AI for Good
Two directions

Michèle Sebag

TAU, CNRS – INRIA – LISN, U. Paris-Saclay

ANITI Days
Nov. 16th, 2023

AI in the field. I. AI for good

The environment: GAFAM

- An AI *niche* for academics ?
  Choose Your Weapon: Survival Strategies for Depressed AI Academics
  Julian Togelius and Georgios N. Yannakakis (2023)
  https://arxiv.org/pdf/2304.06035.pdf

Example: Toward recommending a job for all

- Coll. Pole Emploi & ENSAE-CREST
- Evaluation campaigns

More:

Guillaume Bied et al.: Toward Job Recommendation for All. IJCAI 2023: 5906-5914

The context: energy-hungry AI

- Data beat algorithms
- Resisting the "More is Better" motto
  Green AI.
  Roy Schwartz, Jesse Dodge, Noah A. Smith, Oren Etzioni, (2019)
  https://arxiv.org/pdf/1907.10597.pdf

Example: Meta learning & adapting ML hyper-parameters

- Few, expensive meta-examples (OpenML repository)
- Designing meta-features

More:

Herilalaina Rakotoarison et al: Learning meta-features for AutoML. ICLR 2022

Position of the problem

AI for Social Good: Reducing unemployment

- UN Sustainable Development Goals:
  - Goal 8: Decent work and Economics Growth
  - Goal 10: Reduced Inequalities

Reducing frictional unemployment

- By reducing search costs and suggesting non-obvious opportunities at low marginal cost
- Growing literature in economics: Belot et al. (2019); Altmann et al. (2023); Behaghel et al. (2023); Le Barbanchon et al. (2023)

Highly consequential application of Machine Learning:

- Jobs determine livelihoods and social positions
- "High-risk" according to forthcoming European AI Act

Why, when, how...

How this all started

- Yet another PhD founding a start-up to optimize ad banners ! (2010)
- There should be a real problem with same algorithmic challenges...
- Recommending jobs ? T. Schmitt's PhD (2014-2018) ISN grant
- Collaboration with ENSAE and Pole Emploi: VADORE (2018-now) Dataia grant
- First campaign with Pole Emploi (2022); second (2023).

State of art

Related Work

- Expert systems, *e.g.* WCC ELISE (SDR@PE)
- Collaborative filtering — Bell et al., 2007 (Netflix prize)
- 2016 & 2017 RecSys challenges on job data
  Xiao et al., 2016; Volkovs et al., 2017
- e-recruitment systems based on proprietary data
  Kenthapadi et al., 2017 (LinkedIn)
  Zhao et al., 2021 (CareerBuilder)

The specifics of VADORE

Objectives

- Design a Job Recommender System for Job seekers
- Based on Pole Emploi proprietary data
- ... that scales up (400,000 job seekers in a region)

Two challenges

- Sparsity of interaction matrix
  Phase 1: only signed contracts (sparsity 99.5 %)
  Phase 2: also applications
- Build a service **for all**
  mostly minimum wages; small signal to noise ratio

Highly Sensitive and Complex Data

- Source: Pôle emploi
- **Scope:** Auvergne-Rhône-Alpes region (France); 2019-mid 2022
- **Job seekers** (1.2M)
  - **Qualification**: experience, education, skills, driver's licence, languages, means of transportation, occupation
  - **"Preferences"**: contract, full-time status, commuting time, working hours, reservation wage
  - **Other**: textual information (CV), socio-demographic variables, past employment history, accompaniment by the PES
- **Job ads** (2.2M)
  - Job and firm description (text), occupation, requirements (skills, education), contract, labor conditions
- **Labor market interactions**
  - Hires (285k) monitored by the PES
  - Applications (1.3M)

MUSE: Multi-head Sparse E-Recruitment

A three tier architecture

- 1st tier, Muse.0
  fast; serves to filter most promising (top 1,000) job ads x for each job seeker y

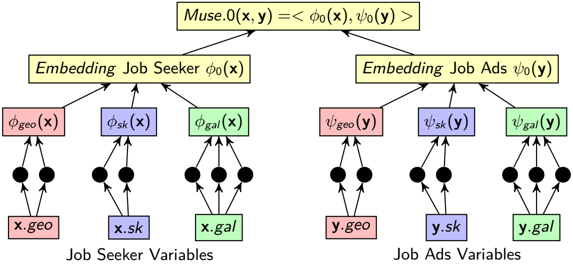
- 2nd tier, Muse.1
  thanks to filter, can consider features $f(x, y)$
  (e.g. distance; skill match)
  Two heads:
  - Muse.1.Hire (trained from contracts)
  - Muse.1.App (trained from applications)

- 3rd tier, Muse.2
  builds on the top of Muse.1.Hire and Muse.1.App

Overview of Muse.0

Three modules

- Geographic
- Skills (11,000 skills in ontology)
- General



Overview of Muse.0, follow'd

Triplet loss

$$\text{Loss} = \sum_{x,y,y'} [s(x,y') - s(x,y) + m]_+$$

with

- job seeker x hired on job ad y
- y' another ad, (uniform in same week as y)
- margin $m = 1$

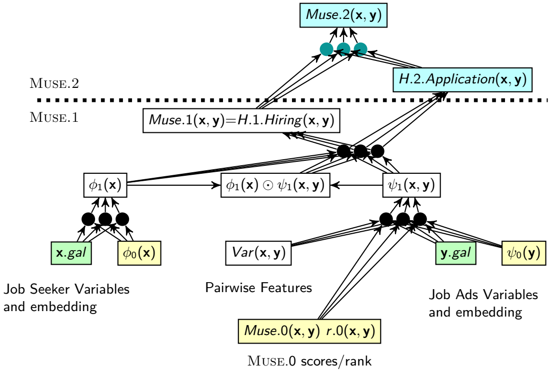
Role

- Fast inference
- Filtering top 1,000 job ads y for each x
- Enabling more expensive feature construction on 2nd tier.

Muse.1 and Muse.2

Goals

- Re-rank the top 1,000 ads selected by the first tier, using more sophisticated attributes (*e.g.* geographic distance; matching salary;)
- Muse.1.Hire: trained from hirings (signed contracts)
- Muse.1.App: trained from applications.
- Muse.2: on top of both, trained from hirings
- All: triplet loss.



Results. I. Public data

- Baseline: XGBoost based on RecSys 2017 Challenge winner
  Volkovs et al., 2017
- Perf. indicator: Recall@k (fraction of x s.t. y is ranked in top-k)

Recall@	XGBoost	Muse.0	Muse.2
10	26.83	22.88	**30.1***
20	35.59	31.55	**40.2***
100	58.88	53.80	**63.2***
1000	**86.47***	82.13	-
Training time (hours)	1.83	7.7	1.25
Recommendation time (seconds)	1.4	0.0004	0.02

Comparative results of Muse and XGBoost: recall, overall training time and recommendation time *per* job seeker.[1]

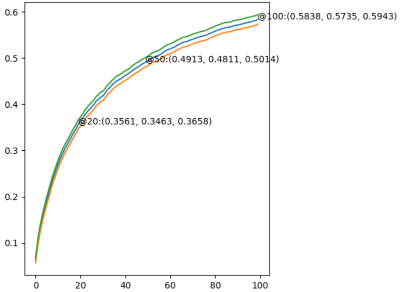
Muse: decent Scalability and Recall

[1] Computational times measured on Intel® Xeon® Silver 4214Y CPU @ 2.20GHz, with 187 GB RAM and a Tesla T4 GPU.

Results II. Pole Emploi data

In the lab

Train data: 85% weeks, Jan. 2019 - Sept 2022



Complementary of the modules

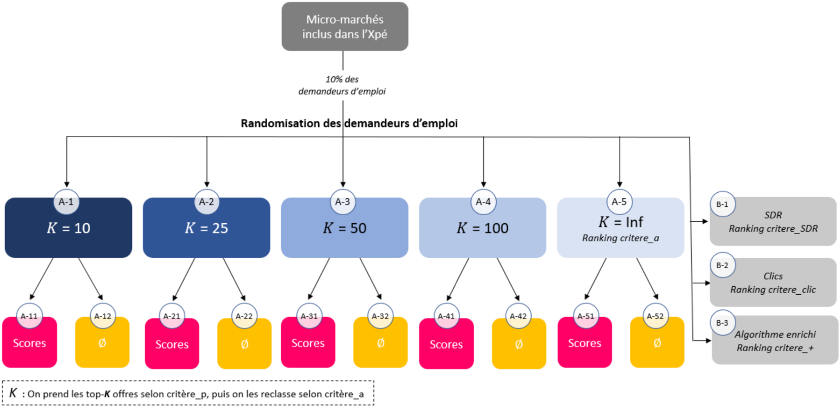
	Single module			All modules but one			Muse.0
R@	M_{geo}	M_{Gal}	M_{sk}	M_{geo}	M_{Gal}	M_{sk}	
100	15.43	34.79	4.80	39.97	47.28	51.96	53.80

Table: MUSE.0: Impact of the three geographical, skills and general modules on the recall@100 through ablation studies. Left: module standalone. Right: MUSE.0 with all modules but one.

Campaigns (March 2022; June 2023)

Evaluation in the field: Recall is not the main thing for PE...

- Check whether recommendations are well accepted
- Identify recommendations that are inappropriate
- Assess combinations of Muse and SDR@PE
  Mix: rank top-k (Muse) after SDR
- Assess impact of interface (neutral; encouraging)



Campaigns (March 2022; June 2023)

Feedback

- Same critiques for all variants (this job is too far; I changed my preferences; this job is not for me, I don't have driving licence)
- When neutral interface, most appreciated (significantly so) variant: mixture of SDR and Muse;
- When "encouraging" interface: no significant difference among variants (and satisfaction significantly decreased).

The Issue of Gender Biases

Recommender systems trained on real-world data may learn job seekers' and recruiters' biases

The Washington Post

The Intersect

Google's algorithm shows prestigious job ads to men, but not to women. Here's why that should worry you.

RETAIL OCTOBER 11, 2018 / 1:04 AM / UPDATED 4 YEARS AGO

Amazon scraps secret AI recruiting tool that showed bias against women

By Jeffrey Dastin 8 MIN READ

SAN FRANCISCO (Reuters) - Amazon.com Inc's AMZN.O machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

- Left: Google Ads - Washington Post — Datta et al, 2015
  NB: Same hold for Facebook ad delivery — Ali et al., 2019
- Right: Reuters - Amazon.

Gender gaps

In data: due to

- Job seeker behavior (applications):
  - Gendered differences in **assessment of success likelihood** (over or under-confidence) & **risk aversion** Cortés et al., 2022
  - Gendered **valuation of job ad characteristics**, e.g. occupation, wage vs. commute Le Barbanchon et al., 2021
- Recruiter side: gendered treatment of applications Arnoult et al., 2021

In recommendations

- Differences are more or less acceptable (depends)
- Issues:
  - Legal issues
  - Illegitimate wrt common fairness definitions or perceptions Pierson et al., 2017
  - Trust in recommendations and in the institution
  - Perpetuation of gender stereotypes
  - Downstream effects on e.g. the gender wage gap (with effects on pensions, intra-household bargaining ...)

Related work

Algorithmic fairness

- Immense literature, mostly on binary classification & decision-making
  Survey: Mehrabi et al. (2019)
- Growing one on recommender systems
  Survey: Ekstrand et al. (2021)
- ... for the labor market — Survey: Kumar et al. (2023)

Audit studies of job recommender systems

Kuhn & Zhang, WP

Gender bias: questions and modelling

Questions

- Is recommendation performance different for men and women?
  - **Measure**: recall@k
- Are different jobs shown to women and men? :
  - Wage, distance, executive status, contract type, working hours, male-dominated occupation
  - Fit to job seeker's search criteria (average fit w.r.t. distance / occupation / wage / contract / working hours)

Model

- Gender G (=1 if woman)
- Outcome Y: characteristics of algorithm's top-1 recommendation (e.g. wage)
- Naive average gender effect (AGE):

$$\delta = \mathbb{E}[Y|G = 1] - \mathbb{E}[Y|G = 0]$$

- AGE controlled w.r.t.:
  - **Qualifications**: experience, education, skills, driver's licence, languages, means of transportation, occupation
  - Both **qualifications** and "**preferences**": contract, full-time status, commuting time, working hours, reservation wage

Average Gender Effect

- X: covariate, job seeker characteristics
- $Z \subset X$: controls (qualifications, or qualifications + preferences)
- Model inspired from potential outcome formalization Robinson 88

$$Y = E[Y|X] + (T - E[T]) \times \tau(X) + \varepsilon$$

- using a partially linear regression model:

$$Y = \tau G + \mu_0(Z) + \varepsilon, \quad E(\varepsilon|Z, G) = 0$$

  where:
  - τ: parameter of interest (gender difference unexplained by Z)
  - $\mu_0(Z)$: nuisance function (valuation of Z in terms of Y for men)
- Main condition: common support (job seekers must be sufficiently comparable in terms of Z)
- τ is estimated using *Double Machine Learning*

Chernozhukov et al., 2018

Results

Comparing with hirings

- MUSE trained on hiring data: does it worsen the actual M/W gaps ?

Comparing with applications

- No impact of recruiter's prejudices (apart from anticipations)
- Applications reflect jobseekers' expected utility

Measured as

$$(Y^{data} - Y^{rec}) = \tau G + \mu_0(Z) + \varepsilon$$

- Y^{rec}: characteristic of recommended job
- Y^{data}: characteristic of hire / application

Recommendation performance

Top k	Recall@k	Men	Women	p-value
10	0.256	0.243	0.267	0.000
20	0.351	0.333	0.366	0.000
100	0.590	0.576	0.603	0.000

Recall higher for W than for M (3.3 points for recall@20)

- Remains significant when controlled for:
  - Qualifications + Preferences: (2.6 points, $p < 0.0001$)
  - Qualifications + Preferences + Distance to job: (2.3 points, $p = 0.001$)
- Why ?
  - Data imbalance ? (W = 54% of hires in training set)
    No: gap remains significant when downsampling
  - Tentative interpretation: W's preference for near-by job ads.

Average Gender Effects on characteristics

	$\hat{\delta}$ (Pop.)	$\hat{\delta}$ (Overlap)	$\hat{\tau}_Q$	$\hat{\tau}_{QP}$
Wage (log)	-0.023***	-0.016***	-0.007***	-0.004***
Distance (km)	-0.474***	-0.231***	0.077*	0.117***
Executive position	-0.004***	-0.009***	-0.004***	-0.001
Long-term contract	-0.040***	-0.034***	-0.016***	-0.012***
Male-dominated job	-0.411***	-0.219***	-0.033***	-0.033***
Hours worked	-2.934***	-1.957***	-0.684***	-0.409***
Fit to search param.	-0.028***	-0.019***	-0.013***	-0.011***

Notes: Results for $n = 228,625$ job seekers. $\hat{\delta}$: difference in means; $\hat{\delta}$ (overlap): difference in means for individuals w/ propensity $\in [0.05, 0.95]$; $\hat{\tau}_Q$: DML estimator when controlling for qualifications; $\hat{\tau}_{QP}$: DML estimator when controlling for preferences and preferences.

Summary

- Average Gender Effects:
  - Less paid (2.3 pp), less often in executive positions, less often in male-dominated occupations ...
- Gaps reduced but still significant:
  when controlled for qualifications Q
  when controlled for qualifications and preferences QP

Average Gender Effect: Recommendations vs Hirings

Hires	Qualifications			Qualifications & Preferences		
	τ_Q (Hire)	τ_Q (Rec.)	τ_Q (Difference)	τ_{QP} (Hire)	τ_{QP} (Rec.)	τ_{QP} (Difference)
Wage (log)	-0.012***	-0.007***	0.004	-0.010***	-0.005**	0.005*
Distance (km)	-0.935	0.344***	1.479*	-0.654	0.391**	1.109
Executive position	-0.007**	-0.003	0.004	-0.006*	-0.002	0.003
Long-term contract	-0.035***	-0.025***	0.010	-0.033***	-0.024**	0.010
Male-dominated job	-0.141***	-0.053***	0.086***	-0.141***	-0.055***	0.085***
Hours worked	-1.435***	-0.934***	0.479***	-1.286***	-0.715***	0.508***
Fit to search param.	-0.021***	-0.020 ***	0.002	-0.021***	-0.019***	0.003

Notes: Results for hired job seekers satisfying the overlap condition ($n = 25,783$). DML estimators for gender gaps in hires, recommendations, and the hire-recommendation differences. Col. 1-3 present results when controlling for qualifications (τ_Q), col. 4-6 results controlling for qualifications and preferences (τ_{QP}).

Summary

- Same gaps as in hirings
- If anything, gaps are reduced (wage, hours worked, male-dominated occupations) in recommendations

Average Gender Effect: Recommendations vs Applications

Applications	Qualifications			Qualifications & Preferences		
	τ_Q (App.)	τ_Q (rec.)	Diff. of Diff.	τ_{QP} (App.)	τ_{QP} (rec.)	Diff. of Diff.
Wage (log)	-0.013***	-0.005**	0.008***	-0.011***	-0.003*	0.007***
Distance (km)	-5.721***	0.081	5.962***	-4.326***	0.136	4.555***
Executive position	-0.006***	-0.001	0.004	-0.004**	-0.000	0.003
Long-term contract	-0.033***	-0.025***	0.010	-0.029***	-0.019**	0.011
Male-dominated job	-0.115***	-0.060***	0.056***	-0.115***	-0.059***	0.058***
Hours worked	-1.410***	-0.763***	0.668***	-1.126***	-0.486***	0.651***
Fit to search param.	-0.023***	-0.016***	0.008***	-0.020***	-0.016***	0.005*

Notes: Results for applications of jobseekers satisfying the overlap condition. DML estimators for gender gaps in applications, recommendations, and the application-recommendation differences. Col. 1-3 present results when controlling for qualifications (τ_Q), col. 4-6 results controlling for qualifications and preferences (τ_{QP}).

Summary

- Similar gaps as in Hirings
- Recommendations do not amplify the gaps

Partial Conclusion

Lessons learned

- Choice of features very informative (ML vs economics)
- Muse: Performance ok wrt scalability and wrt recall (in the lab)
- Biases: they exist; in data, in recommendations.
- Adversarial approaches: suppress bias, at the expense of recall

Perspectives

- Addressing biases: toward recommending a set of job ads
- The real performance indicator: i) decreasing time-to-job; ii) quality of found job.
- Muse $\rightarrow$ a subscription service (also recommended by France Travail).
- Adapting Muse for recruiters

Where we are

Toward a Job for All

- Overview
- Results
- Fairness
- Partial conclusion

Affordable ML with Meta-learning

- Meta-features for tabular data
- Experimental validation
- What did we learn

The irresistible AI/ML Wave

Hardly affordable: Computer vision, Games, NLP...

AI $\neq$ GAMA

- Cost of ML: *(...) GPT-3 could have easily cost 10 million dollars to train.*
- Wanted: **Affordable AI**

Control layer in algorithmic platforms

In some domains

- No Free Lunch — Wolpert & Macready, 97
- No killer algorithm $\Rightarrow$ Algorithm portfolios / Many options
- Algorithm performance governed by hyper-parameter values

Hyper-parameter tuning: a critical task

- In constraint programming — Rice 76
- In stochastic optimization — Grefenstette 87
- In machine learning (meta-learning) — Bradzil et al. 93

Crossing the chasm: software life beyond research labs

- Automatically adjust algorithm parameters depending on current problem
- Select best (expected) algorithm depending on current problem

Meta-Learning

Position of the problem

An optimization problem

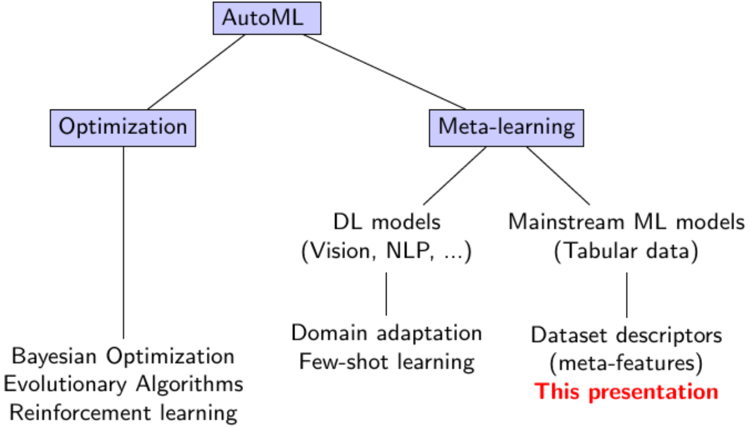
- Black-Box optimization of $\mathcal{L}$, with
- $\mathcal{L}$: expensive objective function
- Θ (hyper-parameter space): Mixed discrete and continuous search space

International Challenges

- CP & CSP: ASLib challenge — Bischle et al. 16
- Open Algorithm Selection Challenge — Lindauer et al. 17
- AutoML challenge — Guyon et al., 15-16
- AutoDL challenge — Guyon et al. 19-21

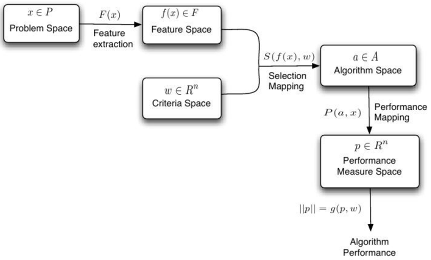
Automated Machine Learning: Methods, Systems, Challenges

Hutter, Kotthoff & Vanschoren, 19



Meta-Learning

Rice 76



Learn a performance model

- Gather problem instances (benchmark suite)
- Design descriptive features for pb instances
- Run algorithms on pb instances
- Build meta-training set:

$$\mathcal{E}_j = \{(\text{desc. } x_i \text{ of } i\text{-th pb instance, perf. of } j\text{-th algo})\}$$

- Learn performance model $\widehat{\mathcal{F}}_j$ from $\mathcal{E}_j$
- Decision making: for pb $\mathbf{x}$

$$\text{Select Algo } j^* = \arg\max_j \left\{ \widehat{\mathcal{F}}_j(\mathbf{x}) \right\}$$

Hand-crafted meta-features for tabular data

Hand-crafted meta-features

Alcobaca et al. 20, Rivolli et al. 22

- shallow m.f: number of instances, number of classes
- statistical m.f.: entropy, average mutual information of features with target
- landmarks: performance of inexpensive classifiers (e.g., Decision Tree)

Pfahringer et al. 00

Ex: Auto-sklearn meta-features

Feurer et al., 2015

Number of features	Number of features with missing values
Ratio numerical to nominal	Mean of categorical feature symbols
Mean of feature kurtosis coefficients	Dataset ratio
Mean of feature skewness	Mean of class probabilities

Limitation: Meta-features

- (meant to) Capture a distribution: the dataset
- (must be) Inexpensive
- (currently) Insufficiently expressive to capture dataset similarity w.r.t. AutoML

Meta-features for tabular data

Given

- An ML algorithm/pipeline SVM, RF, AutoSkLearn, ...
- Its configuration space $\Theta \subset \mathbb{R}^d$
- A set of benchmark problems $A, B, C, \ldots$

We have

- Basic representation: (available for all datasets)

$$A \rightarrow x_A \in \mathbb{R}^D$$

- Target representation: (available for benchmark datasets)

$$A \rightarrow z_A \subset \Theta$$

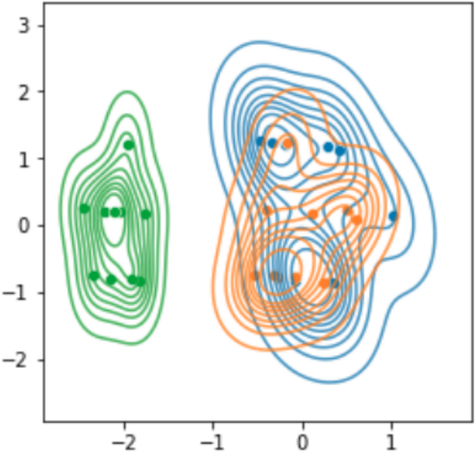
set of configurations reaching top performance on A

We want

- A good metric on the set of datasets: such that the top configurations of the nearest neighbors of A yield good performance on A.

Hand-crafted meta-features do not reflect target metrics

- Datasets A, B, and C
- x_C is the nearest neighbor of x_A in Θ (Euclidean distance)
- z_B is the nearest neighbor of z_A in 2^Θ (Wasserstein distance)



z_A, z_B, z_C in a 2d projection of Θ

Metabu: Learning meta-features for tabular data

Rakotoarison et al. 22

Principle

- Map the basic representation onto a learned representation ***with same metric as the target representation***

MetaBu Algorithm

1. Given benchmark data $A, B, C, \ldots$ with target representation $z_A, z_B, z_C, \ldots$
2. Find intermediate representation $y_A, y_B, y_C, \ldots$ in $\mathbb{R}^d$ s.t.

$$\|y_i - y_j\| \approx d(z_i, z_j)$$

Multi-dimensional scaling, Kruskal, 64

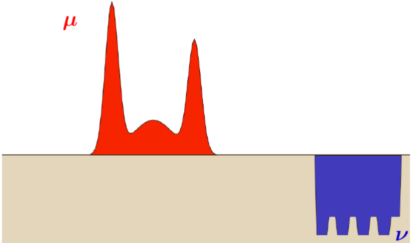
(Adjusting dimension d: see below)

3. Find mapping from basic representation (hand-crafted meta-features) onto intermediate representation:

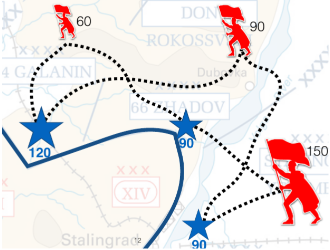
Optimal Transport

Optimal transport in one slide

Cuturi 13; Cuturi & Salomon, 17; Peyre & Cuturi 18



Monge (1781)



Kantorovitch (1939)

Formal background

- Given distribution μ on Ω_x, distribution ν on Ω_y
- Given a transport cost $c : \Omega_x \times \Omega_y \mapsto \mathbb{R}$
- Find γ distribution on $\Omega_x \times \Omega_y$, s.t. $\gamma_{x,.} = \mu$, $\gamma_{.,y} = \nu$ minimizing

$$\int_{\Omega_x \times \Omega_y} c(x,y) \mathrm{d}\gamma(x,y)$$

Optimal transport, a second slide

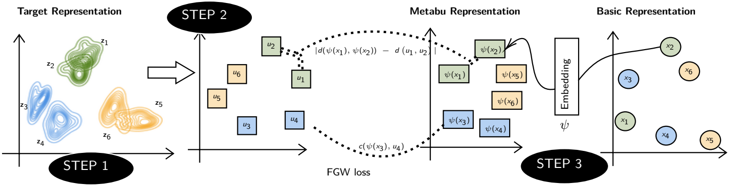
Fused Gromov-Wasserstein

Vayer et al. 19

Let (Ω_x, d_x) and (Ω_y, d_y) denote compact metric spaces, and $\mathbf{x}$ and $\mathbf{y}$ distributions respectively defined on Ω_x and Ω_y.

$$
\begin{aligned}
d^q_{FGW;\alpha}(\mathbf{x},\mathbf{y}) \quad &= \min_{\gamma \in \Gamma(\mathbf{x},\mathbf{y})} (1-\alpha) \underbrace{\left(\int_{\Omega_x \times \Omega_y} c(x,y) \mathrm{d}\gamma(x,y) \right)}_{\text{Wasserstein Loss}} \\
&+ \alpha \underbrace{\left(\int_{\Omega_x \times \Omega_y} \int_{\Omega_x \times \Omega_y} |d_x(x,x') - d_y(y,y')| \mathrm{d}\gamma(x,y) \mathrm{d}\gamma(x',y') \right)}_{\text{Gromov-Wasserstein Loss}}
\end{aligned} \tag{1}
$$

- Wasserstein: map x onto y such that it minimizes the expectation of cost $c(x,y) = \|x-y\|$
- Gromov-Wasserstein: enforce a rigid transport (preserving distances among pairs of points)



Algorithm Given **x** initial representation and **u** intermediate representation, train mapping ψ to optimize:

$$\psi^* = \arg\min_{\psi \in \Psi} \{d_{FGW;\alpha}(\psi_\sharp \mathbf{x}, \mathbf{u}) + \lambda \|\psi\|\} \quad (2)$$

with λ the regularization weight and $\|\psi\|$ the L_1 norm of ψ.

Output

$\psi_\sharp \mathbf{x}$ is new representation, function of the initial representation **x** (known, inexpensive, for all datasets)
with similar metric as the intermediate representation
(which itself approximates the metric of the target representation).

Experimental setting

Goal of experiments

- Compare METABU performance with baseline meta-features:
  - Hand-crafted meta-features (135)
  - AutoSkLearn (38) Feurer et al. 15
  - SCOT (4) Bardenet et al. 13
  - Landmarks (8) Pfahringer et al. 00

Settings

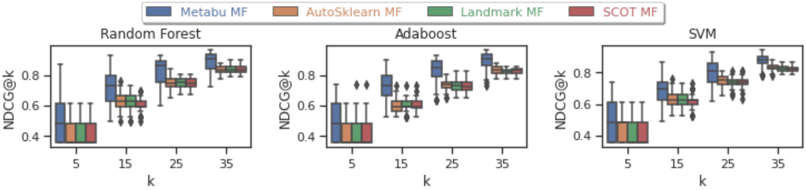
- ML algorithms / pipeline:
  - Adaboost (4), Random Forest (6), SVM (8), Auto-sklearn pipeline (110)
- Benchmark: 72 classification datasets (OpenML CC-18) (Leave one out validation)

Task 1: Metabu captures the target metric

Dataset $A \rightarrow$ nearest neighbor B according to

- Metabu features
- *vs* Baselines
- *vs* Oracle (Target) representation

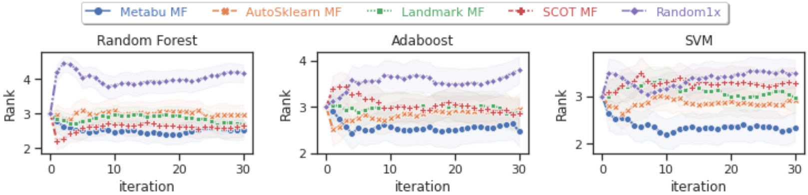
Performance: NDCG(ranks neighbors) wrt oracle representation (the higher, the better)



Metabu meta-features better capture the target metric

Task 2: Use METABU metric to achieve Auto-ML

Dataset $A \rightarrow$ best configurations for its nearest neighbor B
Performance: Average performance rank (the lower the better).

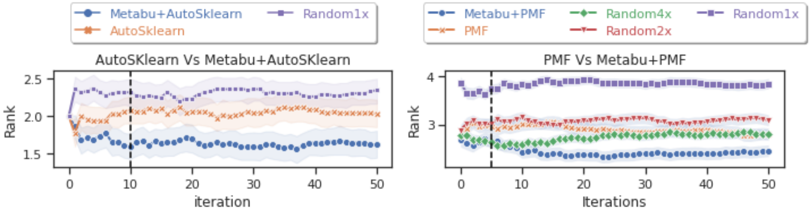


METABU configuration sampler outperforms baselines

Task 3: Use METABU within AutoML search

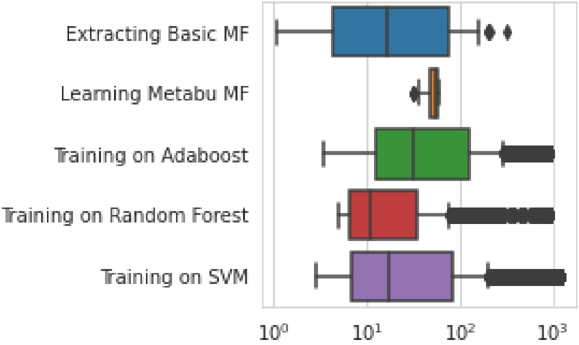
Initialize AutoML search using METABU metric
AutoML optimization:

- AutoSkLearn — Feurer et al., 15
- PMF — Fusi et al., 18



Using METABU meta-features to initialize AUTOSKLEARN and PMF search consistently improves over current AUTOSKLEARN and PMF.

Computational Effort



Runtime (in seconds)

In summary

METABU: Meta-learning for Tabular Data

- learns linear combinations of the hand-crafted meta-features.
- captures the topology of target representation, i.e., top hyper-parameter configurations.
- outperforms SoA meta-features on various configuration spaces.

Code available https://github.com/luxusg1/metabu

What did we learn ? Intrinsic dimension of the set of datasets

Measuring the intrinsic dimension of a space

Facco et al., 17

- For each point x, compute $\mu_x = \frac{d(x, y^{(2)},)}{d((x, y^{(1)})}$. with $y^{(1)}$ and $y^{(2)}$ first and second nearest neighbor of x
- Order points: draw line $(i, \log \mu_i)$ with $\mu_i < \mu_{i+1}$
- intrinsic dimension d: approximates slope of line $(i, \log \mu_i)$

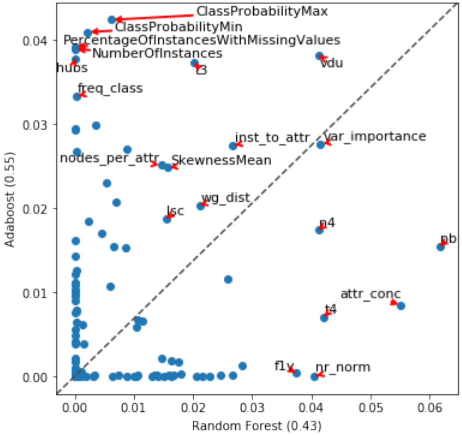
Intrinsic dimension of OpenML-CC

Alg. / Pipeline	dim Θ	**Intrinsic. dim**
Adaboost	4	8
Random Forest	6	9
SVM	8	14
Auto-sklearn	110	6

What did we learn ? Sensitivity of ML alg. wrt meta-features

Importance of meta-features

Random Forest vs Adaboost



- **PercentageOfInstancesWithMissingValues**: percentage of missing values
- **classProbabilityMin**: Minimum of class probabilities
- **var_importance**: features importance of the DT model for each attribute

Perspectives

Meta-representation

- From algo-dependent meta-features
- ... to a comprehensive representation

From a metric on datasets

- to evaluating *a priori* domain adaptation, transfer learning

Assessing ML evaluation

- Measuring the diversity of a benchmark
- Does Auto-ML overfit ?

Thanks!

Guillaume Bied

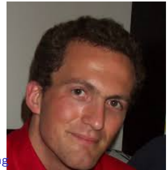
Elia Perennes

gane Hoffmann

Solal Nathan

Christphe Ga

Christophe Caillou

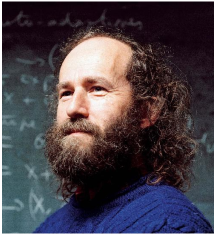


Bruno Crépon

Thanks!

Heri Rakotoarison

Isabelle Guyon



Marc Schoenauer