Congestion-Avoiding Job Recommendation with Optimal Transport

G. Bied, E. Perennes, V. Alfonso, P. Caillou, B. Crépon, C. Gaillac, M. Sebag

Introduction

- Motivation: **recommending job ads to job seekers** in the context of a public employment agency
- Issue of ***congestion***: it is inefficient to recommend the same job ad to many job seekers
- Contribution: new algorithm coupling optimal transport with recommender systems, named *Congestion-Avoiding job Recommendation with Optimal Transport* (CAROT)
- CAROT starts with recommendations scores computed to maximize individual performance (chance to find a job learned from past hiring). It uses OT to flatten the score matrix to reduce the congestion problem.

Related Work

Computational Optimal Transport (OT)

- Aims to map uniform discrete distribution on n users μ onto the uniform discrete distribution on m job ads ν
- Let $\Gamma(\mu,\nu)$ be the set of measures s.t. their marginals wrt 1st and 2nd arguments are μ and ν, $C_{i,j}$ be the cost of mapping i onto j, the OT problem is:

$$\min_{\gamma\in\Gamma(\mu,\nu)} \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{i,j} C_{i,j} \tag{1}$$

Cuturi (2013): tractable relaxation with an entropic term:

$$\min_{\gamma\in\Gamma(\mu,\nu)} \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{i,j}\left(C_{i,j} + \varepsilon \log(\gamma_{i,j})\right) \tag{2}$$

with ε a regularization weight.

Congestion-avoiding recommendation

- Issue of congestion noted early by Gualdi et al. (2013)
- Xia et al. (2019): (NP-hard) multi-objective problem
- Borisyuk et al. (2017): decentralised score shifting based on predicted popularity
- Chen et al. (2019), Li et al. (2019), and related work in econometrics - Chiappori et al. (2016): OT-based approaches
- Unlike main approaches in OT-based recommendation, CAROT does *not* assume the observed matches (the training data) to be the solution of an OT plan

Proposed approach

Step 1: Learning a recommender system

- Two baseline recommender systems are considered:
  - XGBoost (XGB): predict whether a pair matches or not using boosted trees - Chen et al. (2016), Volkovs et al. (2017)
  - Neural networks (NN) mapping descriptions of user x_i and item y_j onto latent spaces $z_{x,i}$ and $z_{y,j}$, with adequacy s_{ij} sought as $z_{x,i}^T A z_{y,j}$ with A a matrix - Chechik et al. (2009). The mappings and A are learned end-to-end using a triplet loss - Weinberger et al. (2009).
- Both models yield a score s_{ij} to rank job ads j for job seeker i

Step 2: Finding a transport plan

- Transform scores / rankings into a cost $C_{ij} = g(s_{ij})$ of matching i and j, with g monotonous function (hyperparameter)
  - Below: g ="$Id+$": essentially linear in scores; g ="$ndcg$": NDCG-like criterion based on rank of j for i
- Solve regularized OT problem, i.e. equation 2

Issuing recommendations

- For a given job seeker i, sort job ads by γ_{ij} (decreasing order)

Data

Proprietary data provided by *Pôle emploi*, the French public unemployment agency.

- Training set: circa 1,650,000 job seekers, 477,000 job ads, 43,000 matches (signed contracts) during the Feb.-Oct 2018 period.
- The representation of job seekers (resp. job ads) is of dimension 448 (resp. 582).
- Test set: job seekers and job ads in the sector of transportation and logistics - circa 110,000 job seekers, 14,200 job ads and 450 matches in Nov. 2018.

Setup: s_{ij} is learned from the training set, the OT plan computed on the test set, the performance indicators are measured on the test set.

Performance criteria

- Recall@k: fraction of users for which the actually preferred item is ranked among the top-k recommendations
- Coverage@k: share of job ads appearing in top-k recommendations to the population of job seekers.
- Congestion@k: KL-type divergence between observed and uniform market shares, normalized to be best (no congestion) at -1 and worst at 0

Results

	Algorithm	Recall (%)		Coverage (%)		Congestion	
	(g,ε)	@1	@10	@1	@10	@1	@10
	Random	0	0.21	99.95	100	-0.99	-0.99
	XGB (no OT)	9.62	31.40	12.94	25.16	-0.62	-0.64
CAROT-XGB	$Id+$,1.0	4.81	21.99	21.61	31.76	-0.74	-0.75
CAROT-XGB	$Id+$,0.1	2.18	15.31	27.54	41.24	-0.78	-0.81
CAROT-XGB	$Id+$,0.01	4.37	20.45	46.75	57.61	-0.85	-0.79
CAROT-XGB	$ndcg$,1.0	9.62	31.61	12.96	26.14	-0.62	-0.67
CAROT-XGB	$ndcg$,0.1	8.97	25.38	14.69	30.84	-0.67	-0.74
CAROT-XGB	$ndcg$,0.01	5.03	14.00	36.81	57.52	-0.82	-0.81
	NN (no OT)	5.68	28.66	6.02	17.78	-0.46	-0.49
CAROT-NN	$Id+$,1.0	6.78	26.14	11.99	26.30	-0.62	-0.65
CAROT-NN	$Id+$,0.1	2.40	19.03	28.23	40.16	-0.80	-0.79
CAROT-NN	$Id+$,0.01	3.93	16.30	27.89	62.35	-0.83	-0.70
CAROT-NN	$ndcg$,1.0	5.68	27.46	6.02	19.75	-0.46	-0.55
CAROT-NN	$ndcg$,0.1	5.25	23.3	8.85	26.40	-0.53	-0.65
CAROT-NN	$ndcg$,0.01	1.53	12.36	35.41	51.56	-0.81	-0.81

- Without OT, less than 18% (NN) or 25.16% (XGB) of job ads would appear in top-10 recommendations, despite job seekers outnumbering job ads by a factor of 8 in the test set
- NN is dominated by XGB for all indicators, but NN is 50 times faster when computing recommendations (and twice faster to train)
- Coverage monotonically increases, and recall monotonically decreases as ε decreases from 1 to .01: it seems hard to combine good coverage and decent recall.
- Congestion@1 can be significantly improved (from -.62 to .78) at the expense of a moderate recall loss (recall@10 goes from 21% to 15.3%) for $g = Id+, \varepsilon = .1$.
- Surprisingly, decreasing ε yields a better (lower) congestion at the expense of a worse recall.