefano; Generative Adversarial Imitation Learning; Advances in Neural Information Processing Systems; 2016; p. 4565–4573

[4] Horev, Rani; Style-based GANs – Generating and Tuning Realistic Artificial Faces; 2019; Lyrn.AI.

[5] Mustafa, Mustafa; Bard, Deborah; Bhimji, Wahid; Lukić, Zarija; Al-Rfou, Rami; Kratochvil, Jan M.; CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks; 2019; Computational Astrophysics and Cosmology Simulations, Data Analysis and Algorithms 6; 1

[6] Erdmann, Martin; Glombitza, Jonas; Quast, Thorben; Precise Simulation of Electromagnetic Calorimeter Showers Using a Wasserstein Generative Adversarial Network; 2019; Computing and Software for Big Science 3; 4

[7] Musella, Pasquale; Pandolfi, Francesco; Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks; Computing and Software for Big Science. 2018;2; 8.

[8] Homanga Bharadhwaj, Homin Park, and Brian Y. Lim; RecGAN: Recurrent Generative Adversarial Networks for Recommendation Systems; In Twelfth ACM Conference on Recommender Systems; 2018

[9] Prosvetov, A.V.; GAN for Recommendation System; J. Phys.: Conf. Ser. 1405 012005; 2019

[10] Zhang, Wei; Shift-invariant pattern recognition neural network and its optical architecture; Proceedings of Annual Conference of the Japan Society of Applied Physics; 1988

[11] S Hochreiter, J Schmidhuber; Long short-term memory; Neural computation 9.8; 1997; p. 1735-1780

[12] Mehdi Mirza, Simon Osindero; Conditional Generative Adversarial Nets; arXiv:1411.1784; 2014

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, Larry Heck; Learning Deep Structured Semantic Models for Web Search using Clickthrough Data; Proceedings of the 22nd ACM international conference on Conference on information & knowledge management; 2013