(IJRST) 2024, Vol. No. 14, Issue No. 1, Jan-Mar **e-ISSN: 2249-0604, p-ISSN: 2454-180X**

Enhanced Spark Cluster Recommendation Engine Powered by Generative AI

Tanvi S Hungund

Senior Manager, Dallas TX
California State University Fullerton

DOI:10.37648/ijrst.v14i01.004

[1]*Received: 02 December 2023; Accepted: 30 January 2024; Published: 13 February 2024*

ABSTRACT

Apache Spark, renowned for its proficiency in processing vast datasets, efficiently handles intricate processing tasks. It disperses these tasks across numerous computing instances autonomously or in conjunction with other distributed computing tools. As the volume of data burgeons and machine learning models advance, the imperative for swift and intricate feature engineering and model training intensifies. Clusters comprising multiple compute instances exhibit a noteworthy performance surge compared to individual cases, expediting data processing. However, leveraging such cluster configurations entails substantial costs due to the amalgamation of multiple compute instances (Worker Nodes) overseen by a Controller Node.

Nonetheless, significant cost reductions are observed as platforms like Databricks and EMR, which host these clusters, offer pay-as-you-use options. Furthermore, the cluster can be tailored to optimize performance while minimizing costs, although this manual process necessitates technical prowess and experience. This paper proposes the automation of cluster selection by developing a GEN-AI-powered recommendation engine. We demonstrate the utilization of Conditional Generative Adversarial Networks (cGANs) to generate additional samples from sparse custom training data. Leveraging data workload, anticipated usage duration, and budget constraints, the recommendation engine suggests controller and worker node instance types and the requisite number of worker nodes.

INTRODUCTION

Artificial Intelligence (AI) encompasses many technologies promising substantial enhancements in business value for organizations. With the exponential growth of data and computational capabilities, organizations increasingly rely on AI to drive business value.

The core objective of recommender systems is to furnish product recommendations to end-users based on their past actions or preferences. Despite significant research efforts to refine recommendation processes for precision and personalization, challenges such as cold start and data sparsity persist. Traditional recommendation engines rely on ratings and user preferences, but prioritizing user input is crucial for tailored recommendations that align with user needs. To craft purpose-driven recommendations, data-driven recommendation engines must surmount two vital technical hurdles: managing diverse data transformation file structures and addressing name inconsistencies in data transformation files.

While most established approaches define recommendation as regression or classification and deploy discriminative models, data sparsity poses a formidable challenge for recommender systems. Aside from exhibiting poor generalization, Discriminative models are prone to noise due to limited interactions. Generative Adversarial Networks (GANs) are increasingly employed in recommendation tasks to tackle data noise and sparsity issues. Recognized for their capacity to learn complex data distributions, GANs offer several advantages for recommendation tasks.

This study incorporates user and item subset interactions as conditional information, framing the generation of conditional rating vectors as a matching problem between users and items. This approach enables a more flexible

[1] How to cite the article: Hungund T.S. February 2024; Enhanced Spark Cluster Recommendation Engine Powered by Generative AI; *International Journal of Research in Science and Technology*, Vol 14, Issue 1, 26-32, DOI: http://doi.org/10.37648/ijrst.v14i01.004

model selection for the generator and discriminator, facilitating the estimation of generative models. Leveraging their ability to grasp intricate data distributions, GANs demonstrate promising outcomes across various domains, including computer vision and natural language processing.

GAN-based recommender systems can be categorized into two main types: one category reconstructs user-related vectors, while the other employs negative sampling using the generator.

Contribution of the Work:

- This work presents the pioneering Spark recommendation engine constructed utilizing conditional GAN.

- Custom data is generated during performance tuning.

- The authors have automated the cluster selection process by developing a recommendation engine based on GEN-AI. The recommendation engine suggests the instance type for master and worker nodes and the required number of worker nodes based on data workload, anticipated usage duration, and budget constraints.

PROBLEM FORMULATION

Apache Spark, an open-source framework for large-scale data analytics, employs in-memory computation to tackle iterative algorithms developed by the AMPLab at UC Berkeley. It offers a broader range of functionalities compared to MapReduce in Hadoop. In a cloud environment, selecting a cluster involves various considerations depending on the chosen criteria and cloud provider.

Defining needs: The initial steps involve outlining specific requirements, including workload types, application scalability, desired performance levels, and any software/hardware dependencies.

- Budget determination: Establishing a budget help narrow down cluster options and aligns them with financial constraints.

- Selecting cluster type: Cloud providers offer diverse cluster types, such as virtual machines, containers, or serverless computing options. Choosing the suitable type depends on job characteristics and required capabilities.

- Cluster size and configuration: Decisions regarding node/instance numbers, CPU/RAM capacity, storage, network bandwidth, and additional services/features are made.

- Fault tolerance and high availability: Evaluating providers' offerings to ensure continuous operation despite hardware failures or disruptions through features like automatic scaling and load balancing.

- Security and compliance: Assessing data encryption, network security, identity/access management, and compliance certifications to meet the firm's security and compliance standards.

- Performance and scalability: Verifying if the cluster can handle workload demands with appropriate scalability features like auto-scaling, vertical, and horizontal scaling.

- Cost optimization: Opting for instance types, storage options, and pricing models to optimize expenses, utilizing cost-saving measures like reserved or spot instances.

- Performance testing and benchmarks: Subject the cluster to performance tests and compare them against industry standards to ensure they meet requirements before deployment.

- Continuous monitoring and optimization: Utilizing monitoring tools to identify bottlenecks, maximize resource utilization, and make necessary adjustments to maintain performance while minimizing costs.

This process relies on manual execution and necessitates appropriate technical skills and expertise to set up a cluster effectively.

METHODOLOGY

Generative Adversarial Networks (GANs) are a generative modelling technique capable of learning deep representations even without heavily annotated training data. The core concept of GANs involves a competition between discriminators and generators in an adversarial process. The generator aims to produce samples that closely match the distribution of the training data, while the discriminator learns to distinguish between real and synthetic samples.

In the original GAN framework, the generator and discriminator are constructed as multilayer perceptrons, as depicted in Figure 1. The process commences by randomly selecting a fixed-length vector from a Gaussian distribution. This vector serves as a random seed for the generator, initiating the generation process in the latent space, where data distribution is projected.

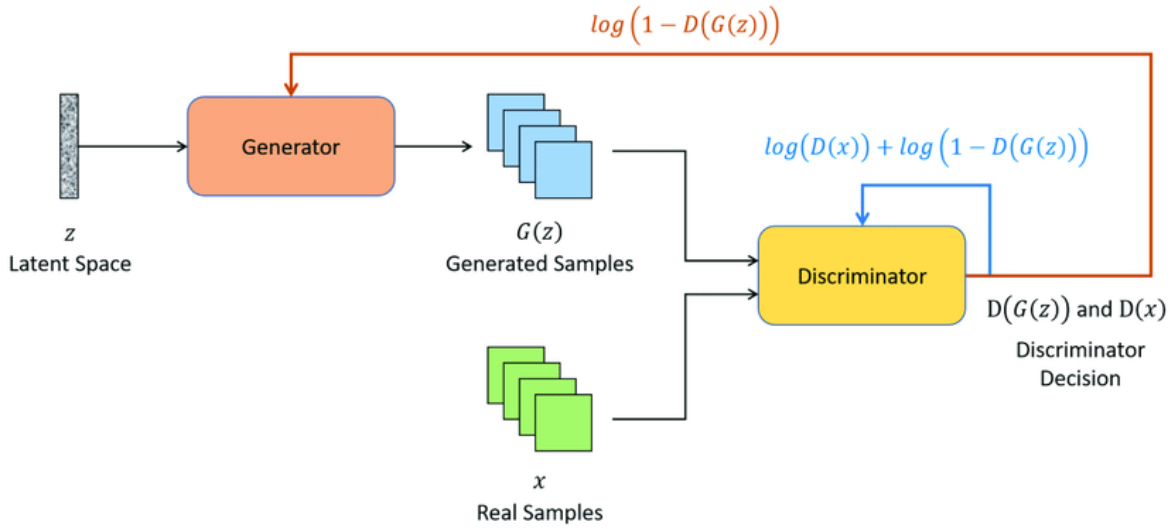

Fig. 1. Generative Adversarial Networks (GAN) architecture [14].

Through adversarial training, the generator and discriminator iteratively refine their abilities. The generator learns to produce outputs resembling accurate data, while the discriminator improves its discrimination skills. The minimax function describes the adversarial objectives of the functions G (generator) and D (discriminator), which adjusts each function's parameters. This continual refinement process facilitates the generation of synthetic data that closely mirrors accurate data distributions.

Equation (1) illustrates the objective of the discriminator in a GAN, which strives to minimize the output D(G(z)) towards 0, while the generator aims to maximize it, approaching 1. Here, $Pdata(x)$ represents the data distribution from which the GAN learns to generate samples. Input samples (z) from a prior distribution, $\rho z(z)$, are transformed into generated examples G(z) using a generator G. The discriminator D assesses the likelihood that a given example is real or fake by comparing actual samples from $Pdata(x)$ with those created by G.

Conditional GANs

Conditional GANs, a sophisticated class of machine learning models, merge the capabilities of GANs with the inclusion of conditional information. This unique feature empowers cGANs to produce highly realistic and contextually specific outputs based on provided conditions, sparking curiosity about their potential applications.

A GCN architecture comprises two main components: a generator G and a discriminator D. The generator takes random noise and conditional inputs as input and generates synthetic samples. In contrast, the discriminator distinguishes between genuine and generated samples.

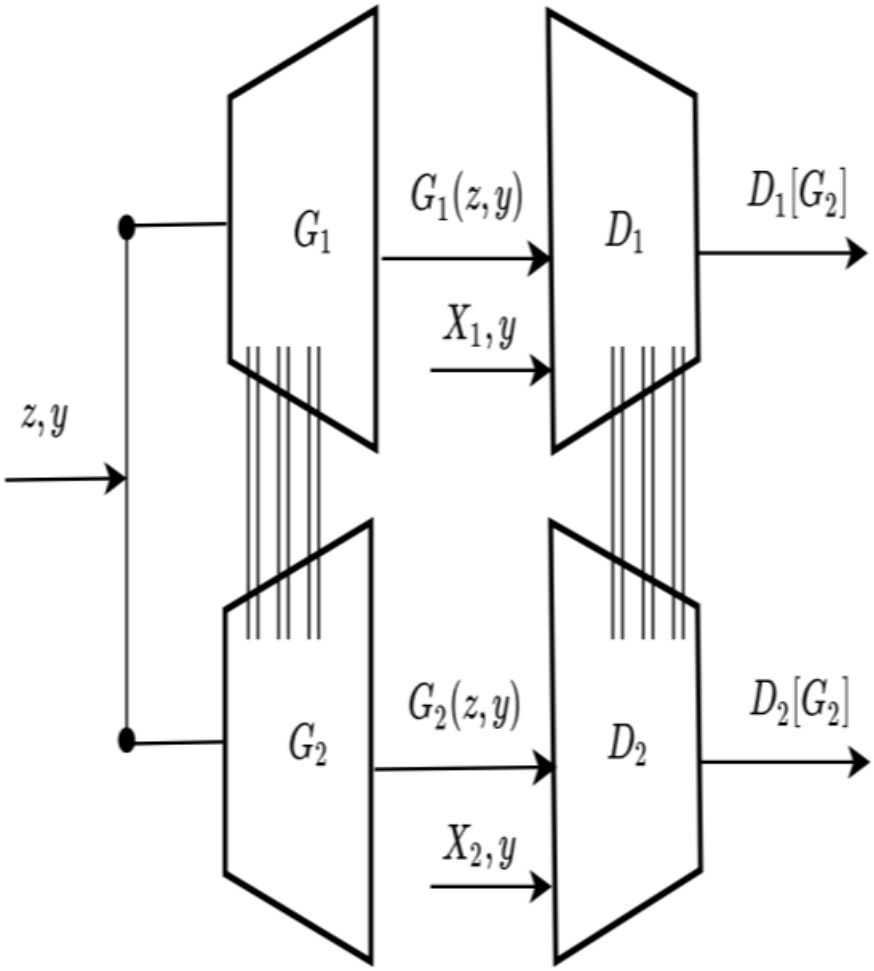

Fig. 2. conditional GAN architecture for recommendation

Incorporating additional information about the input data, the GAN model can consider an auxiliary input signal, y, supplied to both the generator and discriminator functions, enabling them to learn different aspects of the target distribution. Consequently, the objective function of the conditional GAN changes, as depicted in equation (2).

In this context, $Pdata(x)$ represents the joint density that the model is tasked with learning, with E representing the expectation operator.

The GAN model employed in the current study is schematically represented in Figure 2. It demonstrates how GAN-based recommender systems can handle complex data distributions and reconstruct user-related vectors. The latent variables z and user attributes y are input into the generators, yielding matrices G1 and G2. Discriminators are trained to distinguish between these generated matrices and real samples X1 and X2. Any errors in this discrimination process are utilized to adjust the weights in the generators.

In the realm of cluster recommendation, considering a set of overlapping cluster categories, the conversion rate (CR) is defined as the ratio of the number of recommended clusters that are selected to the total number of recommended items.

In this context, $Pdata(x)$ represents the joint density that the model is tasked with learning, with E representing the expectation operator.

Figure 2 schematically represents the GAN model employed in the current study. It demonstrates how GAN-based recommender systems can handle complex data distributions and reconstruct user-related vectors. The latent variables z and user attributes y are input into the generators, yielding matrices G1 and G2. Discriminators are trained to distinguish between these generated matrices and actual samples X1 and X2. Any errors in this discrimination process are utilized to adjust the weights in the generators.

In the realm of cluster recommendation, considering a set of overlapping cluster categories, the conversion rate (CR) is defined as the ratio of the number of recommended clusters that are selected to the total number of recommended items.

In equation (3), (iv, ib) denote items, and (CV, Cb) represent product categories. N signifies the number of GAN realizations, y stands for the user segment, and "#()" denotes the cardinality of the argument.

RESULTS AND DISCUSSION

This adversarial network is proficient in backpropagation updates and adversarial learning by generating user-related values for data augmentation. Custom data was created during performance tuning to train the model effectively. Processing vast amounts of data and training it using the Spark Framework posed the challenge of identifying an affordable yet highly efficient cluster configuration. Therefore, the authors devised permutations and combinations of different data sizes and cluster configurations to evaluate overall performance.

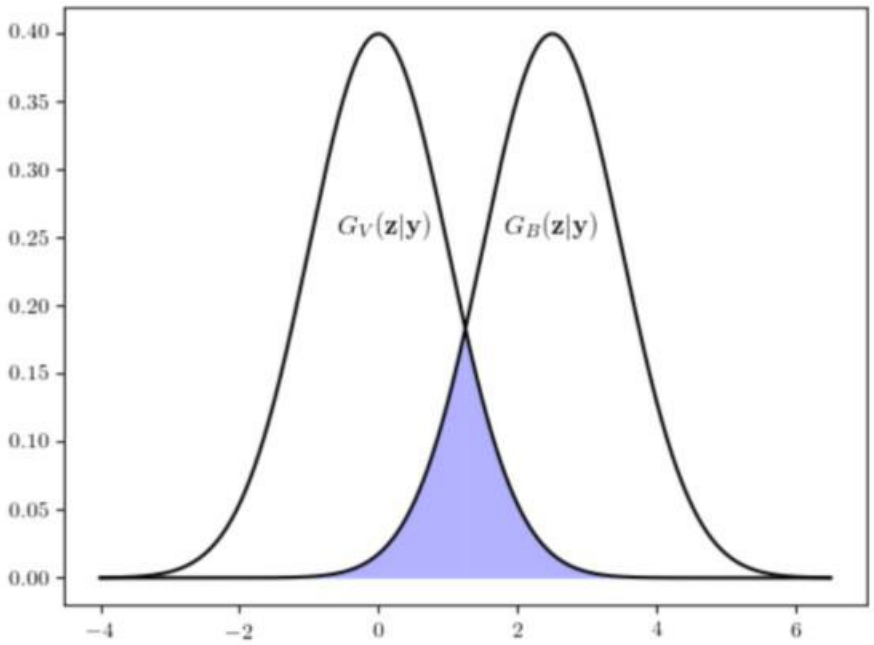

Fig 3. Recommended items are drawn from intersection of outputs generated by the trained cGAN.

During the rigorous model training, it was observed that the incorporation of dropout layers in G1 and G2 significantly facilitated convergence. However, the addition of these layers to the discriminator sub-models did not noticeably impact performance. The optimization method utilized during training was Adam, also known as stochastic gradient descent. Default values were applied to the customizable parameters, ensuring the model's effectiveness.

Over 300 epochs, the system underwent training on encoded data, utilizing randomly selected batches comprising 32 instances each. Throughout these training cycles, the training data statistics for each network constituting the model were closely monitored.

Until approximately epoch 300, when the GAN output began to diverge, G1, G2, and accurate statistics progressively converged. This convergence might indicate that the networks' capacity to encode information for this abstract learning task reached a plateau. However, the measured discriminator accuracies at training termination typically ranged between 45 and 55 per cent, indicating the models' inability to discern between genuine and fraudulent distributions generated by the generators.

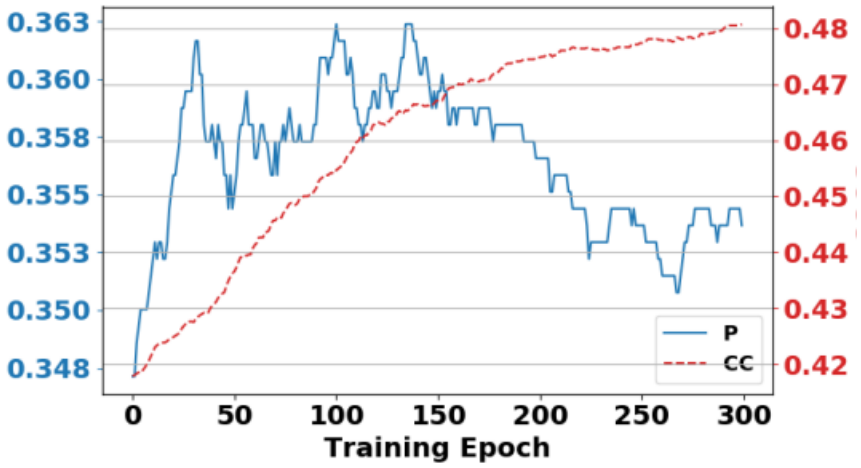

Fig 4. The learning curve of cGAN

Figure 4 illustrates an initial rise in relevance through adversarial training followed by a subsequent decline as diversity steadily increases with expanding training epochs. This trend suggests that relevance and diversity may be enhanced as the generated sets resemble the ground-truth sets more closely, reflecting the increasing capabilities of the generator and discriminator. However, once diversity reaches its peak potential at approximately 140 epochs, relevance deteriorates as diversity increases.

CONCLUSION

The findings of this study suggest that the model's recommendations could benefit customers and other users. By leveraging item pairings, the conditional GAN learns to generate samples from the joint distribution of user requirements, which can then be utilized to select suitable clusters for specific user categories. The technique for Spark cluster recommendation developed in this study introduces several novel features.

To further advance this research, the authors have planned numerous future projects. They aim to explore various hybridization strategies (weighted, mixed, etc.) to uncover additional opportunities for integrating methodologies and data sources, thereby enhancing the quality of recommendations. Additionally, they will focus on evaluating the impact of matrix factorization approaches on the accuracy and scalability of the recommendation system.

REFERENCES

1. J. Wen, B. Y. Chen, C. D. Wang, and Z. Tian, “PRGAN: Personalized Recommendation with Conditional Generative Adversarial Networks,” Proc. - IEEE Int. Conf. Data Mining, ICDM, vol. 2021-Decem, no. Icdm, pp. 729–738, 2021, https://doi.org/10.1109/ICDM51629.2021.00084
2. J. R. Bock and A. Maewal, “Adversarial Learning for Product Recommendation,” Ai, vol. 1, no. 3, pp. 376–388, 2020, https://doi.org/10.3390/ai1030025
3. A. Akbar, P. Agarwal, and A. J. Obaid, “Recommendation engines-neural embedding to graph-based: Techniques and evaluations,” Int. J. Nonlinear Anal. Appl, vol. 13, no. 1, pp. 2008–6822, 2022, [Online]. Available: http://dx.doi.org/10.22075/ijnaa.2022.5941
4. G. Zhu, J. Cao, C. Li, and Z. Wu, “A recommendation engine for travel products based on topic sequential patterns,” Multimed. Tools Appl., vol. 76, no. 16, pp. 17595–17612, 2017, https://doi.org/10.1007/s11042-017-4406-6
5. Q. Wang, Q. V. H. Nguyen, H. Yin, Z. Huang, H. Wang, and L. Cui, “Enhancing collaborative filtering with generative augmentation,” Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 548–556, 2019, https://doi.org/10.1145/3292500.3330873
6. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, 2005, https://doi.org/10.1109/TKDE.2005.99
7. S. Panigrahi, R. K. Lenka, and A. Stitipragyan, “A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark,” Procedia Comput. Sci., vol. 83, no. BigD2M, pp. 1000–1006, 2016, https://doi.org/10.1016/j.procs.2016.04.214
8. D. Liu, G. P. Farajalla, and A. Boulenger, “BRec the Bank: Context-aware Self-attentive Encoder for Banking Products Recommendation,” Proc. Int. Jt. Conf. Neural Networks, vol. 2022-July, pp. 1–8, 2022, https://doi.org/10.1109/IJCNN55064.2022.9892130

9. E. Lacić, D. Kowald, D. Parra, M. Kahr, and C. Trattner, "Towards a scalable social recommender engine for online marketplaces: The case of apache solr," WWW 2014 Companion - Proc. 23rd Int. Conf. World Wide Web, pp. 817–822, 2014, https://doi.org/10.1145/2567948.2579245
10. I. Goodfellow et al., "Generative adversarial networks," Commun. ACM, vol. 63, no. 11, pp. 139–144, 2020, https://doi.org/10.1145/3422622
11. Y. Zheng, Y. Zhang, and Z. Zheng, "Continuous Conditional Generative Adversarial Networks (cGAN) with Generator Regularization," no. 2017, 2021, [Online]. Available: http://arxiv.org/abs/2103.14884
12. C. Sun, H. Liu, M. Liu, Z. Ren, T. Gan, and L. Nie, "LarA: Attribute-to-feature adversarial learning for new-item recommendation," WSDM 2020 - Proc. 13th Int. Conf. Web Search Data Min., pp. 582–590, 2020, https://doi.org/10.1145/3336191.3371805
13. Q. Wu, Y. Liu, C. Miao, B. Zhao, Y. Zhao, and L.Guan, "PD-GAN: Adversarial learning for personalized diversity-promoting recommendation," IJCAI Int. Jt. Conf. Artif. Intell., vol. 2019- Augus, pp. 3870–3876, 2019, https://doi.org/10.24963/ijcai.2019/537
14. D. Vint, M. Anderson, Y. Yang, C. Ilioudis, G. Di Caterina, and C. Clemente, "Automatic target recognition for low resolution foliage penetrating SAR images using CNNS and GANS," Remote Sens., vol. 13, no. 4, pp. 1–18, 2021, https://doi.org/10.3390/rs13040596