

Measuring Feature Quality for Improved Ranking Performance^{*}

Sulagna Gope^{1,*}, Ravi Sugandharaju², Anup Kotalwar¹ and Vamsi Salaka²

¹Amazon Search, Bangalore, India

²Amazon Search, Palo Alto, USA

Abstract

Learning-to-rank models are mostly evaluated based on how good it is able to estimate the user behaviour. Output metrics like NDCG become the obvious choice for the purpose. A model is considered to have a good performance if it is able to predict the correct ranked ordering, else it is considered to be of poor quality. However the performance of a model is not only dependent on the prediction power of the model but also the quality of input features. Hence evaluation via output metrics like NDCG does not truly reflect the underlying problem. In this paper we introduce a simple feature coverage metric (FeCo) that can be used for tracking feature quality for diagnostic purpose as well as to get insight into model performance. Our experiments show that FeCo score is correlated with output metrics like NDCG. We also found that even a small change in FeCo score during training can have significant impact on the feature's contribution to the model. Our findings provide a perspective of having a 360 degree evaluation of model performance for ranking in production setup.

Keywords

learning-to-rank, IR metrics, explainability, feature importance

1. Introduction

Features play a crucial role in learning-to-rank models. The ranked list of items in e-commerce is not only dependent on the model's predictive power but also the quality of input features. In real world large scale e-commerce stores, ensuring a good data quality can become a major challenge. Real world data is often characterised by sparse features. Additionally, poor feature design may also lead to noise or sparsity that in turn affects ranking. During offline model training, we often sample a set of the data, post-process it to get a cleaner version that is finally used for training models. However at inference time, noisy or sparse features may lead to poor ranking. This gives rise to a gap between offline and online evaluation metrics. Moreover, tracking feature quality can be useful for diagnostic purpose and to understand overall trends.

In order to have a full 360 degree view of model performance, we propose a feature evaluation metric FeCo that evaluates the coverage of ranking features. To the best of our knowledge currently there does not exist any such metric that measures the quality of ranking features. Most of the existing metrics in the field of information retrieval (IR) evaluates a model based on


eCom'23: ACM SIGIR Workshop on eCommerce, July 27, 2023, Taipei, Taiwan

*Corresponding author.

✉ gope@amazon.com (S. Gope); ravics@amazon.com (R. Sugandharaju); kanup@amazon.com (A. Kotalwar); vsalaka@amazon.com (V. Salaka)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

the output produced. This does not give a full insight into the model. Essentially the ranking performance may still remain poor if model complexity is increased without focusing on the quality of input metrics. To validate the utility of the new metric, we conducted experiments on in-house Amazon product search dataset as well as one of the publicly available dataset. Our experiments show a high correlation between FeCo and output metrics like NDCG. Furthermore, we show that the metric can give more insight into the nature of the data. In this study we have also explored the impact of FeCo change on model explainability. Though this study has been performed for learning-to-rank models, this can also be extended to any other machine learning models in large scale production setup.

2. Related Work

Learning-to-rank models are generally evaluated using output evaluation metrics which are based on user interactions. Some of the common metrics include Discounted Cumulative Gain (DCG) and Normalised DCG [1, 2], Rank-based precision [3], Expected reciprocal rank [4], Expected browsing utility [5], Time-based gain [6], U-measure [7], INSQ [8], INST [9] and so on. Recently user model based evaluations have been extended to session based evaluations [10, 11]. Ranking models essentially try to simulate the user behaviour under operational conditions. There has been some effort to understand and characterize user behaviour models [12, 9]. [13] introduced an anchor-aware evaluation metric where user's bias towards initial values or starting points is taken into consideration for evaluation of IR systems.

On the other hand, recently there has been a lot of effort on model explainability to get better insight into machine learning based retrieval models. Estimating feature importances in model training using SHAP values, has been widely used for model explanation [14]. For latent factor models, explainability has been attempted by aligning each latent factor with an explicit meaning such as item features [15, 16, 17]. Recently many neural algorithms have been developed with explainable recommendations. [18] proposed to explain recommender models by highlighting important words in user reviews. In [19, 20] model explanation is achieved by ranking user review sentences. [21, 22] proposed a visual recommendation approach by highlighting the important regions in the image. It is evident from the existing literature that input features play an important role in model performance. However, quality of input features are rarely considered while evaluating the learning models. Our work is novel in the sense that we give a simple measure of feature quality that can be indicative of the output performance of the model and affect the internal working of the model.

3. FeCo Design

FeCo metric computation involves three main steps. In the first step, we sample data instances from past user logs for the required period of time. This is followed by feature extraction step and finally the computation of the metric. In the following subsections, we provide the details of each of the steps.

3.1. Data Sampling

Since real world e-commerce applications may witness a high amount of traffic daily, we propose to sample a smaller set of events from the actual user logs on which we measure FeCo. Some of the popular sampling techniques are 1) random sampling 2) stratified and cluster sampling and 3) systematic sampling. Reservoir sampling is another popular sampling technique mostly applicable in case of sampling from streaming data. Though our use case also involves streaming data, we perform the metric computation in offline setup rather than in real time. Our sampling approach is close to systematic sampling. Since the traffic varies across a day in e-commerce sites, we propose to pick a few time windows across a day and sample a fixed set of instances from each window. We compared different strategies by varying the number of windows in a day and number of samples from each window. We set the window size to be of 10 minutes duration. Based on our study we found that 8 uniformly distributed windows with 200K samples from each window to be a close approximation to the actual traffic distribution. We used this strategy to sample data points from daily logs for FeCo computation. Note that this configuration can be varied based on the nature of different e-commerce stores.

3.2. FeCo Computation

In theory, FeCo can be computed on any input feature to a ranking model. However to have more meaningful insights into ranking performance, we select a few features for FeCo computation. We prefer to compute FeCo for features which are important contributors to ranking models and are user-centric in nature. Feature coverage or FeCo measures the percentage of events where the feature has non-zero value compared to all the events. Here an event simply refers to a search event where a user visits an e-commerce site, performs some keyword search and gets a list of ranked products. Such events are generally logged in the search backend for purpose of future model training. We extract the required feature values for the sampled events and compute FeCo for the same using equation 1.

$$FeCo_f = \frac{\sum_{i=0}^S \mathbb{I}_f(event)}{\sum_{i=0}^S event} \quad (1)$$

where

$$\mathbb{I}_f(event) = \begin{cases} 1 & \text{if } f(event) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In equation 1 and 2, $FeCo_f$, stands for FeCo score for feature f , S is the number of events sampled. Here we have considered the threshold for coverage to be greater than 0. Based on the nature of the feature, one may also consider the threshold to be the mean or median of the feature values. FeCo can be measured as a percentage by multiplying the score with 100.

Since FeCo is used for tracking and measuring feature quality health, it is important to ensure that it is stable and movable. Based on our study on in-house Amazon dataset, we observed that the FeCo score on user engagement based signals show considerable variation between days. In order to get a smooth trend, we compute a moving average of the score over a 7 days window.

Table 1
Impact of FeCo drop on NDCG

Feature	Drop in Coverage	NDCG@4 (%drop)	NDCG@8 (%drop)
full data		0.569	0.572
cols-15	5% in test data	0.553 (2.81%)	0.559 (2.27%)
cols-15	25% in test data	0.553 (2.81%)	0.559 (2.27%)
cols-109	5% in test data	0.551 (3.16%)	0.561 (1.92%)
cols-109	25% in test data	0.550 (3.34%)	0.558 (2.45%)
cols-15	5% in train and test data	0.561 (1.4%)	0.567 (0.87%)
cols-15	25% in train and test data	0.560 (1.58%)	0.568 (0.69%)

4. Experiments

In this section we describe the datasets and the experiment performed with the proposed metric.

4.1. Dataset

We have performed our study on a sample of Amazon search data, comprising of query-product pairs which were anonymized and post-processed to remove user specific information. For computing FeCo, we extracted the query and product based features that are used in product search ranking in Amazon. Additionally we have performed our study on the publicly available MSLR-WEB10K [23] dataset. The study on the public dataset is done for purpose of research, to show the generalization power and wider applicability of the proposed metric.

4.2. Relation between FeCo and NDCG metrics

In this experiment we tried to study the impact of FeCo on output NDCG metric. We performed the experiment on public dataset WEB10K, and an in-house Amazon product search ranking dataset. We trained a LightGBM model for ranking for both the datasets. In the first set, we trained and evaluated the model using the standard available dataset, as our baseline. We then reduced the coverage of a few top features in the test set, by setting their value to 0. We then reduced the feature coverage in both the train and test set, retrained our model and computed the NDCG metric. In all these cases, we used the same set of parameters for model training and test. We have only reported NDCG@4 and NDCG@8 for our experiments. We varied the FeCo score and reported the impact on NDCG in the Table 1 for Web10K dataset.

It is to be noted that the impact of FeCO on output metrics like NDCG also depends on the type of model. For example some of the machine learning models like decision trees are more robust to noisy data than other other models. More robust the model is to noisy data, lesser will be the impact of FeCo score on output metrics. In this study we only present the results for tree based LightGBM model[24]. Comparative study of the impact of FeCO on other types of models is left as part of future scope.

In Table 2 we show the impact of coverage drop on NDCG for an internal Amazon search dataset.

Table 2
Impact of FeCo drop on NDCG on Amazon data

Feature	Drop in Coverage	NDCG@4 (%drop)	NDCG@8 (%drop)
full data		0.717	0.759
top-1 feature	25% in test data	0.701 (2.23%)	0.741 (2.37%)
top-1 feature	50% in test data	0.682 (4.88%)	0.723 (4.74%)
top random feature	25% in test data	0.688 (4.04%)	0.734 (3.29%)
top random feature	50% in test data	0.642 (10.46%)	0.693 (8.69%)

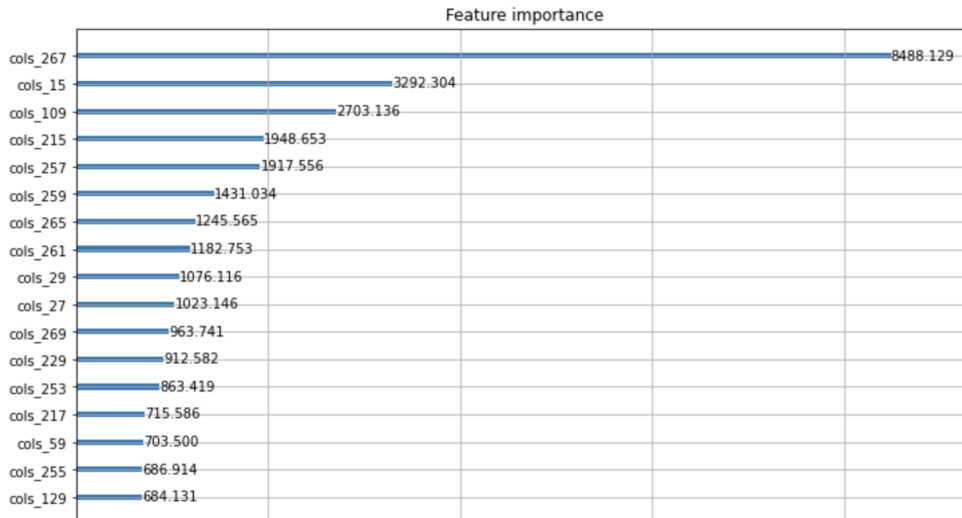


Figure 1: Feature importance curve of LightGBM model trained on actual Web10K dataset

4.3. Model Explainability with FeCo

Model explainability is often characterised by feature importance in trained model. In this experiment we studied the effect of coverage on feature importance. As a baseline, we first trained a LightGBM model on full training data. We then decreased the coverage of some of the top important features and studied the effect on the feature importance.

In Figure 1 we show the baseline feature importance for the top few important features as obtained from the LightGBM model trained on the actual dataset. We have used the standard LightGBM library to compute the feature importances. We then decreased the feature coverage of the top few features by 5 and observed that the importance of that feature drops significantly. Moreover, the importances of the other features also change, indicating that the model undergoes considerable changes by even a minor drop in feature coverage. In Figure 2 we show the feature importance curve for Web10K dataset where the coverage of feature *cols_267* is decreased by 5%.

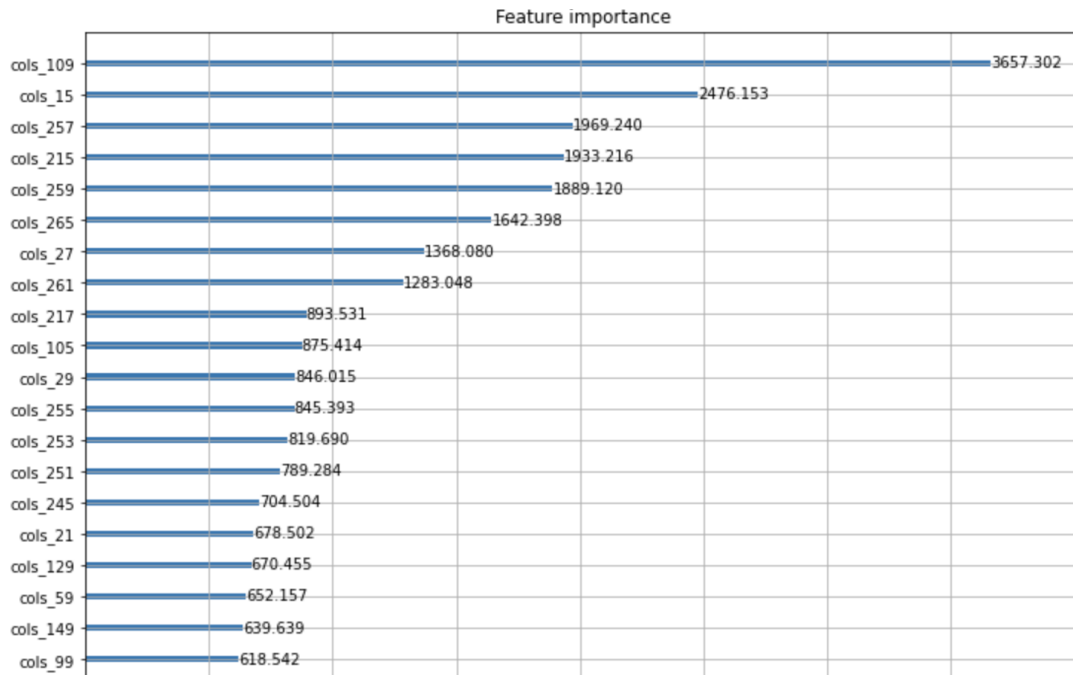


Figure 2: Feature importance curve of LightGBM model after 5% drop in coverage of cols_267 Web10K dataset

We performed similar study with respect to the ranking model that powers product search in Amazon. Decrease in FeCo leads to lesser contribution of the feature during model training. We hence infer that a high FeCo score of all the ranking features helps the model to pick up the right set of features and eventually improve ranking performance. In our study we also found that FeCo drop in more important features has more impact on output NDCG compared to FeCo drop in less important features. However whether there is a direct relationship between feature importance and impact of FeCo drop requires more detailed analysis as feature importance depends on other features and the prediction model. We leave this as a scope for our next study.

5. Usage of FeCo

5.1. Tracking Cold Start with FeCo

It has been found in practice, that ranking models perform better when they leverage past customer engagement data for query-item pairs, besides the item metadata features. However past engagement signals when used in ranking models, give rise to an inherent bias where more a product is clicked in the past, the more it has a chance of being ranked higher up, eventually giving rise to cold start problem for new products with lesser customer engagement. A direct comparison of FeCo on engagement based ranking features for cold start products and older relevant products would help in understanding the cold start impact. Tracking FeCO over a

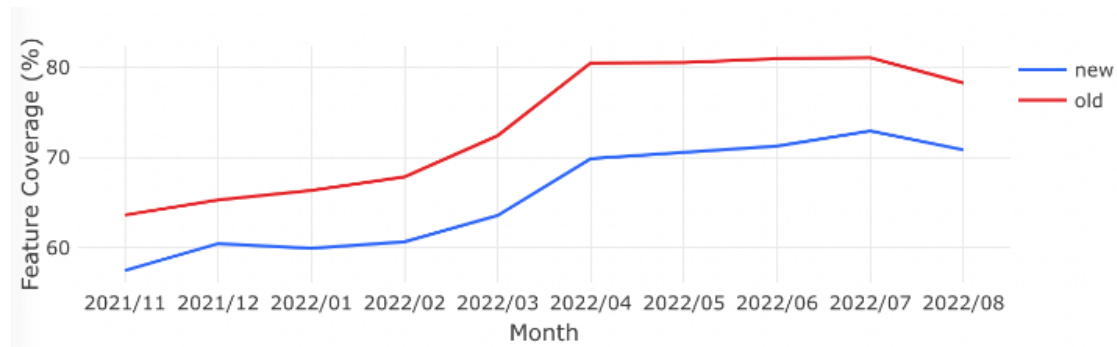


Figure 3: Comparison of FeCo on behavioural features for Old vs New Product

period of time, can help in understanding the growth of engagement based features coverage for new ASINs. For large scale e-commerce search engines FeCo can quantify the magnitude of cold start problem and also track the reduction in cold start over time. In the figure 3 we show a comparison between FeCo on behavioural features for some old versus new products in Amazon, tracked over a period of time.

5.2. Tracking customer behaviour OR overall ranking improvements in newly launched locations

E-commerce systems also suffer from a generic cold start problem when launched in new locations. Since towards the beginning of a launch the customer base is small, it does not give rise to significant engagements that can potentially improve ranking. Due to this ranking models often rely on metadata features only for such scenarios. Over time the engagement based features start getting more coverage and tends to contribute more towards ranking. FeCo can be used in such case to track the feature dynamics in newly launched locations.

In general, FeCo can be used for monitoring feature health in large scale applications. It is also insightful to study FeCo across different segments like product groups, head/torso/tail queries, regions, etc.

6. Conclusion

In this paper we explored the feature coverage metric aka FeCo for evaluation of ranking features in production e-commerce applications. Through this study we show that this simple metric has a strong correlation with output metrics and model explainability through feature importance. In the future scope we would like to attempt to unify FeCo with output metrics like NDCG to have a single metric for overall model evaluation. We would further like to study the impact of FeCo on output for different types of models and also for applications beyond ranking.

References

- [1] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, *ACM Transactions on Information Systems (TOIS)* 20 (2002) 422–446.
- [2] K. Järvelin, S. L. Price, L. M. Delcambre, M. L. Nielsen (Eds.), Discounted cumulated gain based evaluation of multiple-query IR sessions, 4–15, Springer, 2008.
- [3] A. Moffat, J. Zobel, Rank-biased precision for measurement of retrieval effectiveness, *ACM Transactions on Information Systems (TOIS)* 27 (2008) 1–27.
- [4] O. Chapelle, D. Metzler, Y. Zhang, P. Grinspan (Eds.), Expected reciprocal rank for graded relevance, 621–630, 2009.
- [5] E. Yilmaz, M. Shokouhi, N. Craswell, S. Robertson (Eds.), Expected browsing utility for web search evaluation, 1561–1564, 2010.
- [6] M. D. Smucker, C. L. Clarke (Eds.), Time-based calibration of effectiveness measures, 95–104, 2012.
- [7] T. Sakai, Z. Dou (Eds.), Summaries, ranked retrieval and sessions: A unified framework for information access evaluation, 473–482, 2013.
- [8] A. Moffat, P. Thomas, F. Scholer (Eds.), Users versus models: What observation tells us about effectiveness metrics, 659–668, 2013.
- [9] A. Moffat, P. Bailey, F. Scholer, P. Thomas (Eds.), INST: An adaptive metric for information retrieval evaluation, 1–4, 2015.
- [10] A. F. Wicaksono, A. Moffat, Modeling search and session effectiveness, *Information Processing & Management* 58 (2021) 102601.
- [11] A. Lipani, B. Carterette, E. Yilmaz (Eds.), From a user model for query sessions to session rank biased precision (sRBP), 109–116, 2019.
- [12] B. Carterette (Ed.), System effectiveness, user models, and user utility: a conceptual framework for investigation, 903–912, 2011.
- [13] N. Chen, F. Zhang, T. Sakai (Eds.), Constructing Better Evaluation Metrics by Incorporating the Anchoring Effect into the User Model, 2709–2714, 2022.
- [14] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, S. Friedler (Eds.), Problems with Shapley-value-based explanations as feature importance measures, 5491–5500, 2020.
- [15] X. Chen, Z. Qin, Y. Zhang, T. Xu (Eds.), Learning to rank features for recommendation over multiple categories, 305–314, 2016.
- [16] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma (Eds.), Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, 83–92, 2014.
- [17] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, S. Ma (Eds.), Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification, 1027–1030, 2014.
- [18] S. Seo, J. Huang, H. Yang, Y. Liu (Eds.), Interpretable convolutional neural networks with dual local and global attention for review rating prediction, 297–305, 2017.
- [19] X. Chen, Y. Zhang, Z. Qin (Eds.), Dynamic explainable recommendation based on neural attentive models, volume 33, 2019.
- [20] L. Li, Y. Zhang, L. Chen (Eds.), Extra: Explanation ranking datasets for explainable recommendation, 2463–2469, 2021.
- [21] X. Chen, H. Chen, H. Xu, Y. Zhang, Y. Cao, Z. Qin, H. Zha (Eds.), Personalized fashion rec-

ommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation, 765–774, 2019.

- [22] S. Geng, Z. Fu, Y. Ge, L. Li, G. de Melo, Y. Zhang (Eds.), Improving Personalized Explanation Generation through Visualization, 244–255, 2022.
- [23] T. Qin, T. Liu, Introducing LETOR 4.0 datasets, CoRR abs/1306.2597 (2013).
- [24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, *Advances in neural information processing systems* 30 (2017).