DISPERSE: Combating Congestion with Ensemble Job Recommendation

Camille Pliquet[1,2,*], Solal Nathan[1], Yann De Coster[2], Bruno Crépon[3], Christophe Gaillac[4], Philippe Caillou[1] and Michèle Sebag[1]

[1]LISN, Université Paris-Saclay, CNRS, Inria, Gif-sur-Yvette, France
[2]France Travail, France
[3]CREST, ENSAE, Institut Polytechnique de Paris, Palaiseau, France
[4]University of Geneva, GSEM, Geneva, Switzerland

Abstract

Job recommender systems, aimed at addressing frictional unemployment due to information overload, might exacerbate the congestion of the job market, i.e., increase competition among job seekers for some over-recommended job ads. This paper investigates how to design an ensemble recommendation system, building upon a set of diverse recommendation models, such that it addresses the congestion at little or no cost in recommendation recall. The proposed approach, called Disperse (*DIverse-Seed Penalized Ensemble Re-ranking for Spread Exposure*), is based on a decision rule applied to each job seeker. For each candidate job ad, this rule combines over all models: the rank of the job ad w.r.t. the job seeker, its market share within each model, and the number of times it has actually been recommended by the ensemble. Disperse is empirically assessed in two different job recommendation contexts: the public dataset CareerBuilder-Large, and a proprietary Public Employment Service dataset. The respective trade-offs between the performance and the congestion are discussed. They suggest that a key factor controlling the trade-off is the tightness of the underlying job market, i.e., the ratio between the number of job ads and the number of job seekers.

Keywords

Job recommendation, Recommender systems, Congestion, Ensemble methods, Market share, Public employment services

1. Introduction

Following the general trend toward recommender systems [1], human resources agencies aim to leverage their data resources to best recommend job ads to job seekers and *vice versa*. As could be expected, the recommendation policies are geared to the specific priorities of the stakeholders, i.e., job seekers and recruiters, while addressing ethical concerns and complying with legal regulations [2]. Typically, public employment services (PES) aim at a *Job Recommendation for all* [3], as their mission is to leave no one behind [4]; they handle a vast majority of comparatively low-qualified job ads. In contrast, private companies might be mostly interested in job ads with high added value [5, 6].

In all contexts however, a job ad is a *rival good*: it is meant to be filled once, and a job seeker eventually holds a single job. A recommendation therefore is not only a statement about a job seeker and a job ad taken in isolation. It also lays a claim on a resource that every other job seeker might be claiming as well. Hence the phenomenon of congestion — when many job seekers are recommended the same job ad — is undesired for all stakeholders: it increases the effort required to get a match. On the job seeker side, it increases the competition, entailing a lower probability of getting the job. On the recruiter side, it increases the selection and interview load. Congestion avoidance thus is among the chief priorities of Job Recommender Systems (JRS).[1]

Building upon the state of the art in the general domain related to the recommendation of over-popular items [7], congestion avoidance has mostly been handled in three manners (more in Section 2). Pre-processing balances the training data. In-processing adds congestion-related terms to the loss used to train the recommendation policy. Post-processing modifies an existing recommendation policy to comply with some prescribed requirement.

The proposed approach is based on the claim that congestion avoidance requires a global, population-level view of the job market. Most JRSs, in contrast, are based on individual-level losses, aimed at recovering the matches in the data. Some approaches enforce this global mapping of the job seeker population onto the catalog of job ads, based on, e.g., optimal transport [8]. The recommendations of a trained model can be re-allocated in a post-processing fashion [9], or the mapping can be enforced along an in-processing scheme [10]. In both cases, the individual perspective is accounted for through the cost of transport between a job seeker and a job ad.

In a post-processing approach, the global perspective is directly available. Simulating the effect of the current policy on the job seeker population yields the congestion borne by each job ad, and by each job seeker. This counter-factual information measures the congestion *as if* the policy had been applied. It can be readily exploited to adjust the trade-off between the congestion at the population level, and the recommendation performance at the individual level.

The proposed approach, called Disperse (*DIverse-Seed Penalized Ensemble Re-ranking for Spread Exposure*), is a post-processing approach building upon ensemble learning [11, 12]. Specifically, several recommendation models are independently trained from the data, e.g. using different random seeds, training periods or regions. While such models achieve similar recall, they only moderately agree on which job ads they recommend to a given job seeker. This diversity thus offers each job seeker several roughly equally

RecSys in HR '26: The 6th Workshop on Recommender Systems for Human Resources, in conjunction with the 20th ACM Conference on Recommender Systems, September 28–October 2, 2026, Minneapolis, MN, United States.
*Corresponding author.
camille.pliquet@francetravail.fr (C. Pliquet); solal.nathan@inria.fr (S. Nathan); yann.de-coster@francetravail.fr (Y. D. Coster); crepon@ensae.fr (B. Crépon); christophe.gaillac@unige.ch (C. Gaillac); caillou@lri.fr (P. Caillou); sebag@lri.fr (M. Sebag)
https://solalnathan.com/ (S. Nathan); https://www.cgaillac.com/ (C. Gaillac); https://perso.lisn.upsaclay.fr/caillou/ (P. Caillou); https://www.lri.fr/~sebag/ (M. Sebag)

[1]Another chief priority, namely the fairness of the JRS with respect to protected attributes, is outside the scope of this paper.

relevant candidate job ads, leaving room to divert the recommendations away from the most congested ones at little relevance cost. Interestingly, DISPERSE can be viewed as a two-tier approach. In a first tier, the set of job ads relevant to a specific job seeker is selected, according to at least one recommendation model. This restricted set allows for more expensive computation to take place; considering all job ads for each job seeker would not be feasible at scale. In a second tier, the eventual recommended job ads are selected among the candidate ones, by combining their rank in each model with a penalty on their actual and counter-factual congestion. This congestion is measured by a running count of the recommendations accumulated by each job ad, as the job seekers are sequentially processed (Section 3). Note also that this candidate set filters out irrelevant recommendations, to some extent: every job ad recommended to a job seeker was deemed appropriate by at least one recommendation policy.

The empirical validation of DISPERSE is conducted on two datasets, the public CareerBuilder-Large (CB-L) [10] and a proprietary France Travail dataset [3], showing that different trade-offs between congestion and recall can be obtained.

Contributions.

- The main contribution is the DISPERSE approach, a scalable aggregation of recommender policies. It aims at limiting the congestion faced by each job seeker, and the market share of each job ad (number of job seekers the job ad is recommended to). DISPERSE only requires the ranked lists returned by the base models, and a running count of the recommendations issued so far. Its principles thus carry over to any recommender system where the exposure must be spread over the catalog, whether or not the items are rival goods;
- The analysis of the empirical results suggests that a key factor controlling the feasible trade-offs between congestion and recall is the tightness of the job market. Tightness is the ratio of the number of job ads to the number of job seekers [13, 14]. For a balanced market (tightness around 1 in CB-L), DISPERSE efficiently moves recommendations toward less popular and still relevant job ads. When the market is slack (tightness around .6 on France Travail, with job seekers outnumbering job ads), the congestion can only be limited at a high cost in recall.[2]

The remainder of the paper is organized as follows. Section 2 positions our work with respect to job recommender systems, congestion, and ensemble recommendation. Section 3 describes the congestion-aware ensemble policy. The experimental setting and the results are respectively presented in Sections 4 and 5. The paper concludes with some perspectives for further work.

2. State of the Art

This section briefly introduces the state of the art in job recommender systems (Section 2.1), before focusing on congestion avoidance in recommender systems (Section 2.2) and on the general approaches of ensemble recommendation (Section 2.3). The gap addressed by the presented approach is discussed in Section 2.4.

2.1. Job Recommender Systems

Job recommender systems (JRS) are considered along two independent dimensions. The first one is the number of tasks being tackled. A single task recommends job ads to job seekers, or, less frequently, job seekers to recruiters. The two tasks can also be tackled jointly, in the so-called reciprocal recommendation setting. The second dimension is the number of models trained per task: a single model, or an ensemble thereof. The present section is concerned with the first dimension and considers single-model approaches; ensemble approaches are reviewed in Section 2.3.

Both the single-task and the reciprocal settings are surveyed by Mashayekhi et al. [15]. Most of this literature aims to learn a relevance score for a (job seeker, job ad) pair. Each contribution addresses one obstacle to that goal: the sparsity of the interaction data [16], the two-way nature of the selection [17], the representation of resumes and job ads [18], the scale of the pool to be ranked [6], the graph structure of the past matches [19], the drift of the job seeker preferences [20], and the granularity of the skills [21]. That survey also lists congestion among the open challenges of the field.

The reciprocal setting raises specific evaluation issues, as a recommendation only succeeds if it is accepted on both sides. Yang et al. [22] accordingly complement the standard indicators with metrics assessing the coverage of the actual matches, the bilateral stability of the recommendations (whether two matched parties are recommended to each other), and the balance of the two rankings. A recent instance in the job market is that of Rus et al. [23], who train one model per task. A job ad is recommended to a job seeker when the interest is mutual. How the two rankings are combined is discussed in Section 2.3.

The setting considered in this paper is that of the French public employment service, where a JRS is deployed at the scale of a national job market [3]. In the same setting, Bied et al. [24] show through two randomized field experiments that ranking job ads by their sole hiring probability is suboptimal, even from the individual standpoint of the job seeker. A welfare-optimal ranking combines the probability that an application succeeds with the utility the job would deliver. This line of work considers each job seeker in isolation, deliberately abstracting away from congestion.

Most of the above approaches share a limitation: they score a (job seeker, job ad) pair independently of the other pairs. They are thus blind to the competition faced by a job seeker on a given job ad. ConFit [18] is a case in point. A manual inspection of 50 of its errors attributes 28% of them to job seekers who meet all the requirements of the job ad, but are outranked by more experienced competitors. The authors explicitly relate this error mode to their scoring a pair independently of the other candidates. Such errors are not amenable to better representations. The information needed to avoid them is by construction absent from the scored pair: who else is applying to the same job ad.

2.2. Popularity biases and congestion in recommender systems

Recommender systems are widely documented to over-expose already popular items — the so-called popularity

[2] Further experiments are required to support this interpretation, as the two settings also differ in other respects, notably the number of available base models (five *vs* two). More in Section 6.

bias, or *Matthew effect*, "the rich get richer" [25, 26, 7]. Two distinct mechanisms are at work. In an online setting, the bias is self-reinforcing: the recommendations shape the interactions, which are in turn used to retrain the model, so that the bias is amplified along successive retraining cycles [27]. In an offline setting, no such accumulation takes place and the over-exposure rather reflects the heavy-tailed distribution of the item popularity in the training data. Mitigation strategies mostly proceed by causal or counterfactual corrections [28, 29, 30, 31], possibly made robust to shifts of the popularity distribution [32, 33, 34]. Ning et al. [35] further distinguish the global popularity of an item, computed over all users, from its personal popularity, computed over the users sharing similar interests. Both effects are removed at inference time.

As job ads are rival goods, the popularity bias results in *congestion*: job seekers compete for a few over-recommended ads while others receive no application, wasting the matching capacity of the market. The term *congestion* is long-standing in the economics of matching markets [36, 13]; in the recommender-systems literature it appears to have been introduced by Ren et al. [37]. They work on generic catalogs, though motivated by occupancy constraints such as restaurant tables or traffic jams. Congestion is there measured by the Gini index of the recommendations received by each item, and mitigated by a re-ranking inspired from the physics of diffusion.[3]

Congestion-aware approaches are classified, as usual, into pre-, in- and post-processing. Pre-processing rebalances the training data before any model is trained. It is the least common of the three for the popularity bias [7]: Boratto et al. [38] for instance draw the training pairs along the item popularity rather than uniformly. To our best knowledge, it has not been applied to congestion. In-processing incorporates congestion in the learning objective, through an optimal transport term, entropy-regularized so as to be solved by Sinkhorn iterations within each mini-batch [39, 10], or through differentiable market-share losses [40]. Post-processing modifies the recommendations of an already trained model. The job seeker population can be transported onto the catalog of job ads, at a cost derived from the recommendation scores [9, 41]. The recommendations can also be re-ranked under per-ad quantity constraints, there with a demographic fairness objective [42].

A neighboring line of work allocates the exposure fairly among the items [43, 44, 45]. It shares with congestion avoidance the goal of spreading the recommendations over the catalog. But the exposure of an item can be granted to many users at once: nothing is lost when two users are shown the same item. The rivalry is thus left out of the picture. Li et al. [46] are an exception, and measure the harm caused by rival goods as the envy between the users competing for a same item.

The economics of matching markets brings a complementary, welfare-based perspective, reading congestion as a coordination failure. As job seekers cannot coordinate, they over-apply to the same attractive vacancies while comparable ones go unfilled [47], so that ranking each job seeker along their individual relevance is collectively suboptimal [48]. A signal posted by the market — a wage in the original setting, a ranking in a recommender system — can act as a coordination device and restore efficiency [49]. This literature also documents that such interventions cut both ways. Capping the number of applications improves welfare in some market regimes only [50]. In a field experiment, a soft cap does cut wasted applications at no cost to hiring, whereas charging for applications reduces the number of hires [51]. These levers act on the applications; a recommender system acts one step earlier, on what the job seekers get to see.

[3] Diffusion here refers to the heat equation, and not to the family of generative models bearing the same name. Note that no rivalry is modeled: what is measured is a popularity bias in the above sense.

2.3. Ensembling and fusion of independently trained recommenders

Ensemble methods build upon the diversity of models [52], be they independently trained considering (subsets of) the training data as in bagging [11, 12], or trained in a manner enforcing their decorrelation as in boosting [53, 54], or weighted *a posteriori* [55, 56]. An early ensembling strategy for recommender systems has been proposed by Jahrer et al. [57], considering heterogeneous models trained on the Netflix data, and using a user-dependent weighted linear combination thereof.

A key issue is whether the ensemble decision proceeds by combining the scores of each model, or the ranks derived from them; scores are usually not calibrated. Rank aggregation builds upon the rich literature on computational social choice [58, 59, 60] (see also [61]).

To our best knowledge however, few ensemble recommenders focus on the congestion issue. Recent ensemble rankers such as PolimiRank [62], HarmonRank [63] and UMRE [64] all aim to improve the accuracy of the fused ranking, while keeping it consistent with the rankings or objectives being aggregated (see also Yu et al. [65]). None of them inspects the impact of the ensemble on congestion. The closest exception is Dong et al. [66], who linearly aggregate the user-oriented and the item-oriented rankings produced by a single algorithm. They report a reduced Gini index of the item exposure, at a limited accuracy cost. The diversity is here obtained from the two reading directions of one model, rather than from several independently trained ones. Rank aggregation has also been made fairness-aware by Aledo et al. [67], whose integer-programming framework constrains the consensus ranking to preserve the proportion of items belonging to a protected group. The objective is there the representation of protected attributes, and not congestion.

In the JRS setting, several approaches tackle the domain adaptation problem of transferring recommendation models across markets or domains [68] (see also [62, 69]). As expected [70], the benefit of domain adaptation depends on the proximity of the source and target domains. Returning to the two-task setting of Section 2.1, Rus et al. [23] illustrate a distinct type of ensemble. The candidate-oriented and the recruiter-oriented models are trained separately, and their rankings are fused, so that a job ad is recommended when the interest is mutual. A variant learns the fusion weights for each candidate, trading the mutual interest against the exposure granted to under-represented groups of job ads.

2.4. Discussion

To summarize, congestion *per se* has essentially been tackled at the level of a single model, either by adding a congestion term to the training objective [35, 39], or by repairing the recommendations of a trained model in a post-processing

step [9]. Ensembles of recommenders, on the other hand, mostly aim at maximizing the accuracy of the recommendation, and/or the fairness of the exposure among items.

The fairness objective, e.g., ensuring a similar exposure to all items, is indeed related to the congestion objective. We claim however that both objectives differ in imbalanced contexts, such as that of the job market, where job seekers commonly outnumber job ads. Equalizing exposure on average does not prevent *some* job seekers from facing severe competition on *some* recommended job ads.

Ensembling is not, in itself, a decongestion device. Averaging several models does flatten the tails of their score distributions, which might be expected to ease congestion. Empirically, however, the naive ensemble does not reduce congestion on either dataset: it leaves it essentially unchanged on BFC and slightly degrades it on CB-L (Section 5). One mechanism works against decongestion. Fusing models trained on different regions or periods amounts to ranking along the popularity of the pooled population, rather than that of any single model. This pooled popularity is a coarser aggregate, and thus more congestion-prone, in the sense of the global *vs* personal popularity of Ning et al. [35]. A related caveat is that correcting a single bias factor is known to be insufficient when the item selection is driven by several factors at once [71]. Here, these factors are the popularity of a job ad, and the region or period its model originates from. Our experiments do not probe this regime, as the base models differ by their random seed only.

The question is thus how an ensemble can best be turned into a decongestion device. The approach presented below builds upon an ensemble of models, and seeks an ensembling method that reduces the congestion over the job ads. The diversity of the models can be obtained by training them with different random seeds, or on different periods of time or geographical regions.

3. Algorithm

We propose a model-agnostic, post-processing algorithm that turns the outputs of several base job recommenders into a single set of recommendations while controlling congestion. The algorithm proceeds in two stages. First, a seed-ensembling stage reconciles the ranked lists produced by the different seeds into a single score per (seeker, job ad) pair, combining, for each job ad, how highly it is ranked and its market share (subsection 3.2, subsection 3.3). Second, a greedy congestion-control stage sequentially assigns job ads to job seekers, discounting job ads in proportion to how often they have already been recommended (subsection 3.4, subsection 3.5).

3.1. Notations and problem setup

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ denote a set of n job seekers and $\mathcal{Y} = \{y_1, \ldots, y_m\}$ a set of m job ads. A recommender model M expresses the adequacy between a job seeker x and a job ad y through a real-valued score $s(x, y)$. For a given x, the score function thus induces a ranking over the ys, denoted $r_x(y)$. A JRS usually recommends the top-k job ads according to r_x to each user x, with $k = 10$ most often.

Let $M_1, \ldots, M_N$ denote a set of N recommender models.[4] For a job seeker x, each model returns an ordered Top-K list ($K \gg k$; K is set to 100 in the experiments) :

$$M_i(x) = \big(y_{i,1}, \ldots, y_{i,K}\big), \quad i \in \{1, \ldots, N\}$$

where $y_{i,1}$ is the highest-ranked job ad.

Candidate pool (the bag): For a job seeker x we consider the union of the recommendations produced by all models,

$$B(x) \;=\; \bigcup_{i=1}^{N} \{y_{i,1}, \ldots, y_{i,K}\} \;\subseteq\; \mathcal{Y} \tag{1}$$

so that $K \le |B(x)| \le NK$: if the lists of the N models overlap completely, the candidate pool is minimal ($|B(x)| = K$), whereas if they are pairwise disjoint it reaches its maximal size ($|B(x)| = NK$). All subsequent scoring is defined over the pairs (x, y) with $y \in B(x)$.

Guidelines The market-share penalty of subsection 3.2 is computed within each base model, so that the correction tracks the bias of each model separately. The ensemble uses a global counter (subsection 3.4) to handle the actual congestion of the fused output. Overall the ensemble algorithm aims to: (i) exploit seed diversity to obtain a more robust relevance estimate than any single model provides, and (ii) spread recommendations across the non-recommended or less recommended job ads.

3.2. Rank and market-share signals

Two complementary signals are extracted from the base recommenders.

Rank: For each model M_i and each pair (x, y) with $y \in B(x)$ we define the rank

$$r_i(x, y) = \begin{cases} \text{position of } y \text{ in } M_i(x), & \text{if } y \in M_i(x) \\ K + 1, & \text{otherwise} \end{cases} \tag{2}$$

so that $r_i(x, y) \in \{1, \ldots, K+1\}$, the value $K+1$ encoding the absence of y from the list of model M_i. We use the normalized rank:

$$\bar{r}_i(x, y) \;=\; \frac{r_i(x, y)}{K + 1} \;\in\; \Big(0,\, 1\Big] \tag{3}$$

Market share: For a model M_i, the market share of a job ad y is its empirical recommendation frequency over the whole population,

$$MS_i(y) \;=\; \frac{\big|\{x \in \mathcal{X} : y \in M_i(x)\}\big|}{K\, n} \tag{4}$$

the denominator Kn being the total number of recommendations produced by M_i. The market shares measure model-level congestion: in the case of high congestion, a few job ads concentrate a large recommendation mass while most are almost never recommended (the ensemble aims to leverage the fact that over-popular job ads may differ from one model to another). Note that MS_i is computed once for each model.

The two indicators (rank and model-based market share) are aggregated into a linear combination (Eq. 6).

[4] In the experiments, the models are independently learned from the same training set using the same neural architecture. More generally, the model diversity can be enforced using different model spaces, and/or training data gathered across different geographical or temporal contexts.

Log-normalization of the market share: The choice of a linear aggregation is hindered by the scale of the two indicators and the distribution of the market shares: most values of MS_i are concentrated near 0 when the normalized rank $\bar{r}_i$ lives on $[0, 1]$. Therefore, setting a single λ weight on the market shares can hardly be effective: for small λ values the market share impact is negligible against the rank term, whereas a large λ over-penalizes the most recommended job ads[5] and overwhelms the rank signal (thus adversely affecting the recommendation relevance) and leaves them behind moderately recommended job ads.

For this reason, a log transformation [72] is first applied on the market shares (where the small positive factor ε set to 10^{-8} in the experiments, is used to prevent numerical issues) :

$$\widetilde{MS}_i(y) = \frac{\log\big(MS_i(y) + \varepsilon\big) - \log \varepsilon}{\log\big(MS_i^{\max} + \varepsilon\big) - \log \varepsilon} \in [0, 1], \quad (5)$$
$$MS_i^{\max} = \max_{y \in \mathcal{Y}} MS_i(y)$$

This transformation maps the market share onto [0,1], expanding the low-mass region while contracting the tail, so that $\widetilde{MS}_i$ is commensurable with $\bar{r}_i$.

3.3. Per-model score

For each model M_i we combine the two normalized signals into a per-model score (the higher the better) :

$$v_i(x, y) \;=\; -\,\bar{r}_i(x, y) \;-\; \lambda\, \widetilde{MS}_i(y), \quad \lambda \geq 0 \quad (6)$$

The first term rewards top-ranked job ads; the second penalizes job ads that are already congested within model M_i. Hyperparameter λ acts as a trade-off weight between rank quality and model-level popularity.[6]

3.4. Congestion-aware ensemble score

Aggregation: The per-model scores $v_1(x, y), \ldots, v_N(x, y)$ are aggregated into a single score through an operator $Agg \in \{\max,\ \text{mean},\ \min\}$:

$$V(x, y) \;=\; Agg(v_1(x, y), ..., v_N(x, y)) \quad (7)$$

where the aggregation operator Agg is one of

$$\max_i v_i(x, y), \qquad \frac{1}{N}\sum_{i=1}^{N} v_i(x, y), \qquad \min_i v_i(x, y). \quad (8)$$

The three operators induce qualitatively different ensembling regimes. The max operator is optimistic: it retains a job ad as soon as one seed ranks it well, favoring recall over agreement. The min operator is conservative (consensus): a job ad scores well only if all seeds concur, favoring precision and stability. The mean interpolates between the two.

Congestion-counter score: The ensemble decision making involves a counter $c(y)$ ($c : \mathcal{Y} \to \mathbb{N}$), initialized to 0, recording the number of times job ad y has been recommended so far during the algorithm pass over the job seekers. Eventually, the ensemble selection considers the linear combination of the aggregated score $V(x, y)$ and $c(y)$, weighted with penalty weight $\alpha \geq 0$:

$$S(x, y) = V(x, y) \;-\; \alpha\, c(y) \quad (9)$$

Along this selection rule, a job ad already recommended many times becomes progressively less attractive, even if its base score is high; note that the penalty is unbounded.[7]

3.5. Sequential assignment

Sequential decision making is considered in DISPERSE, iteratively selecting k recommendations for each job seeker (in random order; more on this below). For each job seeker x, top-10 job ads are selected from the candidate pool $B(x)$, forming a ranked list $L(x)$:

$$L(x) = \text{top-}k_{\,y \in B(x)}\, S(x, y), \quad (10)$$

Counter c is incremented by 1 for all job ads in $L(x)$, enabling to gradually lower their S scores for all further job seekers and enabling to consider other alternatives, thus regulating the global congestion suffered by the ensemble.

Because the counter $c(\cdot)$ is updated during the process, the output is order-dependent: the later a job seeker x is considered, the lower their chances of being recommended their best job ads, if these have often been recommended earlier. Processing job seekers in a random order (the choice in our experiments) treats them exchangeably in expectation, whereas a fixed priority order lets attractive job ads be allocated first to a chosen subgroup. We shall return to this point in Sections 5 and 6.

4. Experimental setting

The goal of the experiments is to assess the trade-off between the recommendation performance and the congestion achieved by the ensemble.[8] A secondary goal is to assess the stability of the ensemble w.r.t. its hyperparameters, and identify the hyperparameter configurations most appropriate depending on the considered job market. Another goal is to assess the variance of the performance w.r.t. the order of the job seekers (Section 3.5).

Datasets and Individual Models For the sake of reproducibility, we first consider the public CareerBuilder-Large (CB-L) dataset [10], an extract of the CareerBuilder dataset released for the Kaggle Job Recommendation Challenge [73]. It involves 42,346 job seekers and 40,542 job ads, with a median number of 7 applications per job seeker. Five models, independently trained using the same training and validation subsets with different random seeds, are learned by optimizing the binary cross-entropy loss [40].

The second dataset is a proprietary France Travail dataset, gathered in the Bourgogne Franche-Comté region in 2025

[5] Note that the ensemble will consider instant market shares, sequentially computed using a counter (Eq. 9), thus offering another mitigation of the (ensemble-based) over-recommended job ads.

[6] The $[0, 1]$ scale is not strictly necessary, since λ is unbounded and could be any given value, it could absorb any rescaling of the signals. Its role is instead to make λ interpretable as a relative weight, so that $\lambda = 1$ weights market share and rank.

[7] An alternative, for further work, is to cap the number of times a job ad can be recommended.

[8] The ensemble code and the per-model job-seeker/job ad rankings on CB-L are publicly available at https://gitlab.com/vadore/DISPERSE. The BFC rankings are derived from proprietary France Travail data and cannot be released.

and referred to as BFC, involving 72,589 job seekers and 44,735 job ads, with one match per job seeker. Two models[9] are likewise independently trained on BFC (weeks from the 9th week in 2023 to the 9th week in 2025) using different random seeds.

Performance indicators A first performance indicator is the recall@10 (R@10), averaging over the job seekers the percentage of their applications (for CB-L) or matches (for BFC) that belong to the top-10 recommended job ads. On CB-L, the recall is measured on the pre-defined test set; on BFC, the recall is measured over the matches during the 18th week in 2025.

The congestion is measured by the Gini index of the market shares of the job ads. For a policy recommending a list $L(x)$ of k job ads to each job seeker x (Eq. 10), the market share of a job ad y is its share of the $k\,n$ recommendations issued :

$$MS(y) \;=\; \frac{\big|\{x \in \mathcal{X} \;:\; y \in L(x)\}\big|}{k\,n}, \qquad \sum_{y\in\mathcal{Y}} MS(y) = 1 \tag{11}$$

Eq. 4 is the same quantity, computed on the top-K lists internally used by DISPERSE. Originally a measure of income inequality [74] and since adapted to item exposure in recommender systems [75], the Gini index over this distribution reads :

$$\text{Gini} \;=\; \frac{\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}}\big|MS(y) - MS(y')\big|}{2\,m} \tag{12}$$

It ranges from 0, when all job ads are recommended equally often, to $1-1/m$, approached when a single job ad captures all recommendations: the lower the Gini index, the lower the congestion. Being computed over the whole catalog $\mathcal{Y}$, it accounts for the job ads that are never recommended, and it is computed identically for the individual models, their naive ensemble and DISPERSE. We also report the coverage (percentage of job ads recommended at least once).

Because DISPERSE processes the job seekers sequentially, its output depends on their ordering as soon as $\alpha > 0$. Each configuration is therefore run over 10 independent random orderings of the job seekers, and we report the mean and standard deviation of every performance indicator over these 10 runs.

Hyperparameter setting The bag of job ads associated with each job seeker includes the union of top-K (K=100) job ads over all models. Other hyperparameters include the penalty weight $\lambda \geq 0$ on the market shares (Eq. 6) and the weight $\alpha \geq 0$ on the instant market share of the job ads (Eq. 9). A discrete hyperparameter is the mode of aggregation of the values (Eq. 8). Their domain of variation is reported in Table 1, amounting to 324 configurations.

Analysis The results, measuring the performance and Gini index obtained over all hyperparameter configurations, are summarized in a 2D plot, depicting the best (non-dominated) trade-offs along their Pareto front [76].

These trade-offs are assessed with respect to reference points, respectively corresponding to the recall-congestion of each model, and that of their naive ensemble, corresponding to DISPERSE with configuration $\alpha = \lambda = 0$ and *mean* aggregation mode.

[9]These models are two-tier neuronal nets using both job seeker and job ad descriptions [3]. Only two models were considered due to the computational resources; further work is concerned with handling more models in BFC.

Table 1
DISPERSE hyperparameters and range

Hyperparameter	Values
λ (market-share weight)	$\{0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5, 1\}$
agg (aggregation)	$\{\texttt{max}, \texttt{mean}, \texttt{min}\}$
α (congestion penalty)	$\{0, 10^{-4}, 5\cdot10^{-4}, 10^{-3}, 2\cdot10^{-3}, 5\cdot10^{-3}, 10^{-2}, 2\cdot10^{-2}, 5\cdot10^{-2}, 10^{-1}, 2\cdot10^{-1}, 5\cdot10^{-1}\}$

5. Empirical validation

This section reports on the empirical results of DISPERSE on CB-L and BFC. It proposes some analysis of the effect of the job market tightness, the ratio between the number of job ads and the number of job seekers, on the trade-off between the performance and the congestion of the ensemble recommender.

5.1. Results on CB-L

The reference regime On this dataset, the individual recommendation models obtain on average moderate recall (.181) and Gini index (.495), with a coverage of .940. The 5 individual models explore different regions of the job-ads space (the top-10 recommendations overlap by 44% on average). Most interestingly, the naive ensemble approach yields a gain in recall (from .181 to .212) while moderately degrading the Gini index (from .495 to .544), and the coverage (from .940 to .912). The dataset thus appears to be comparatively difficult in terms of recall, with a low congestion, where the naive ensemble alone yields recall gains, while degrading lightly the congestion.

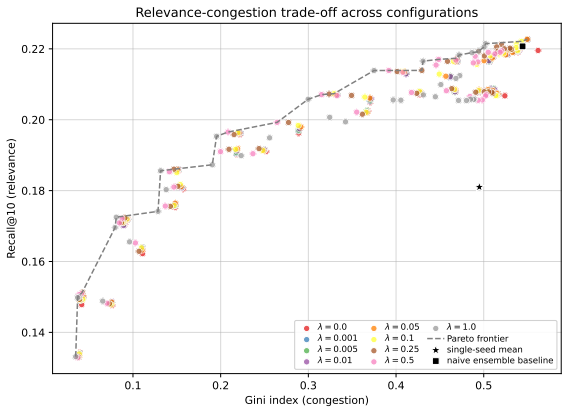

Figure 1: DISPERSE on CB-L: Recall-Gini performances over all hyperparameter configurations, and the corresponding Pareto front computed on the validation set.

The Pareto front The Pareto front (Fig. 1) starts (top right) at its highest recall configuration, and correspondingly highest Gini index (R@10.223 and Gini .549, validation set). When going to the left (lower recalls and Gini

indices), the Pareto front shows two regimes. In a first segment with low slope, the recall degrades very gracefully for a substantial improvement of the Gini index: bringing Gini to $\approx .10$ reduces the recall to $\approx .17$ (thus a 23% loss). In the second segment (sharp slope) recall decreases very sharply as the Gini index reaches its empirical minimum (Gini .031, coverage 100%).

A few representative DISPERSE configurations are reported in Table 2: recall (respectively Gini) monotonically degrades (resp. improves) as α increases.

Table 2
DISPERSE on CB-L: representative Pareto-optimal configurations selected on the validation Pareto front, with R@10measured on the test set, from relevance-optimal (top) to Gini-optimal (bottom). The first line is the baseline (naive ensemble). Values are means over 10 seeds (independent random orderings of the DEs). Dispersion across seeds is small: standard deviations do not exceed .0054 (R@10), .0004 (Gini), .0003 (Entropy) and .0005 (Cov.)

λ	agg	α	Gini	Cov.	Entropy	R@10
.0	*mean*	*.000*	$.544_{\pm.0000}$	*.912*	$10.097_{\pm.0000}$	$.212_{\pm.0000}$
.05	max	.000	$.549_{\pm.0000}$	.911	$10.086_{\pm.0000}$	$.202_{\pm.0000}$
1.0	mean	.000	$.502_{\pm.0000}$	.942	$10.177_{\pm.0000}$	$.212_{\pm.0000}$
.5	mean	.005	$.449_{\pm.0004}$	.957	$10.273_{\pm.0002}$	$.207_{\pm.0018}$
1	mean	.01	$.375_{\pm.0002}$	.980	$10.378_{\pm.0003}$	$.202_{\pm.0029}$
.5	max	.010	$.264_{\pm.0004}$	.993	$10.493_{\pm.0002}$	$.187_{\pm.0029}$
1.0	mean	.050	$.195_{\pm.0003}$	1.000	$10.548_{\pm.0002}$	$.185_{\pm.0022}$
1.0	mean	.100	$.131_{\pm.0002}$	1.000	$10.582_{\pm.0001}$	$.174_{\pm.0028}$
1.0	mean	.200	$.081_{\pm.0004}$	1.000	$10.599_{\pm.0001}$	$.162_{\pm.0035}$
2.0	max	.500	$.031_{\pm.0003}$	1.000	$10.608_{\pm.0000}$	$.128_{\pm.0054}$

Two configurations with $\alpha = 0$ deserve a specific mention. With *mean* aggregation, $\lambda = 1.0$ matches the recall of the naive ensemble (.212) while lowering the Gini index from .544 to .502: the model-level market-share penalty alone, with the congestion counter switched off, already relieves 8% of the congestion at no cost in recall. The relevance-optimal point of the validation front, $\lambda = .05$ with *max* aggregation, does not carry over to the test set: its test recall (.202) falls below that of the naive ensemble, at a marginally higher Gini (.549 *vs* .544).

This quasi-concave Pareto front suggests that a substantial part of the initial congestion can be mitigated, by moving recommendations toward less popular job ads at little recall cost. Such a move is facilitated by the fact that the job market tightness (ratio of the number of job ads to the number of job seekers) is around 1, thus a balanced market, and each job seeker is associated with several job ads, making it feasible to reallocate recommendations (all the more so as each job seeker applies on several job ads).

Sensitivity w.r.t. hyperparameter α The sensitivity of the recall and Gini index w.r.t. α (the penalty weight of the current market share in the ensemble decision, Eq. 9) is respectively displayed in Figs. 2 and 3, for different values of λ (penalty weight on the model-dependent market share, Eq. 6), and different aggregation mechanisms (Eq. 8). For readability, only four values of λ are depicted, the remaining values are deferred to Appendix A.1.

More generally, as α increases from 0 to .1 (rightmost part of the Pareto front, Fig. 1), the recall degrades gradually, whereas the Gini index decreases sharply. For higher α values ($> .1$, leftmost part of the Pareto front), the recall decreases sharply whereas the Gini index is almost flat and close to its minimum.

The *max* aggregation mode is the one with the most impact on both recall and Gini. For $\lambda = .1$, increasing α from 0 to 10^{-3} entails a 9% Gini decrease (from .543 to .492) and a 3% recall drop (from .222 to .216). Note however that in terms of recall only, *max* is dominated by other aggregation modes (*min* and *mean*) except for very small values of α (close to the reference point).

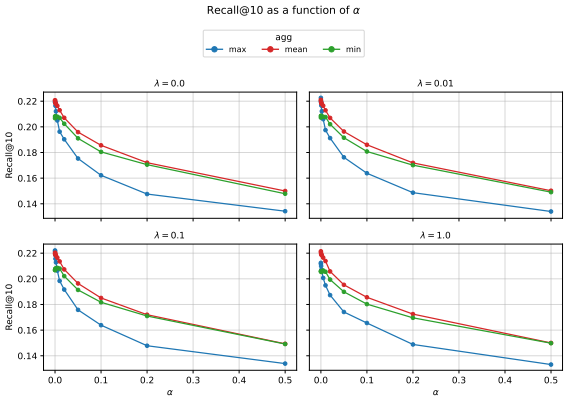

Figure 2: DISPERSE on CB-L: Sensitivity of R@10w.r.t. α, for different aggregation modes and $\lambda \in [0, .01, .1, 1]$ (from top-left to bottom-right sub-images).

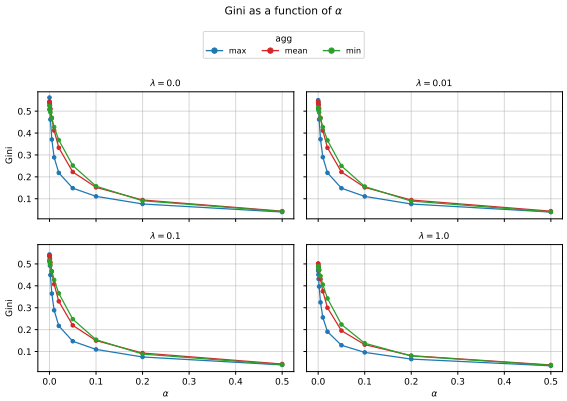

Figure 3: DISPERSE on CB-L: Sensitivity of Gini index w.r.t. α, for different aggregation modes and $\lambda \in [0, .01, .1, 1]$ (from top-left to bottom-right sub-images).

The respective gains in Gini (ΔGini) and recall (ΔRecall) and their ratio are depicted in Table 3, showing that most gains in terms of the Gini index occur for low values of α, at the expense of a moderate loss of recall (a gain of 50% in Gini costs circa 10% in recall). Additionally, α is fully effective in the sense that it can effectively bring Gini close to its minimum.

Sensitivity w.r.t. hyperparameter λ and aggregation mode Figure 4 shows the (recall, Gini index) Pareto front attached to the different values of λ (in color) for different values of α for all three aggregation modes (large values of λ are discarded due to poor results). As above, the Pareto front shows a gentle slope close to the reference point. The *max* aggregation mode reaches the reference point; the Pareto fronts associated with diverse λ values show a relatively

Table 3
DISPERSE on CB-L: Recall, Gini index and coverage achieved *vs* α for a few configurations on the Pareto front; the efficiency is measured by the ratio of Gini gain per unit of recall lost ($\lambda = .1$, $\texttt{agg} = \texttt{max}$).

α	R@10	Gini	Cov.	$\frac{\lvert\Delta\text{Gini}\rvert}{\lvert\Delta\text{Rec.}\rvert}$	$\frac{\Delta\text{Cov.}}{\lvert\Delta\text{Rec.}\rvert}$
.000	.222	.543	.918		
.001	.216	.492	.939	7.87	3.26
.002	.213	.449	.953	15.58	5.20
.005	.206	.364	.975	13.10	3.32
.010	.198	.288	.988	9.47	1.64
.020	.192	.217	.995	10.62	1.10
.050	.176	.147	.999	4.42	.23
.100	.164	.110	1.000	3.12	.07
.500	.134	.039	1.000	2.37	.00

high variance. The *mean* aggregation mode is much more stable w.r.t. λ and it dominates the other modes except for very low λ.

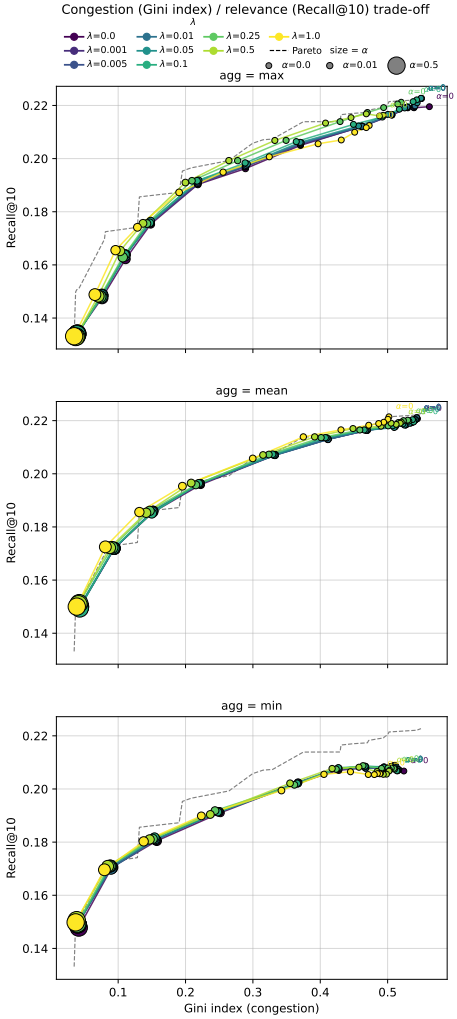

Figure 4: DISPERSE on CB-L: Sensitivity of the (recall; Gini index) Pareto front for λ varying in $[0, 1]$, with aggregation mode *max*, *mean* and *min* from top to bottom.

Discussion Two facts characterize this dataset. Firstly, the naive ensemble trades a marginal congestion cost for a large recall gain over the individual models (R@10 from .181 to .212, Gini index from .495 to .544). Secondly, DISPERSE improves on the naive ensemble on the congestion side (Gini index from .544 to .502) at unchanged recall (R@10 .212); moving further along the Pareto front then trades recall for lower congestion.

The two penalty weights play distinct roles. Weight λ acts on the popularity each base model has inherited from its training data: with *mean* aggregation and $\alpha = 0$, raising λ from 0 to 1 lowers the Gini index from .544 to .502 at unchanged recall. Weight α acts on the congestion the ensemble itself builds up as it processes the job seekers, and is the weight that actually navigates the front: it alone brings the Gini index close to its empirical minimum, and it does so at a recall cost.

The choice of the best hyperparameter configuration, that is, the desired trade-off on the Pareto front, indeed belongs to the stakeholders. With similar priorities between the recall and the congestion objectives, an achievable trade-off could be to halve the Gini index w.r.t. the naive ensemble (from .544 to .264) and raise coverage to 99% while keeping 88% of its recall, with configuration $\lambda = .5$, aggregation $\texttt{max}$, $\alpha = 10^{-2}$. Most generally, the Pareto front shows that trade-offs exist, with lower congestion at the expense of little performance loss.

5.2. Results on BFC

The reference regime On this dataset, the two considered individual recommendation models obtain comparatively high recall (.411 and .418) and a very high Gini index (.921 and .915), with a moderate coverage (.402, .418). The two models are moderately different (the top-10 recommendations overlap by 50% on average). The naive ensemble approach yields a moderate gain in recall (.427) at an essentially unchanged Gini index (.921, *vs* .921 and .915 for the individual models), and a slightly lower coverage (.376, *vs* .402 and .418). The congestion is very high near the reference point (Gini above .89) and, as shown below, cannot be brought below .537, reflecting a market where each job seeker has essentially one match. Unlike on CB-L, where ensembling slightly worsens the Gini, here it leaves it essentially unchanged; in neither dataset does ensembling reduce congestion.

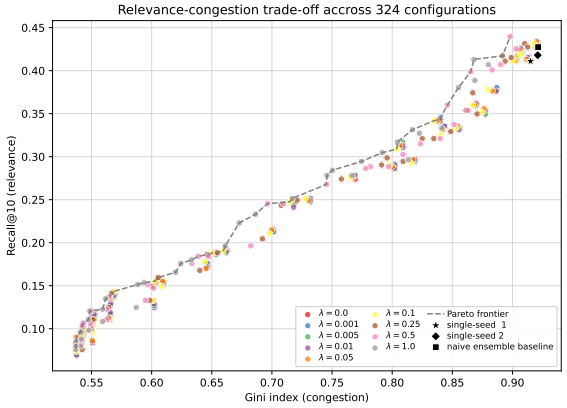

Figure 5: DISPERSE on BFC: Recall-Gini performances over all hyperparameter configurations, and the corresponding Pareto front. The Gini index is computed over the whole catalog.

The Pareto front Figure 5 displays the Pareto front obtained on the BFC dataset. Interestingly, DISPERSE dominates the naive ensemble (Recall@10 = .440 *vs* .427, Gini .898 *vs* .921, coverage .453 *vs* .376). Thereafter, the Pareto front declines from the reference point and flattens against a Gini floor around .537, reached only for a very poor recall (.076); it cannot be pushed lower.

A few representative DISPERSE configurations are reported in Table 4: recall (respectively Gini) sharply decreases (resp. slightly improves) monotonically as α increases.

Table 4
DISPERSE on BFC: representative Pareto-optimal configurations, from relevance-optimal (top) to Gini-optimal (bottom). The first line is the baseline (naive ensemble). The Gini index is computed over the whole catalog. Results are for a single ordering of the job seekers (no multi-seed averaging on BFC).

λ	agg	α	Gini	Cov.	Entropy	Recall@10
.0	*mean*	*.000*	*.921*	*.376*	*8.306*	*.427*
.5	mean	.000	.898	.453	8.607	.440
1.0	mean	.000	.868	.610	8.908	.413
1.0	mean	.0001	.855	.631	9.052	.380
.1	max	.001	.804	.566	9.375	.309
1.0	max	.001	.750	.727	9.619	.284
1.0	max	.002	.717	.749	9.737	.252
1.0	mean	.005	.673	.791	9.876	.223
.1	mean	.010	.650	.753	9.924	.188
1.0	mean	.020	.588	.816	10.069	.151
1.0	mean	.100	.547	.826	10.131	.110
1.0	mean	.500	.537	.829	10.141	.076

A key difference compared to the CB-L experiments is that the Gini index cannot be decreased below .537 (obtained for a very low recall) whereas it reaches .031 for CB-L. This difference is interpreted as reflecting the fact that job seekers have essentially one match in the training data (corresponding to a hire as opposed to an application in CB-L), favoring the lesser diversity of the models; moreover, only two models are considered.[10] Given this restricted diversity, the ensemble does not manage to move recommendations toward less popular job ads at limited recall cost.

Sensitivity w.r.t. hyperparameter α The sensitivity of the recall and Gini index w.r.t. α (Eq. 9) is respectively displayed in Figs. 6 and 7, for different values of λ (Eq. 6), and different aggregation mechanisms (Eq. 8). For readability, only four values of λ are depicted, the remaining values are deferred to Appendix A.2.

The respective gains in Gini (ΔGini) and recall (ΔRecall) and their ratio are reported in Table 5, showing that substantial losses in recall are suffered to gradually improve the Gini index and the coverage. For $\lambda = .1$ and $\texttt{agg} = \texttt{max}$, increasing α to 10^{-4} decreases the Gini index by 3% (from .907 to .879), at the cost of a recall drop of 10% (from .419 to .378). For $\alpha = 10^{-3}$ the Gini index decreases by 11% (from .907 to .804), at the cost of a recall drop of 26% (from .419 to .309).

There exists no DISPERSE configuration in the considered range that can halve the Gini index, although the coverage can be considerably increased (though for a poor R@10).

[10] Furthermore, the slackness of the job market (tightness around .6) also explains the overlap among recommendations, focusing on e.g. sectorial or geographical markets.

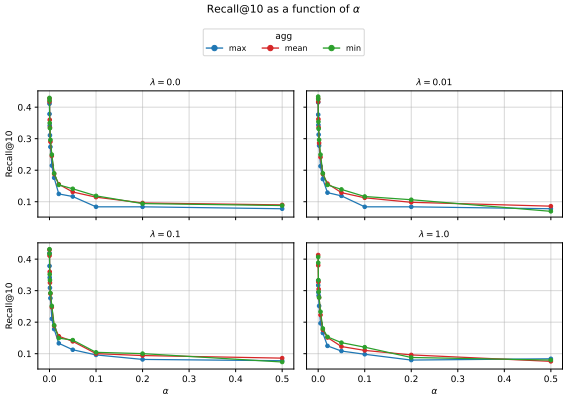

Figure 6: DISPERSE on BFC: Sensitivity of Recall@10 w.r.t. α, for different aggregation modes and $\lambda \in [0, .01, .1, 1]$ (from top-left to bottom-right sub-images).

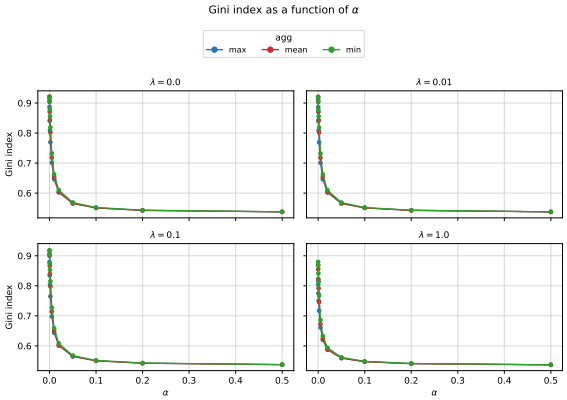

Figure 7: DISPERSE on BFC: Sensitivity of Gini index w.r.t. α, for different aggregation modes and $\lambda \in [0, .01, .1, 1]$ (from top-left to bottom-right sub-images).

Table 5
DISPERSE on BFC: Recall, Gini index and coverage achieved *vs* α for a few configurations on the Pareto front; the efficiency is measured by the ratio of Gini gain per unit of recall lost ($\lambda = .1$, $\texttt{agg} = \texttt{max}$). The Gini index is computed over the whole catalog.

α	Recall@10	Gini	Cov.	ΔRec.	$\frac{\lvert\Delta\text{Gini}\rvert}{\lvert\Delta\text{Rec.}\rvert}$	$\frac{\Delta\text{Cov.}}{\lvert\Delta\text{Rec.}\rvert}$
.000	.419	.907	.456			
.0001	.378	.879	.481	−.041	.67	.60
.0005	.342	.834	.530	−.037	1.23	1.33
.001	.309	.804	.566	−.033	.93	1.11
.002	.276	.765	.614	−.033	1.19	1.46
.005	.211	.698	.695	−.065	1.02	1.24
.010	.178	.644	.749	−.033	1.65	1.64
.020	.133	.601	.785	−.045	.96	.80
.050	.112	.565	.809	−.020	1.74	1.16

Sensitivity w.r.t. λ and aggregation mode High values of λ result in a poor trade-off between congestion and recall, even more markedly than in CB-L. This fact is attributed to the larger market share of job ads (Eq. 6), making the value poorly informative of the rank and adversely affecting the relevance of the recommended job ads by putting job ads with low market share first. The impact of the aggregation mode is comparatively negligible (this must be taken with a grain of salt as only two models are considered).

Interestingly however, the best recall is obtained with DISPERSE with configuration $\lambda = .5$, mean, $\alpha = 0$, improving recall w.r.t the baseline (.427 to .440) and Gini index (.921 to .898).

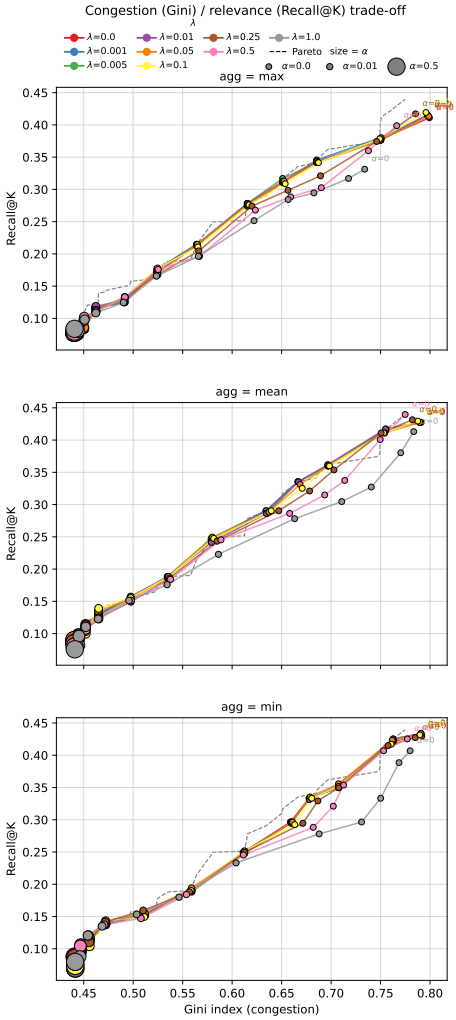

Figure 8: DISPERSE on BFC: Sensitivity of the (recall; Gini index) Pareto front for λ varying in $[0, 1]$, with aggregation mode *max*, *mean* and *min* from top to bottom.

Discussion DISPERSE results on BFC contrast with those on CB-L, as the trade-off between congestion and recall offers no sweet spot: the Pareto front is nearly linear before flattening against a Gini floor around .537, in contrast with the concave shape of the CB-L front. Two tentative interpretations are offered for this negative result. On the one hand, the dataset reflects a slacker job market (tightness around .6), with job seekers outnumbering job ads. On the other hand, only two models were considered due to computational and memory bounds, thus supporting the ensemble with less information.

On the impact of the job-seeker ordering Since the counter $c(y)$ is updated as job seekers are processed, the output depends on their ordering once $\alpha > 0$; at $\alpha = 0$ each recommendation is computed independently and every metric is invariant across orderings. On CB-L, averaging over 10 random orderings shows this dependence to be small for the concentration metrics (the Gini standard deviation does not exceed .0004) and moderate for recall (up to .0054), well below the variation induced by the hyperparameters; a single random ordering is thus already representative. On BFC, results are reported for a single ordering; given its tighter offer supply, the ordering effect is expected to be at least as large, but it is not quantified here.

6. Conclusion and Perspectives

The presented ensemble recommender system, DISPERSE, has demonstrated its ability to comprehensively explore the trade-off between congestion and recall on two datasets with different characteristics. On the public CareerBuilder-Large dataset, a concave Pareto front is obtained, and some decent improvement of the congestion can be obtained with limited loss in recall. On the proprietary France Travail BFC dataset, the Pareto front decreases quasi linearly and abruptly from its reference recall and Gini index to a situation where the congestion is still high and the recall is quite low.

Still, on both datasets DISPERSE improves on the individual models and on their naive ensemble, albeit in different ways. On BFC, a single configuration ($\lambda = .5$, *mean* aggregation, $\alpha = 0$) improves on all of them on every indicator: recall (.440 *vs* .427 for the naive ensemble), Gini index (.898 *vs* .921) and coverage (.453 *vs* .376). On CB-L, DISPERSE relieves congestion at unchanged recall (Gini index .502 *vs* .544 for the naive ensemble) but does not raise recall over it on the test set; moving further along the front trades recall for lower congestion.

Regarding the Pareto fronts, a factor potentially explaining their difference is the difference in the tightness of the two job markets. A first research perspective is to confirm this conjecture by extracting a subset of the public CB-L dataset with the same initial characteristics as those of BFC. Incidentally, this remark also points to the poorly documented differences between publicly available job recommendation datasets, and those encountered in Public Employment Services.

Another short-term perspective is to reconsider the sequential DISPERSE processing, and investigate the impact of different orderings of the job seekers. One natural criterion in the employment setting is to prioritize job seekers along some exogenous score (e.g. reflecting their employability), i.e. to favor job seekers for whom a good match is most rare. Ordering the job seekers along such a score in DISPERSE would ensure that they can be recommended their best job ads before these are saturated. Along the same lines, the ordering can be used to spread the congestion more evenly across job seekers, e.g., by serving first those facing the highest competition according to the individual models.

Acknowledgments

The authors warmly thank Guillaume Bied for many discussions and suggestions, and France Travail for granting access to the data used in this study.

Ethical concerns The proposed approach deliberately trades individual relevance for collective decongestion: some job seekers are recommended job ads that are not the most relevant ones according to the models, and this cost is not necessarily borne evenly, as it may depend on

the position of the job seeker in the processing order or on the tightness of their segment of the job market. Furthermore, recall is only a proxy for the actual goal of supporting the return to employment; a policy that traces a favorable recall-congestion trade-off offline might still harm some job seekers if deployed without further validation. These risks command a cautious and closely monitored deployment. The current AI policy in the considered Public Employment Service involves extensive experimental campaigns before any new recommendation policy can be deployed, monitoring both its acceptability by the job seekers and its efficiency, and its impact on the most disadvantaged populations.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Opus 4.8 in order to: Grammar and spelling check, Text Translation, Paraphrase and reword. Thereafter, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Bennett, S. Lanning, The Netflix prize, in: Proceedings of KDD Cup and Workshop 2007, 2007, pp. 3–6.

[2] European Parliament and Council of the European Union, Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), Official Journal of the European Union, L series, 2024. URL: http://data.europa.eu/eli/reg/2024/1689/oj, cELEX:32024R1689.

[3] G. Bied, S. Nathan, E. Pérennes, M. Hoffmann, P. Caillou, B. Crépon, C. Gaillac, M. Sebag, Toward job recommendation for all, in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI), 2023, pp. 5906–5914. URL: https://www.ijcai.org/proceedings/2023/655. doi:10.24963/ijcai.2023/655.

[4] United Nations General Assembly, Transforming our world: the 2030 Agenda for Sustainable Development, General Assembly resolution A/RES/70/1, 2015. URL: https://docs.un.org/en/A/RES/70/1, adopted 25 September 2015, seventieth session.

[5] W. Shalaby, B. AlAila, M. Korayem, L. Pournajaf, K. AlJadda, S. Quinn, W. Zadrozny, Help me find a job: A graph-based approach for job recommendation at scale, in: 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 1544–1553. doi:10.1109/BigData.2017.8258088.

[6] S. C. Geyik, Q. Guo, B. Hu, C. Ozcaglar, K. Thakkar, X. Wu, K. Kenthapadi, Talent search and recommendation systems at LinkedIn: Practical challenges and lessons learned, in: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2018, pp. 1353–1354. doi:10.1145/3209978.3210205. arXiv:1809.06481.

[7] A. Klimashevskaia, D. Jannach, M. Elahi, C. Trattner, A survey on popularity bias in recommender systems, User Modeling and User-Adapted Interaction (UMUAI) 34 (2024) 1777–1834. doi:10.1007/s11257-024-09406-0. arXiv:2308.01118.

[8] G. Peyré, M. Cuturi, Computational optimal transport: With applications to data science, Foundations and Trends in Machine Learning 11 (2019) 355–607. doi:10.1561/2200000073.

[9] G. Bied, E. Pérennes, V. A. Naya, P. Caillou, B. Crépon, C. Gaillac, M. Sebag, Congestion-avoiding job recommendation with optimal transport, in: FEAST Workshop at ECML-PKDD 2021, 2021. URL: https://hal.science/hal-03540316.

[10] Y. Mashayekhi, B. Kang, J. Lijffijt, T. De Bie, Scalable job recommendation with lower congestion using optimal transport, IEEE Access 12 (2024) 55491–55505. doi:10.1109/ACCESS.2024.3390229.

[11] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140. doi:10.1007/BF00058655.

[12] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32. doi:10.1023/A:1010933404324.

[13] C. A. Pissarides, Equilibrium Unemployment Theory, 2 ed., MIT Press, Cambridge, MA, 2000.

[14] C. A. Pissarides, Short-run equilibrium dynamics of unemployment, vacancies, and real wages, The American Economic Review 75 (1985) 676–690. URL: https://www.jstor.org/stable/1821347.

[15] Y. Mashayekhi, N. Li, B. Kang, J. Lijffijt, T. De Bie, A challenge-based survey of e-recruitment recommendation systems, ACM Computing Surveys 56 (2024) 252:1–252:33. doi:10.1145/3659942. arXiv:2209.05112.

[16] S. Bian, X. Chen, W. X. Zhao, K. Zhou, Y. Hou, Y. Song, T. Zhang, J.-R. Wen, Learning to match jobs with resumes from sparse interaction data using multi-view co-teaching network, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), 2020, pp. 65–74. doi:10.1145/3340531.3411929. arXiv:2009.13299.

[17] C. Yang, Y. Hou, Y. Song, T. Zhang, J.-R. Wen, W. X. Zhao, Modeling two-way selection preference for person-job fit, in: Proceedings of the 16th ACM Conference on Recommender Systems (RecSys), 2022, pp. 102–112. doi:10.1145/3523227.3546752. arXiv:2208.08612.

[18] X. Yu, J. Zhang, Z. Yu, ConFit: Improving resume-job matching using data augmentation and contrastive learning, in: Proceedings of the 18th ACM Conference on Recommender Systems (RecSys), 2024, pp. 601–611. doi:10.1145/3640457.3688108. arXiv:2401.16349.

[19] P. Liu, H. Wei, X. Hou, J. Shen, S. He, K. Q. Shen, Z. Chen, F. Borisyuk, D. Hewlett, L. Wu, S. Veeraraghavan, A. Tsun, C. Jiang, W. Zhang, LinkSAGE: Optimizing job matching using graph neural networks, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2025, pp. 2448–2457. doi:10.1145/3690624.3709396. arXiv:2402.13430.

[20] X. Han, C. Zhu, X. Hu, C. Qin, X. Zhao, H. Zhu, Adapting job recommendations to user preference drift with behavioral-semantic fusion learning, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2024, pp. 1004–1015. doi:10.1145/3637528.3671759. arXiv:2407.00082.

[21] W. Jouanneau, M. Palyart, E. Jouffroy, Skill matching at scale: freelancer-project alignment for efficient multilingual candidate retrieval, in: RecSys in HR 2024

Workshop, in conjunction with ACM RecSys, 2024. arXiv:2409.12097.

[22] C. Yang, S. Dai, Y. Hou, W. X. Zhao, J. Xu, Y. Song, H. Zhu, Revisiting reciprocal recommender systems: Metrics, formulation, and method, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2024, pp. 3714–3723. doi:10.1145/3637528.3671734. arXiv:2408.09748.

[23] C. Rus, M. Mansoury, A. Yates, M. de Rijke, Joint modeling of candidate and recruiter preferences for fair two-sided job matching, in: Advances in Information Retrieval: 48th European Conference on Information Retrieval (ECIR), LNCS 16485, 2026, pp. 335–351. doi:10.1007/978-3-032-21324-2_27.

[24] G. Bied, P. Caillou, B. Crépon, C. Gaillac, E. Pérennes, M. Sebag, A job I like or a job I can get: Designing job recommender systems using field experiments, 2026. arXiv:2603.21699.

[25] R. K. Merton, The Matthew effect in science, Science 159 (1968) 56–63. doi:10.1126/science.159.3810.56.

[26] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, X. He, Bias and debias in recommender system: A survey and future directions, ACM Transactions on Information Systems (TOIS) 41 (2023) 1–39. doi:10.1145/3564284. arXiv:2010.03240.

[27] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, R. Burke, Feedback loop and bias amplification in recommender systems, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), 2020, pp. 2145–2148. doi:10.1145/3340531.3412152. arXiv:2007.13019.

[28] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, T. Joachims, Recommendations as treatments: Debiasing learning and evaluation, in: Proceedings of the 33rd International Conference on Machine Learning (ICML), volume 48 of *Proceedings of Machine Learning Research*, 2016, pp. 1670–1679.

[29] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, D. Jin, Disentangling user interest and conformity for recommendation with causal embedding, in: Proceedings of The Web Conference 2021 (WWW), 2021, pp. 2980–2991. doi:10.1145/3442381.3449788. arXiv:2006.11011.

[30] T. Wei, F. Feng, J. Chen, Z. Wu, J. Yi, X. He, Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2021, pp. 1791–1800. doi:10.1145/3447548.3467289. arXiv:2010.15363.

[31] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, Y. Zhang, Causal intervention for leveraging popularity bias in recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2021, pp. 11–20. doi:10.1145/3404835.3462875. arXiv:2105.06067.

[32] Z. Zhu, Y. He, X. Zhao, J. Caverlee, Popularity bias in dynamic recommendation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2021, pp. 2439–2449. doi:10.1145/3447548.3467376.

[33] A. Zhang, J. Zheng, X. Wang, Y. Yuan, T.-S. Chua, Invariant collaborative filtering to popularity distribution shift, in: Proceedings of the ACM Web Conference 2023 (WWW), 2023, pp. 1240–1251. doi:10.1145/3543507.3583461. arXiv:2302.05328.

[34] H. Zhou, H. Chen, J. Dong, D. Zha, C. Zhou, X. Huang, Adaptive popularity debiasing aggregator for graph collaborative filtering, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2023, pp. 7–17. doi:10.1145/3539618.3591635.

[35] W. Ning, R. Cheng, X. Yan, B. Kao, N. Huo, N. A. H. Haldar, B. Tang, Debiasing recommendation with personal popularity, in: Proceedings of the ACM Web Conference 2024 (WWW), 2024, pp. 3400–3409. doi:10.1145/3589334.3645421. arXiv:2402.07425.

[36] D. T. Mortensen, The matching process as a noncooperative bargaining game, in: J. J. McCall (Ed.), The Economics of Information and Uncertainty, University of Chicago Press, Chicago, 1982, pp. 233–258.

[37] X. Ren, L. Lü, R.-R. Liu, J. Zhang, Avoiding congestion in recommender systems, New Journal of Physics 16 (2014) 063057. doi:10.1088/1367-2630/16/6/063057.

[38] L. Boratto, G. Fenu, M. Marras, Connecting user and item perspectives in popularity debiasing for collaborative recommendation, Information Processing & Management (IPM) 58 (2021) 102387. doi:10.1016/j.ipm.2020.102387. arXiv:2006.04275.

[39] Y. Mashayekhi, B. Kang, J. Lijffijt, T. De Bie, ReCon: Reducing congestion in job recommendation using optimal transport, in: Proceedings of the 17th ACM Conference on Recommender Systems (RecSys), 2023, pp. 696–701. doi:10.1145/3604915.3608817. arXiv:2308.09516.

[40] S. Nathan, G. Bied, E. Pérennes, P. Caillou, B. Crépon, C. Gaillac, M. Sebag, JoLA: Job landscape aware job recommendation, in: Proceedings of the 5th Workshop on Recommender Systems for Human Resources (RecSys in HR 2025), co-located with the 19th ACM Conference on Recommender Systems, volume 4046 of *CEUR Workshop Proceedings*, Prague, Czech Republic, 2025. URL: https://ceur-ws.org/Vol-4046/RecSysHR2025-paper_6.pdf.

[41] G. Bied, E. Pérennes, S. Nathan, V. A. Naya, P. Caillou, B. Crépon, C. Gaillac, M. Sebag, RECTO: REcommandation diminuant la congestion par transport optimal, in: Actes de la 9e Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle (APIA), PFIA 2023, Strasbourg, France, 2023, pp. 89–98. URL: https://hal.science/hal-04158566.

[42] Y. Li, M. Yamashita, H. Chen, D. Lee, Y. Zhang, Fairness in job recommendation under quantity constraints, in: AAAI-23 Workshop on AI for Web Advertising, 2023.

[43] A. Singh, T. Joachims, Fairness of exposure in rankings, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2018, pp. 2219–2228. doi:10.1145/3219819.3220088. arXiv:1802.07281.

[44] G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, A. Chakraborty, FairRec: Two-sided fairness for personalized recommendations in two-sided platforms, in: Proceedings of The Web Conference 2020 (WWW), 2020, pp. 1194–1204. doi:10.1145/3366423.3380196. arXiv:2002.10764.

[45] T. Joachims, Fairness and control of exposure in two-sided markets, in: Proceedings of the 2021 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR), 2021. doi:10.1145/3471158.3472226.

[46] N. Li, B. Kang, J. Lijffijt, T. De Bie, FEIR: Quantifying and reducing envy and inferiority for fair recommendation of limited resources, ACM Transactions on Intelligent Systems and Technology (TIST) 15 (2024) 80:1–80:24. doi:10.1145/3643891. arXiv:2311.04542.

[47] R. Shimer, The assignment of workers to jobs in an economy with coordination frictions, Journal of Political Economy 113 (2005) 996–1025. doi:10.1086/444551.

[48] Y. Su, M. Bayoumi, T. Joachims, Optimizing rankings for recommendation in matching markets, in: Proceedings of the ACM Web Conference 2022 (WWW), 2022, pp. 328–338. doi:10.1145/3485447.3511961. arXiv:2106.01941.

[49] E. R. Moen, Competitive search equilibrium, Journal of Political Economy 105 (1997) 385–411. doi:10.1086/262077.

[50] N. Arnosti, R. Johari, Y. Kanoria, Managing congestion in matching markets, Manufacturing & Service Operations Management (M&SOM) 23 (2021) 620–636. doi:10.1287/msom.2020.0927.

[51] J. J. Horton, S. Vasserman, M. Watt, Reducing congestion in labor markets: A case study in simple market design, 2024. URL: https://shoshanavasserman.com/files/2024/12/HVW_nov24.pdf.

[52] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: A survey and categorisation, Information Fusion 6 (2005) 5–20. doi:10.1016/j.inffus.2004.04.004.

[53] R. E. Schapire, The strength of weak learnability, Machine Learning 5 (1990) 197–227. doi:10.1007/BF00116037.

[54] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1997) 119–139. doi:10.1006/jcss.1997.1504.

[55] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259. doi:10.1016/S0893-6080(05)80023-1.

[56] L. Breiman, Stacked regressions, Machine Learning 24 (1996) 49–64. doi:10.1007/BF00117832.

[57] M. Jahrer, A. Töscher, R. Legenstein, Combining predictions for accurate recommender systems, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2010, pp. 693–702. doi:10.1145/1835804.1835893.

[58] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proceedings of the 10th International Conference on World Wide Web (WWW), 2001, pp. 613–622. doi:10.1145/371920.372165.

[59] J. A. Aslam, M. Montague, Models for metasearch, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2001, pp. 276–284. doi:10.1145/383952.384007.

[60] M. Montague, J. A. Aslam, Condorcet fusion for improved retrieval, in: Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM), 2002, pp. 538–548. doi:10.1145/584792.584881.

[61] C. J. C. Burges, From RankNet to LambdaRank to LambdaMART: An Overview, Technical Report MSR-TR-2010-82, Microsoft Research, 2010. URL: https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/.

[62] C. Bernardis, Multi-stage ensemble model for cross-market recommendation, 2022. arXiv:2202.08824, arXiv preprint; PolimiRank team, WSDM Cup 2022 (Cross-Market Recommendation), ranked 4th.

[63] B. Xia, Z. Yu, Z. Zhu, H. Sun, B. Han, J. Wang, R. Liu, W. Ou, Rethinking multi-objective ranking ensemble in recommender system: From score fusion to rank consistency, 2026. arXiv:2601.02955.

[64] Z. Xu, Z. Yang, Z. Guo, S. Liu, L. Lin, X. Liu, Y. Liu, H. Li, UMRE: A unified monotonic transformation for ranking ensemble in recommender systems, 2025. arXiv:2508.07613.

[65] W. Yu, B. Liu, B. Xia, X. Xu, Y. Chen, Y. Li, L. Hu, Unsupervised ranking ensemble model for recommendation, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2024, pp. 6181–6189. doi:10.1145/3637528.3671598.

[66] Q. Dong, Q. Yuan, Y.-B. Shi, Alleviating the recommendation bias via rank aggregation, Physica A: Statistical Mechanics and its Applications 534 (2019) 122073. doi:10.1016/j.physa.2019.122073.

[67] J. A. Aledo, C. Domínguez, J. d. D. Jaime-Alcántara, M. Landete, Optimizing weak orders via integer linear programming, 2025. arXiv:2511.19345.

[68] H. Bonab, M. Aliannejadi, A. Vardasbi, E. Kanoulas, J. Allan, Cross-market product recommendation, in: Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021, pp. 110–119. doi:10.1145/3459637.3482493. arXiv:2109.05929.

[69] C. Wang, Z. Fan, L. Yang, M. Yang, X. Liu, Z. Liu, P. S. Yu, Pre-training with transferable attention for addressing market shifts in cross-market sequential recommendation, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2024, pp. 2970–2979. doi:10.1145/3637528.3671698.

[70] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine Learning 79 (2010) 151–175. doi:10.1007/s10994-009-5152-4.

[71] J. Huang, H. Oosterhuis, M. Mansoury, H. van Hoof, M. de Rijke, Going beyond popularity and positivity bias: Correcting for multifactorial bias in recommender systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2024, pp. 416–426. doi:10.1145/3626772.3657749. arXiv:2404.18640.

[72] G. E. P. Box, D. R. Cox, An analysis of transformations, Journal of the Royal Statistical Society: Series B (Methodological) 26 (1964) 211–243. doi:10.1111/j.2517-6161.1964.tb00553.x.

[73] CareerBuilder, Job recommendation challenge, Kaggle competition, 2012. URL: https://www.kaggle.com/competitions/job-recommendation, CareerBuilder

job/application dataset; source of the CareerBuilder-Large benchmark extract.
[74] C. Gini, Variabilità e Mutabilità: Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche, Tipografia di Paolo Cuppini, Bologna, Italy, 1912.
[75] M. Elahi, D. K. Kholgh, M. S. Kiarostami, S. Saghari, S. P. Rad, M. Tkalčič, Investigating the impact of recommender systems on user-based and item-based popularity bias, Information Processing & Management 58 (2021) 102655. doi:10.1016/j.ipm.2021.102655.
[76] V. Pareto, Manuale di economia politica, Società Editrice Libraria, Milano, 1906.

A. Supplementary Material

A.1. Sensitivity w.r.t α in CB-L

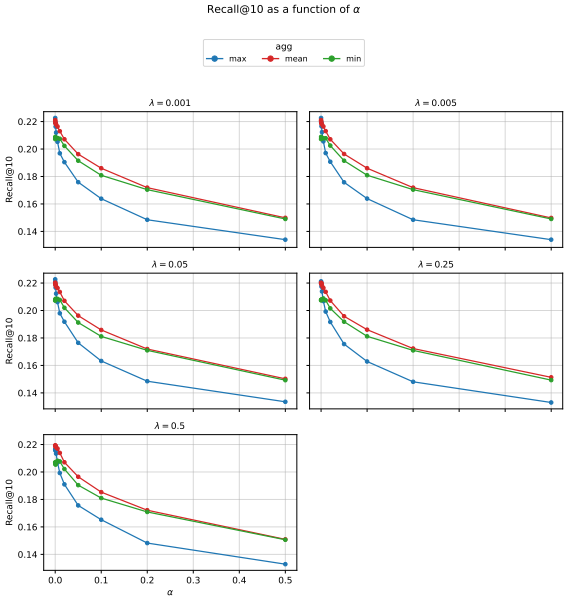

Figure 9: DISPERSE on CB-L: Sensitivity of Recall@10 w.r.t. α, for different aggregation modes and $\lambda \in [.001, .005, .05, .25, .5]$ (from top-left to bottom-right sub-images).

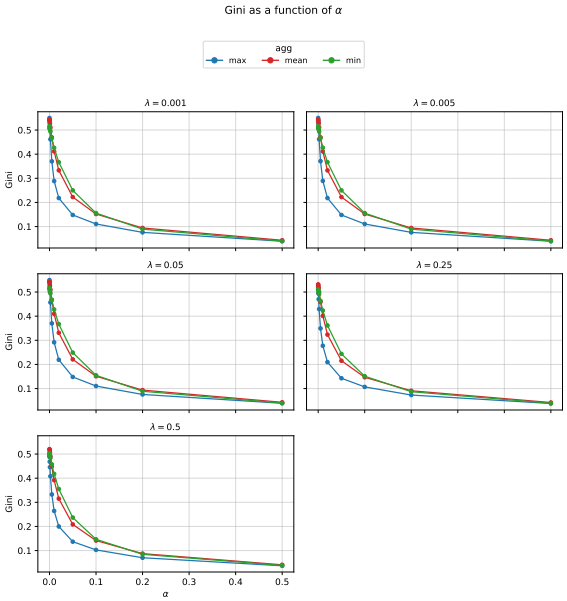

Figure 10: DISPERSE on CB-L: Sensitivity of Gini index w.r.t. α, for different aggregation modes and $\lambda \in [.001, .005, .05, .25, .5]$ (from top-left to bottom-right sub-images).

A.2. Sensitivity w.r.t α in BFC

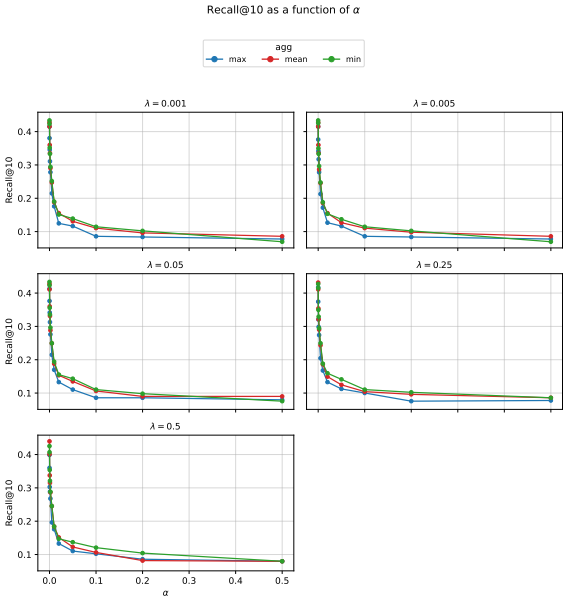

Figure 11: DISPERSE on BFC: Sensitivity of Recall@10 w.r.t. α, for different aggregation modes and $\lambda \in [.001, .005, .05, .25, .5]$ (from top-left to bottom-right sub-images).

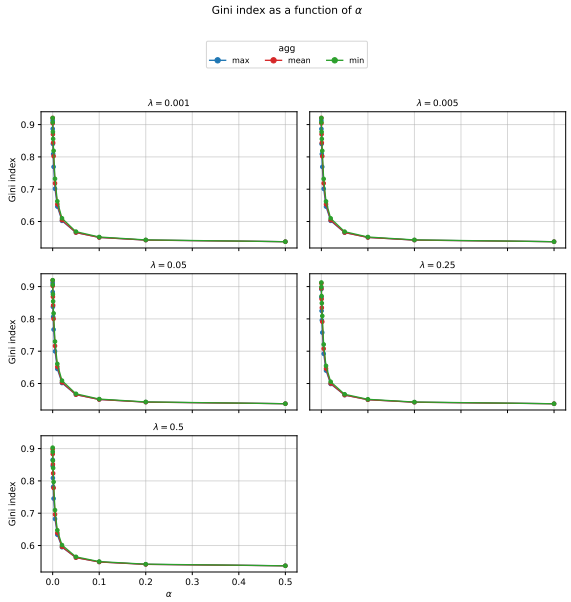

Figure 12: DISPERSE on BFC: Sensitivity of Gini index w.r.t. α, for different aggregation modes and $\lambda \in [.001, .005, .05, .25, .5]$ (from top-left to bottom-right sub-images).