Predicting Online Performance of Job Recommender Systems With Offline Evaluation

Adrien Mogenet, Tuan Anh Nguyen Pham, Masahiro Kazama, Jialin Kong
{amogenet,tanh,mkazama,jkong}@indeed.com
Indeed
Tokyo, Japan

ABSTRACT

At Indeed, recommender systems are used to recommend jobs. In this context, implicit and explicit feedback signals we can collect are rare events, making the task of evaluation more complex. Online evaluation (A/B testing) is usually the most reliable way to measure the results from our experiments, but it is a slow process. In contrast, the offline evaluation process is faster, but it is critical to make it reliable as it informs our decision to roll out new improvements in production. In this paper, we review the comparative offline and online performances of three recommendations models, we describe the evaluation metrics we use and analyze how the offline performance metrics correlate with online metrics to understand how an offline evaluation process can be leveraged to inform the decisions.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

evaluation; statistical analysis; comparative studies; accuracy metrics

ACM Reference Format:
Adrien Mogenet, Tuan-Anh Nguyen Pham, Masahiro Kazama, Jialin Kong. 2019. Predicting Online Performance of Job Recommender Systems With Offline Evaluation. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3298689.3347032

1 INTRODUCTION

Offline evaluations and offline metrics have been frequently adopted in industry as the main method to measure the progress, the incremental value of additional features, or to compare several algorithms. This is an efficient process as it helps model developers get an immediate feedback for their decision to run the model in production or not. Without an offline evaluation process, they have to release every change in production, presumably using an A/B testing platform, to evaluate the actual improvement their new model can deliver. This is especially tedious in the context of parameter tuning, where the number of options can grow exponentially. Our work answers the following questions in the context of job recommendation:

- What are the offline evaluation metrics we should monitor to expect an impact in production?
- What is the level of confidence we can have in the offline results?
- How should we decide to push or not push a new model to production?

This complements the work from Yi et al. [12] which analyzes the problem in the context of CTR prediction and Maksai et al. [7] for news recommendation.

1.1 The problem of job recommendation

A constraint specific to job recommendations is that they come in very limited supply, since the vast majority of the openings (also later referred as "entities") will accept only one job seeker (also later referred as "user"). In addition, the problem is tripartite, involving entities (jobs), users (job seekers), and suppliers (employers providing jobs). Here is the typical conversion funnel:

$$Impression \rightarrow Click \rightarrow Apply \rightarrow Interview \rightarrow Get\ a\ Job$$

What happens after the application mostly depends on the last part of the job recommendation problem: the employer. In job recommendation, we cannot provide the resource to everyone, as the employer can decide to accept or decline an application. This contrasts with traditional recommender systems, such as movie or product recommender systems, where there is no reason to not display a movie, or to not sell an item to a user. As a result, "Popular Recommender", often considered as a strong baseline [3], cannot be used for job recommendation.

2 OPTIMIZING TASK

The funnel presented above raises a question of the right online metric that should be used to measure the value of job recommendation models. We want every user to acquire the resource they are asking for. Unfortunately, this is a very sparse signal, and we decide to model the job recommendation problem with only the first half of the funnel:

$$Impression \rightarrow Click \rightarrow Apply$$

Although we have mechanisms to improve the second half of the funnel in production and help people get hired, our recommendation models are optimized for applies. To define our session success rate metric, therefore, we will further refer to *apply-rate*, and more

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347032

precisely *apply-rate at rank 10*, as defined by:

$$apply{-}rate@10 = \frac{\#\ applies\ up\ to\ rank\ 10}{\#\ impressions\ up\ to\ rank\ 10}$$

This is more relevant than the global rate, as we return a potentially long list of recommendations but want to maximize the value of the first result page, showing 10 results. We intentionally do not measure other sets of metrics, such as serendipity as recommended by McNee et al. [9].

3 COLLECTED FEEDBACK

In this work, we consider *applies* as an explicit positive feedback. We use *clicks* as signals in most of our models as an implicit feedback with lower weight. We do not collect explicit ratings which could help rank user preferences as we do not provide such options to our users (e.g. a five star rating system), thus two jobs which result in applies are considered equally good, and two jobs with no apply are viewed as equally bad during the evaluation.

4 RECOMMENDATION MODELS

In this experiment, the three following models are used:

- `word2vec` (w2v), an embedding model as presented by Grbovic, M. and Cheng, H. [4], where the model captures the sequence of actions
- `word2vecHS` (w2vhs), a variant of `word2vec` using hierarchical softmax[4]
- `knn`, as an item-based collaborative filtering technique

These models return personalized recommendations: a set of jobs is returned to the users based on their past interactions. The following sections of this paper focus on the correlation between the measured offline performance and the actual performance in production. Hence we will no analyze the comparative performance of these models.

5 DATASET

The dataset we use for our offline evaluation is a sample from a snapshot taken from 21 days of anonymized interactions between jobs and job seekers, where each row follows this schema:

`<user>;<job>;time;clicked;applied`

Where `<user>` and `<job>` fields are identifiers and features available at the moment we generate the recommendations, i.e. they do not contain any *a posteriori* information, and `clicked` and `applied` are boolean values. The dataset contains on average 125M events sampled from our 250M monthly users and a total of 20M jobs. We use 5% of those data to generate the testset and perform the offline evaluation.

6 EVALUATION

6.1 Evaluation Metrics

In this work, we evaluate the correlation of online metrics with various offline metrics commonly used to evaluate the quality of a ranking [5, 8]: *MAP*, *MPR*, *precision@k* (later referred as *p@k*), *NDCG@k*, *recall@k*, for k in {3, 10, 20, 30, 40}. These metrics are often preferred over probability-based (AUC, MLE, etc.) or log-likelihood metrics (RIG, cross-entropy, etc.) in the context of search or recommendation [12]. We do not use error-based metrics such as *MAE* and *RMSE* since they are not appropriate to evaluate recommender systems in the top-N recommendation task [2, 11]. We intentionally omit novelty and serendipity metrics as covered by Maksai et al. [7] since we only optimize for *apply-rate*.

6.2 Evaluation Methodology

During two weeks, we run an A/B test with one bucket for each model described in section 4. Daily, we generate new recommendations based on the past data and compare the performance in production (*apply-rate*) with the offline performance (*p@k*, etc.). That way, we can observe how the models evolve online and offline, and reliably compute the correlations between the different metrics, while avoiding stale states as much as we can (for instance, a changing behavior, distribution of features in our jobs or users could affect the online performance if the recommendation model is not retrained). More than 90% of the traffic is still served by a different baseline running in production. Therefore the dataset is continuously populated with data from independent models.

6.3 Correlation metrics

Since the goal is to better understand the type and degree of correlation, we use two different correlation measurements. First, we use product-moment Pearson correlation which is defined as

$$\rho = \frac{\mathrm{cov}(X, Y)}{\sigma_x \sigma_y} \tag{1}$$

where X and Y are random variables representing offline and online metrics, and σ_x and σ_y are their standard deviations, respectively. As we are mostly interested in whether progress observed from offline evaluation leads to online improvement, we also compute the Spearman ρ_s coefficient as a rank-based version of the Pearson correlation:

$$\rho_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2}$$

where $d_i = rg(X_i) - rg(Y_j)$ is the pairwise distance between the two ranks of each observation and n the number of observations. The Spearman coefficient describes the monotone relationship between two variables, and is better to use when this is a non-linear relationship. We compute both Pearson and Spearman coefficients with their associated estimates. These coefficients are used only to measure the correlation between the offline and online metrics; they are not used for the offline evaluation as presented by Herlocker et al.[6]. In the following sections, we consider results reaching statistical significance with $p < 0.05$.

6.4 Interpretation

For both Eqs. (1) and (2), the sign of the coefficient shows the direction of the association between the two variables. The absolute value ranges from 0 to 1 and indicates the degree of relationship between the two variables. A value close to 0 indicates no relationship. As the absolute value of the coefficient gets closer to 1, it means X and Y become perfect monotone (Spearman) or linear (Pearson) function of each other. In accordance with MM Mukaka [10], we consider the following interpretation of the magnitudes:

- $[0.00 - 0.30]$: negligible

Table 1: Cross-model comparison, averaged over 21 days; word2vec is used as baseline. The metrics in bold do not have the expected sign, e.g. online performance increased, but offline evaluation metric decreased.

Model	apply-rate	apply-rate@10	p@10	MAP	MPR	NDCG@10	recall@10	recall@100
w2v	-	-	-	-	-	-	-	-
knn	+17%	+11%	+9.3%	**-54%**	**+11%**	**-47%**	**-38%**	+3.2%
w2vhs	+48%	+46%	+90%	+51%	-5.1%	+60%	+70%	+65%

Table 2: Per-model Pearson coefficients. Non statistically significant values have been removed. ρ_{METRIC} denotes the pearson correlation for a named $METRIC$.

Model	ρ_{MAP}	$\rho_{NDCG@10}$	$\rho_{recall@10}$	$\rho_{P@10}$
knn	0.692	0.692	0.696	0.619
w2vhs	0.768	0.772	0.608	0.755
w2v	0.673	0.673	0.664	0.629

Table 3: Correlations of MPR and MAP with *apply-rate@10*

Metric	Pearson	Spearman
MPR	-0.5434	-0.5785
MAP	0.7615	0.7849

- $[0.30 - 0.50]$: low correlation
- $[0.50 - 0.70]$: moderate correlation
- $[0.70 - 0.90]$: high correlation
- $[0.90 - 1.00]$: very high correlation

As per the definition of the offline metrics we choose, we expect *MPR* to be negatively correlated (the lower, the better) while we expect *MAP*, *p@k*, *recall@k* and *NDCG@k* to be positively correlated (the higher, the better) to *apply-rate@k*.

6.5 Dimensions of analysis

We analyze the correlations following three angles:

- **across models**, to understand if looking at averaged offline evaluation metric is sufficient to predict the online performance.
- **per model**, to understand if, over time and for a given model, the offline evaluation metrics correlate with the actual online performance.
- **per metric**, to understand if, across all the models, a specific set of metrics stands out and allows to rank the recommendation models. We need to assume that if the exhibited online and offline performances were correlated in the past, they will remain correlated in the future.

In addition, we also add metrics we expect to be *not* correlated, such as the number of words in the job title. This is a sanity check to understand if the observed correlations are pure chance.

7 RESULTS

7.1 Comparative performance

To get an overview, we build the table 1 to compare the performances of the three models over the course of 21 days, and compute the average for each metric, as we usually do to conclude on an A/B test. We compute the correlations for the different values of k but report the most notable values. The metrics in bold are the ones which do not have the expected sign, i.e. the online metric showed an improvement (e.g. increase of *apply-rate@10*, but the offline metric did not, e.g. *MPR* increased). From the table 1, it is hard to visually identify a solid correlation between the sets of metrics. For instance, the offline results suggest to stop running *knn*, although it performs better than the baseline. Also, these results show that the magnitude of the changes can be large.

7.2 Per-model correlation

Since the table 1 exhibits inconsistent behaviors when comparing offline metrics across different models, we also compute the Pearson correlation coefficients, per model, over the series of 21 data points (1 per day). In Eq. (1), $X = \{x_1, x_2, ..., x_{21}\}$ denotes an online metric, here *apply-rate@10* while $Y = \{y_1, y_2, ..., y_{21}\}$ denotes an offline metric. The table 2 references the correlations. The results show a moderate to high degree of correlation between *apply-rate@10* and *MAP*, *NDCG@10*, and *p@10*. During this analysis, we also observe that the global *apply-rate* is slightly less correlated than *apply-rate@10* and that the Spearman coefficient shows a similar degree of correlation.

7.3 Per-metric correlation

While the table 1 shows the offline metrics do not always translate to online improvement, but table 2 shows some high degrees of correlation at the model level, we want to determine how those metrics correlate overall by combining all the accumulated data points from the three models running for 21 days. Figure 2 illustrates the relationship between *apply-rate@10* and *p@10* for different recommendation results across all the models, from which we can see the correlation between the two metrics, without being apparently affected by the Simpson's Paradox[1], i.e. "time" and "model" are not seen as confounding variables. For the quantitative assessment, for all the offline metrics we identified, we compute both Pearson and Spearman coefficients and report the values for *NDCG@k*, *recall@k* and *p@k* in tables 1 and 3, for $k \in \{3, 10, 20, 30, 40\}$. We also report the Pearson and Spearman correlation coefficients for *MPR* and *MAP* with *apply-rate@10* in Table 3. All the reported correlations reach $p < 0.005$. Overall, the results suggest that *recall@k* (ρ=0.83, ρ_s=0.85, for k=30, $p < 0.0001$) and *p@k* (ρ=0.84, ρ_s=0.88 for k=20, $p < 0.0001$) are the best predictors of the online performance.

8 CONCLUSION

In this paper, we reviewed the correlation levels of various offline metrics for job recommendation and showed these metrics can be

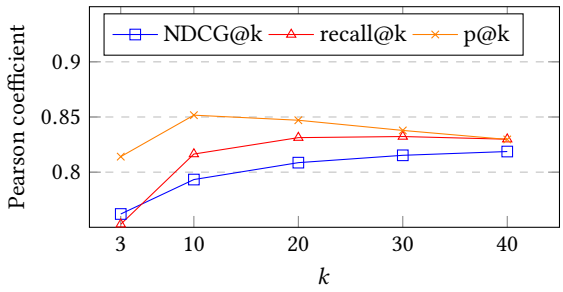

Figure 1: Pearson correlation with *apply-rate*@10

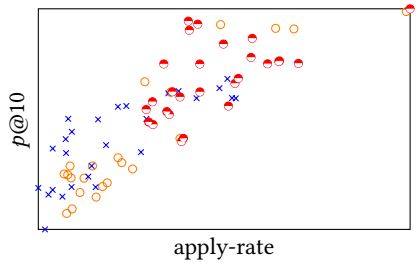

Figure 2: Visual relationship of *apply-rate@10* and *p@10*

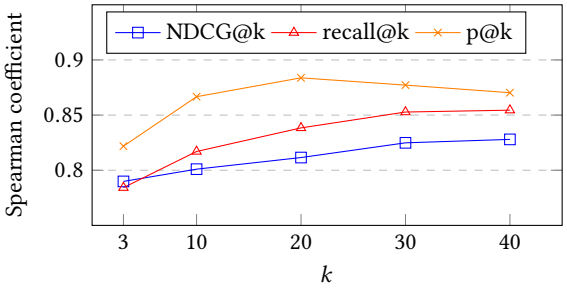

Figure 3: Spearman correlation with *apply-rate*@10

considered as highly correlated ($\rho \in [70–90]$) with online metrics. We conclude those offline evaluation metrics are reliable enough to decide to not deploy the new models when the offline performances are significantly negative; and to deploy the new models when there is a positive impact on the offline metrics. We recommend *p@k*, which showed a consistent predictive power, when the recommendation task is focused on precision. In the future, we plan to collect more data points and to compare the reliability of the decisions based on offline evaluation with humans manually labelling items from the same set of variants.

REFERENCES

[1] Nazanin Alipourfard, Peter G. Fennell, and Kristina Lerman. 2018. Can you Trust the Trend? *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18* (2018). https://doi.org/10.1145/3159652.3159684

[2] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 39–46. https://doi.org/10.1145/1864708.1864721

[3] Leidy Esperanza MOLINA FERNÁNDEZ. 2018. Recommendation System for Netflix.

[4] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time Personalization Using Embeddings for Search Ranking at Airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, USA, 311–320. https://doi.org/10.1145/3219819.3219885

[5] Asela Gunawardana and Guy Shani. 2009. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *J. Mach. Learn. Res.* 10 (Dec. 2009), 2935–2962. http://dl.acm.org/citation.cfm?id=1577069.1755883

[6] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53. https://doi.org/10.1145/963770.963772

[7] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 179–186. https://doi.org/10.1145/2792838.2800184

[8] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Evaluation in information retrieval*. Cambridge University Press, 139–161. https://doi.org/10.1017/CBO9780511809071.009

[9] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 1097–1101. https://doi.org/10.1145/1125451.1125659

[10] Mukaka MM. 2012. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi Med J.* 24(3) (2012), 69–71. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/

[11] Humberto Jesús Corona Pampin, Houssem Jerbi, and Michael P. O'Mahony. 2015. Evaluating the Relative Performance of Collaborative Filtering Recommender Systems. *Journal of Universal Computer Science* 21, 13 (dec 2015), 1849–1868.

[12] Jeonghee Yi, Ye Chen, Jie Li, Swaraj Sett, and Tak W. Yan. 2013. Predictive Model Performance: Offline and Online Evaluations. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, New York, NY, USA, 1294–1302. https://doi.org/10.1145/2487575.2488215