Learning Augmented Methods for Matching: Improving Invasive Species Management and Urban Mobility

Johan Bjorck [1], Qinru Shi [1], Carrie Brown-Lima [2]
Jennifer Dean [3], Angela Fuller [2 4 5], Carla Gomes [1]
[1] Department of Computer Science, Cornell [2] Department of Natural Resources and the Environment, Cornell
[3] New York Natural Heritage Program, College of Environmental Science and Forestry, SUNY
[4] U.S. Geological Survey, [5] New York Cooperative Fish and Wildlife Research Unit
{njb225, qs63, cjb37}@cornell.edu, jennifer.dean@dec.ny.gov, angela.fuller@cornell.edu, gomes@cs.cornell.edu

Abstract

With the success of machine learning, integrating learned models into real-world systems has become a critical challenge. Naively applying predictions to combinatorial optimization problems can incur high costs, which has motivated researchers to consider learning augmented algorithms that can make use of faulty or incomplete predictions. Inspired by two matching problems in computational sustainability where data are abundant, we consider the learning augmented min-cost matching problem where some nodes are revealed online while others are known a priori, e.g., by being predicted by machine learning. We develop an algorithm that is able to make use of this extra information and provably improves upon pessimistic online algorithms. We evaluate our algorithm on two settings from computational sustainability – the coordination of opportunistic citizen scientists for invasive species management and the matching between taxis and riders under uncertain trip duration predictions. In both cases, we perform extensive experiments on real-world datasets and find that our method outperforms baselines, showing how learning augmented algorithms can reliably improve solutions for problems in computational sustainability.

Introduction

With the success of machine learning models for prediction, integrating such learned models into real-world systems has become a critical challenge. While the models are typically evaluated with aggregate statistics, e.g., accuracy on a test set, for many applications, such averages might not necessarily imply efficiency. For example, in combinatorial optimization, predicting most of the choices correctly might nonetheless lead to poor solutions. An emerging paradigm for integrating machine learning into optimization algorithms, while giving theoretical guarantees, is to consider *learning augmented* algorithms (Lykouris and Vassilvitskii 2018). Here a machine learning model gives incomplete or partly incorrect predictions and given access to this oracle, one constructs an algorithm that provably performs well as long as the machine learning advice is sound but still gives guarantees for inaccurate predictions. This approach is applicable to many problems in computational sustainability where data are abundant and one wants to provide guarantees that resources are spent efficiently (Dilkina and Gomes 2010; Bondi et al. 2018). Information might also only be partly known for other reasons; e.g., land managers might only know approximately what resources are available in the future.

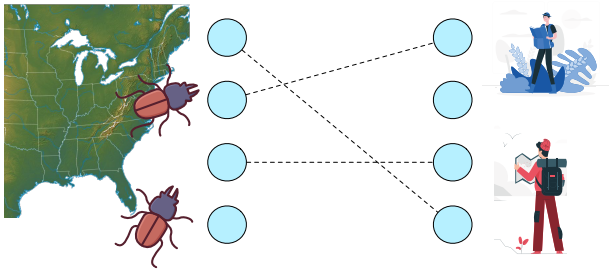

Figure 1: The economic damage of invasive species is estimated to be in the billions of dollars annually (Pimentel, Zuniga, and Morrison 2005), and in invasive species management, one needs to match observers to spatial locations in a landscape that might contain invasive species. Observers might be both employees whose efforts can be planned as well as opportunistic citizen scientists who's engagement is unknown at planning time. Orchestrating the efforts of these agents provides an important challenge.

In this work, we consider two matching problems subject to partial information: matching citizen scientists to observation locations for invasive species management and matching riders to taxis in urban mobility. In both settings, some parts of the matching might be known a priori, e.g., we can use predicted trip durations to guess what taxis are available, whilst other parts might be unknown, e.g., some taxis might arrive early due to favorable traffic conditions or model misspecification. We model these problems as a semi-online matching problem, where we want to find the minimum cost matching in an online fashion given partial predictions of the graph, e.g., taxis become available one by one and must irrevocably be matched to customers as they become available.

A primary motivation for this project is coordinating the efforts of citizen scientists for invasive species management as part of an ongoing collaboration with the New York Nat-

ural Heritage Program, who contribute to the state invasive species database through the online mapping system iMapInvasives (NatureServe 2020 (Accessed 2020-07-01). Citizen scientists is a broad term for engaging volunteer citizens in science projects (Bonney et al. 2009; Cox et al. 2015), and data from citizen scientists and paid managers are combined in the database to help the state of New York assess and monitor the spread of over four hundred invasive species across the state. A pervasive problem in these settings is that citizen scientists are opportunistic and possibly have misaligned incentives (Xue et al. 2013). Like many similar projects (Sullivan et al. 2009; Follett and Strezov 2015), the database uses citizen scientists to perform monitoring for invasive species, but the engagement of such citizen scientists is not known ahead of time. In addition to these volunteers, the database collaborates with paid employees whose schedules can be decided ahead of time; and coordinating the effort of opportunistic citizen scientists with predictably scheduled paid employees proves an important challenge in computational sustainability. In the case of urban mobility, inspired by a competition using data from the NYC Taxi and Limousine Commission (TLC), we consider a machine learning model predicting the duration of taxi rides and adjusting the matching between customers and taxis for model errors.

Modeling both these problems as learning-augmented min-cost matching problems, we develop an algorithm that provably improves upon pessimistic algorithms in the learning augmented setting, with an approximation bound that depends upon the amount of knowledge available. The algorithm is evaluated on two large real-world datasets, the taxi dataset of the NYC Taxi and Limousine commission and invasive species records from the database. We find that our algorithm consistently outperforms baselines. We summarize our contributions as follows:

- Formulate the problem of learning augmented min-cost matching motivated by applications in computational sustainability and provide an algorithm for the problem.
- Prove approximation guarantees for the algorithm that depends upon how much knowledge we have access to.
- Evaluate our algorithm on two large real-world datasets from urban mobility and invasive species management, showing that our method outperforms baselines.

Background

Min-Cost Matching. Consider a weighted bipartite graph $G = (V, E)$ with weights w_e for edge e. The two sides of the graph might represent, e.g., taxis and potential customers. We will refer to the two sides of the graphs as the jobs S and workers R. A perfect matching E is a subset $M \subseteq E$ such that each node is incident to exactly one edge in E. The weight of a matching E is $w(E) = \sum_{e \in E} w_e$. We will assume that there are n workers and jobs; a perfect matching is then of size n. In our applications, we primarily consider matching over spatial domains, e.g., matching taxis and customers; we thus assume that the graph G is *metric*. Let $d(i, j)$ denote the distance between nodes i, j, the metric property then implies

$$d(i, j) + d(j, k) \leq d(i, k) \qquad \forall i, j, k \in V \tag{1}$$

The min-cost perfect matching problems entail finding a matching M that minimizes $w(M)$, which can be done in poly-time (Kuhn 1955). Solutions often rely on so-called *augmenting paths* P (augmenting w.r.t some matching M), which are paths in G whose ending and starting nodes are unmatched in M such that every other edge $\in P$ is also $\in M$. Given such a path, one can expand the matching M by setting

$$M \leftarrow M \otimes P \tag{2}$$

Here $\otimes$ is the symmetric difference between the two sets, returning elements found in exactly one of the operand sets.

Online Matching. Online algorithms model scenarios where parts of a problem are revealed one-by-one, but one needs to make irreversible choices before all information is revealed. A canonical example is the ski-rental problem, where an agent has traveled to a ski resort and will continue to ski for as long as the weather is good. The duration of favorable weather is unknown, and each day the agent must choose between renting skis or paying a larger sum to buy skis, which can be used for the rest of the vacation. Online algorithms are typically evaluated on their *competitive ratio*, which measures the expected cost $\mathbb{E}[w(M)]$ of the obtained solution M and the optimal solution M_* that can be obtained if all information was known a priori. I.e., the competitive ratio c is

$$c = \mathbb{E}\left[\frac{w(M)}{w(M_*)}\right] \tag{3}$$

Learning Augmented Metric Matching. As machine learning has become increasingly successful, researchers have been interested in so-called *learning augmented* algorithm design. These are algorithms that can provably make use of incomplete or noisy advice coming from e.g. a machine learning model (Purohit, Svitkina, and Kumar 2018; Lykouris and Vassilvitskii 2018). It is typically not specified how the side-information is obtained, one often assumes that it comes from some oracle (which need not be a machine learning model).

We will consider an online matching problem where parts of the network are known ahead of time, e.g. by being predicted by a machine learning model. Our goal is to construct an algorithm that can make use of such side information, and improve upon pessimistic online algorithms. We will assume that the workers arrive in a random order and formally define our problem as

Definition 1 *The learning augmented metric matching problem consists of*

- *a metric bipartite graph G with jobs S and workers R. We assume $|R| = |S| = n$*
- *A set R_p of predicted workers containing $n - k$ workers known ahead of time.*

workers from R are revealed one by one in a uniformly random order and must be irrevocably matched to a job upon arrival. An algorithm must output a perfect matching E.

A Method for Learning Augmented Matching

We now present a method for solving problems of the type given in definition 1 by incorporating machine learning advice into classical online algorithms (Raghvendra 2016), a high-level illustration can be found in fig. 2. We will first present the algorithm and then discuss how to improve it with local search.

Algorithm 1:

input : Graph $G = (E, N)$, jobs $S \subseteq N$, predicted workers $R_p \subseteq N$, a random permutation R.
output: A matching M

$M \leftarrow \{\}$ // current matching
$G_0 \leftarrow$ induced subgraph of G on $R_p \cup S$
$M_* \leftarrow$ MinMatch (G_0)
$M_f \leftarrow M_*$ // planned matching for predicted workers not revealed
$\phi \leftarrow w(M*)$
for $r \in R$ **do**
  if $r \in R_p$ **then**
    $M \leftarrow M \cup M_f(r)$
  else
    $P \leftarrow \text{aug}(r, M_*)$ minimizing $\Delta\phi$ eq. (4)
    $\phi \leftarrow \phi + \Delta\phi(P, M_*)$ via eq. (4)
    $M_* \leftarrow M_* \bigoplus P$
    $M \leftarrow M \cup (r, s)$ where (r, s) are endpoints of P
  end
end
output M

Algorithm

The algorithm will maintain a matching M, which it outputs at the end, and auxiliary matchings M_f and M_*, the latter being updated throughout the procedure. At a high level, our strategy is to first find a suitable set of jobs for the predicted nodes R_p and then update our matching M as workers are revealed one by one via augmenting paths with respect to M_*. Throughout the algorithm, we will keep track of the variable ϕ, which will be useful for the analysis. At the very start, the algorithm initializes M to the empty set and M_* to the min-cost matching from all predicted workers, using, e.g., the Hungarian method (Kuhn 1955). Thereafter, when predicted workers arrive, they are assigned as per M_f. When an adversarial node $r \in R$ is revealed we find an augmenting path P from r to an empty job s that minimizes $\Delta\phi$, defined as

$$\Delta\phi(P, M_*) = \sum_{e \in P \setminus M_*} w(e) \tag{4}$$

after this, we update $M_* \leftarrow M_* \bigoplus P$ and assign the worker to the endpoint of P. See Algorithm 1.

Formal Analysis

It is straightforward to verify that Algorithm 1 gives a perfect matching, the reason being that we always connect unpredicted workers with endpoints of paths that are augmenting w.r.t M_*. Since all predicted workers are matched in M_* at the start, such paths must terminate in unmatched jobs. A formal proof is given in the Appendix. Of more interest is the approximation guarantees, which will depend upon k the number of unpredicted workers as follows.

Theorem 1 *Algorithm 1 has competitive ratio $\mathcal{O}(1+\log k)$.*

As is common in the analysis of learning augmented algorithm (Purohit, Svitkina, and Kumar 2018; Lykouris and Vassilvitskii 2018), our algorithm will interpolate between known methods at the extremes of perfect or unavailable machine learning advice. The algorithm reduces to the Hungarian method or the online method of (Raghvendra 2016) at such extremes, borrowing from the analysis of these methods but improving upon the $\mathcal{O}(\log n)$ guarantee the latter provides in the case of partial information. The algorithm can be analyzed by bounding the value of ϕ; it is first possible to lower bound it.

Proposition 1 *Let M_t be the matching at iteration t. For any t we then have*

$$w(M_t) + w(M_t^*) + w(M_f \setminus M_t) \leq 2\phi \tag{5}$$

Proof. We prove this by induction. It holds at $t = 0$ trivially. Let us assume that it holds for t and consider $t + 1$ and the node r we add this step. If $r \in R_p$ the RHS of eq. (5) is unchanged as per Algorithm 1. r will be matched to $s = M_f(r)$, and thus $w(M_t)$ will increase by $w(e_{(r,s)})$ whereas $w(M_f \setminus M_t)$ will decrease by the same amount. Then the LHS of eq. (5) is also the same and the inductive statement holds. Let us assume $r \notin R_p$. We then have

$$w(M_i^*) - w(M_{i-1}^*) = \sum_{e \in P \setminus M_{i-1}^*} w(e) - \sum_{e \in P \cap M_{i-1}^*} w(e)$$

$$= \Delta\phi(P_i) - \left(\frac{1}{2} \sum_{e \in P \cap M_{i-1}^*} w(e) + \frac{1}{2} \sum_{e \in P \cap M_{i-1}^*} w(e) \right)$$

we add and subtract $\frac{1}{2} \sum_{e \in P \setminus M_{i-1}^*} w(e)$ in the parenthesis

$$= \Delta\phi(P_i) - \Bigg(\underbrace{\frac{1}{2} \sum_{e \in P \cap M_{i-1}^*} w(e) + \frac{1}{2} \sum_{e \in P \setminus M_{i-1}^*} w(e)}_{= \frac{1}{2}\ell(P_i)} +$$

$$\underbrace{\frac{1}{2} \sum_{e \in P \cap M_{i-1}^*} w(e) - \frac{1}{2} \sum_{e \in P \setminus M_{i-1}^*} w(e)}_{= -\frac{1}{2}(w(M_i^*) - w(M_{i-1}^*))} \Bigg)$$

thus

$$\frac{1}{2}\big(w(M_i^*) - w(M_{i-1}^*)\big) = \Delta\phi(P_i) - \frac{1}{2}\ell(P_i) \tag{6}$$

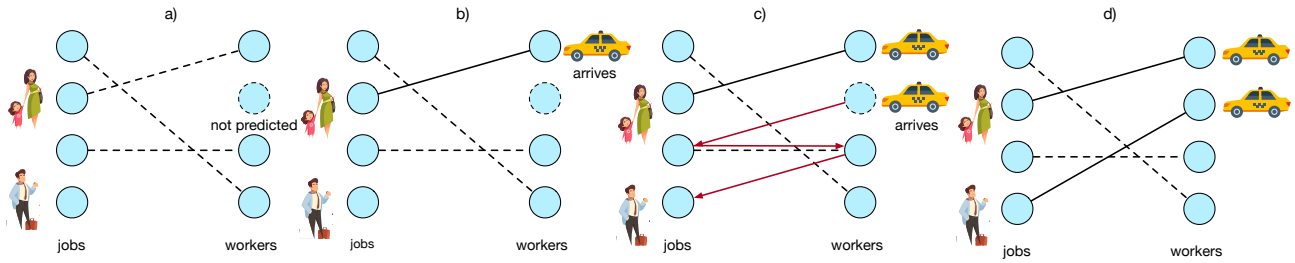

Figure 2: The high-level idea of our algorithm. a) For all predicted workers, we first calculate an optimal matching M_*. b) When a predicted worker arrives, we match it according to M_*. c) When an unpredicted worker arrives, we calculate an augmenting path relative to M_* and d) match the endpoints. We improve this general strategy by adding a local search procedure.

Using the metric property we have $\ell(P) \geq d(r, s)$, and we note that $d(r, s)$ is $\Delta w(M)$. This gives us

$$\Delta w(M_*) + \Delta w(M) \leq 2\Delta\phi \tag{7}$$

We know that s is unmatched in M_f, and since M_f is unchanged $w(M_f \setminus M)$ is also unchanged. Thus the inductive statement still holds. ■

After lower bounding ϕ, it is possible to upper bound it, utilizing the random permutation to bound its expectation, using the fact that not all workers are adversarial to improve upon pessimistic adversarial bounds (Raghvendra 2016).

Proposition 2 $\mathbb{E}[\phi] \leq (1 + H_k)w(M_{opt})$.

Proof. At $t = 0$ we have $\phi \leq w(M_{\text{opt}})$. We thus only need to bound the change $\Delta\phi$ from when we add $r \notin R_p$. Let us consider the augmenting paths from M_i^* to M_{opt}. Since M_{opt} is a perfect matching, there are $n-i$ vertex disjoint augmenting paths – one for each node that has not yet arrived. We let P'_j be the set of these augmenting paths, then

$$\sum_{j=1}^{n-i} \Delta\phi(P'_j) = \sum_{j=1}^{n-i} \left(\sum_{(s,r \in P'_j \notin M_i^*)} d(s, r) \right) \tag{8}$$

$$= \sum_{j=1}^{n-i} \left(\sum_{(s,r \in P'_j \cap M_{\text{opt}})} d(s, r) \right) \leq w(M_{\text{opt}})$$

Here we have used the fact that the paths are vertex disjoint. Let us now relabel such that worker r_i is r'_j. Since we always chose the path with the smallest $\Delta\phi$ we must have

$$\Delta\phi(P_i) \leq \Delta\phi(P'_j)$$

In the random arrival model, all adversarial nodes are equally likely to arrive at any time. Let us assume that k' adversarial nodes have already arrived, using eq. (8) we then have

$$\mathbb{E}[\Delta\phi(P_i)|r_i \in] \leq \frac{1}{k - k'} \sum_{q \in} \Delta\phi(P_q)$$

$$\leq \frac{1}{k - k'} \sum_{q} \Delta\phi(P_q) \leq \frac{w(M_{\text{opt}})}{k - k'}$$

In total, we then have

$$\mathbb{E}[\sum_{q} \Delta\phi(P_q)] = \sum_{q} \mathbb{E}[\mathbf{1}_{r_q \in}]\mathbb{E}[\Delta\phi(P_i)|r_i \notin R_p]] \tag{9}$$

$$\leq w(M_{\text{opt}}) \sum_{k'=1}^{k} \frac{1}{k'}$$

adding (9) to the fact that $\phi \leq w(M_{\text{opt}})$ at $t = 0$ yields the claim. ■

Proof of Theorem 1. Adding proposition 1 evaluated at $t = n$ and proposition 2 gives us

$$\frac{1}{2}\mathbb{E}[w(M)] \leq \frac{1}{2}\mathbb{E}\big[\big(w(M) + w(M_f \setminus M) + w(M_*)\big)\big]$$

$$\leq \mathbb{E}[\phi] \leq (1 + H_k)w(M_{\text{opt}})$$

inspecting the ends of this inequality yields the result. ■

Improvement via Local Search

It is possible to perform a local search before assigning a worker to a job to further improve the solution. When an adversarial worker r is revealed, we find a minimum weight augmenting path P from r to empty job s and use P to update M_*. Then, we define N_f to be the set of nodes that are matched in M_f but not matched in M, and H to be subgraph of G induced on $N_f \cup \{r, s\}$. We observe that if we update M_f and assign r based on the minimum weight matching on H, we can reduce the weight of the final matching without disturbing the overall structure of the algorithm. See Algorithm 2 for a complete description of the local search routine. If we replace line 13 in Algorithm 1 with the local search routine, we can show that M_* will remain the same in every iteration. It is straightforward to prove that this procedure can only improve Algorithm 1, see the Appendix.

Algorithm 2: LocalSearch(r, s)

1 $N_f \leftarrow$ nodes that are matched in M_f but not matched in M
2 $H \leftarrow$ induced subgraph of G on $N_f \cup \{r, s\}$.
3 $M_f \leftarrow$ MinMatch(H)
4 $M \leftarrow M \cup M_f(r)$

method	2/21	2/22	2/23	2/24	2/25	2/26	2/27	3/21	3/22	3/23	3/24	3/25	3/26	3/27
opt	132	179	87	129	92	143	256	249	110	77	192	137	117	118
bipartite 0.5	163	235	133	176	131	196	337	319	164	104	253	218	180	178
bipartite 1.0	192	271	142	195	143	222	401	381	180	107	304	228	198	190
bipartite 2.0	257	354	179	242	193	279	520	501	215	137	387	301	245	238
online	156	227	111	170	130	191	333	294	143	94	235	201	174	181
hybrid	156	227	112	171	129	178	333	294	143	94	234	201	172	181
greedy	158	238	136	177	135	204	353	299	173	104	241	210	177	181
local	**154**	**203**	**99**	**156**	**118**	**177**	**282**	**262**	**133**	**90**	**218**	**180**	**167**	**154**
adversarial	158	238	137	177	135	204	352	300	176	104	242	212	181	183
method	4/21	4/22	4/23	4/24	4/25	4/26	4/27	5/21	5/22	5/23	5/24	5/25	5/26	5/27
opt	89	168	106	160	184	135	115	260	161	134	116	184	196	149
bipartite 0.5	137	236	138	228	256	206	160	339	230	187	176	259	257	204
bipartite 1.0	142	277	159	250	277	215	180	380	266	200	184	292	291	223
bipartite 2.0	185	363	200	328	341	260	227	476	332	255	257	370	379	267
online	130	228	130	221	237	186	156	315	229	163	166	244	235	191
hybrid	130	228	129	217	237	186	154	315	227	163	167	242	235	191
greedy	140	241	133	222	239	210	160	322	219	190	178	243	242	199
local	**116**	**200**	**124**	**201**	**230**	**152**	**144**	**301**	**201**	**151**	**154**	**232**	**220**	**173**
adversarial	142	242	134	225	239	211	161	323	224	189	179	246	243	202

Table 1: Total distance (km) traveled for various methods on the taxi matching problem for dates (given as month/day) in 2016. Less is better. We see that our method local consistently outperforms alternatives, a hybrid greedy algorithm being a competitor.

Experiments

For evaluating our algorithm, we will consider experiments using graphs generated from two real-world datasets from invasive species management and taxi-cab matching. We will consider finding the minimum cost perfect matching in the learning augmented setting of section definition 1, comparing our algorithms to the following baselines:

- A greedy algorithm that myopically assigns a worker r to the closest unmatched job s. This comes with no guarantees, and we refer to it as greedy.
- A naive matching algorithm that first predicts the min-cost matching of the predicted workers, and reserves the workers not matched to this set to be greedily matched to unpredicted workers. This comes with no guarantees, and we refer to it as hybrid.
- The semi-supervised matching algorithm of (Kumar et al. 2018), which reserves a set of jobs for unpredicted workers with low "externality". This algorithm was developed for unweighted maximum matching; we adopt it by considering edges present if their distance is less than some cutoff proportional to the average distance per edge in the optimal solution. For constant of proportionality p, we refer to these as bipartite-p. Nodes that cannot be matched with edges below the threshold are matched greedily.
- The online algorithm of (Raghvendra 2016) which gives pessimistic performance guarantees and does not use any additional information. We refer to it as adversarial.

We will refer to our algorithm, with and without the local search routing, as online and local, respectively. It is worth noting that our algorithm provides guarantees, which can be useful beyond just providing performance. E.g., it might be useful for policy-makers to know how close to optimality the solutions are. Experiments are repeated five times; the mean is given in the main paper and the standard deviations in the Appendix.

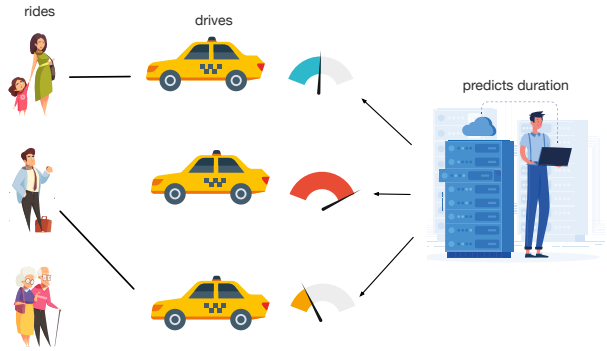

Figure 3: In urban mobility, matching customers to drivers is challenging as, e.g., changing traffic conditions makes it hard to estimate the travel time. Using a machine learning model to predict the trip duration, we evaluate our algorithm on the ability to generate matchings robust to prediction errors.

Taxi-Cab Matching

We conduct experiments on matching taxi drivers to customers. Traffic conditions and other issues are typically hard to predict, to this end, NYC Taxi and Limousine Commission (TLC) released a dataset of roughly two million taxi rides in the greater NYC area, and a competition to predict

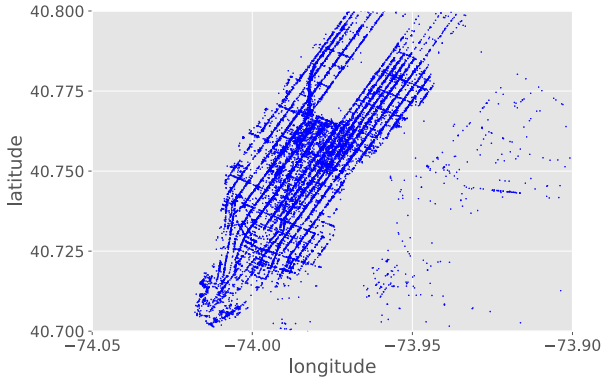

Figure 4: The area for the taxi dataset, each dot corresponds to a ride. Note the outline of Manhattan and Central park.

the duration of taxi rides was organized . The dataset contains the location and time of pickup/dropoff of customers and various other data (e.g., number of passengers, taxi company), but not the time of the cab order. See Figure 4 for an illustration over the geographic area. We consider matching taxis (**workers**) to riders (**jobs**). To construct the graph, we consider all n taxi cabs that become available in some time interval $[T, T+t_d]$ (i.e. they drop off a customer in this interval) and try to match them to the next n customers. We aim to minimize the Manhattan distances from the initial dropoff to the next pickup, and the edge weights correspond to this quantity. In the spirit of the original competition, we construct a machine learning model that predicts the duration of taxi rides. Taxis that were predicted to be free for new customers after $T + t_d$, but nonetheless arrived before due to, e.g., beneficial traffic conditions or miscalibrated prediction, are treated as unpredicted workers, other taxis are treated as predicted. The machine learning regression is random forest implemented via XGBoost (Chen and Guestrin 2016); features used include distance, pickup and dropoff area, time of the day, among others; see the Appendix for further details. In practical applications, one might also want to optimize for waiting time, as such information is not available for this dataset, we defer such studies to future work. We consider seven days for each of four months in 2016, taking $T = 3$ pm and t_d as 30 minutes. Results are given in Table 1, where we see that our method with local search consistently gives the best matchings.

[1]Available at www.kaggle.com/c/nyc-taxi-trip-duration

Invasive Species Management

We now consider the problem of invasive species management as part of a collaboration with the New York Natural Heritage Program. The dataset upon which the experiments are conducted comes from the iMapInvasives project (NatureServe 2020 (Accessed 2020-07-01). Its database currently consists of more than 200,000 observations and 408 species, spread over more than 30 years and 2,200 observers. Of central importance is the heterogeneous agents collaborating on this initiative; the state is divided into eight regions administered by individual organizations that make use of both employees and individual citizen scientists. For each year, we will aim to match volunteers/employees (**workers**) to sites deemed necessary to investigate (**jobs**). For this task, we do not assume that the side-advice comes from a machine learning model, but instead that volunteers are unpredicted due to their opportunistic engagement, whereas em-

method	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
bipartite 0.5	4777	4896	5310	4587	3723	4783	5645	6897	5873	4312	6723	3246	4003	5410
bipartite 1.0	5346	5494	5519	4880	3944	4925	5964	7103	6224	4560	6995	3360	4165	5812
bipartite 2.0	5444	5685	6132	5466	4837	6158	7395	8806	7209	5455	8003	4601	5109	6478
online	4316	4899	5005	4305	**3094**	4469	5042	6665	5006	3836	5967	2821	3897	4689
hybrid	4316	4899	4963	4268	**3094**	4469	5042	6665	4898	3814	5967	2851	3753	4690
greedy	4792	4835	5075	4387	3616	4689	5719	6734	5738	4349	6543	3441	4027	5314
local	**4085**	**4583**	**4706**	**4000**	**3094**	**4250**	**4692**	**6204**	**4881**	**3497**	**5695**	**2700**	**3229**	**4417**
adversarial	5079	5006	5202	4441	3801	4737	5758	7144	5966	4352	6462	3550	3944	5952
opt	4083	4443	4701	3973	3092	4158	4659	6145	4862	3415	5635	2575	3079	4399
method	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
bipartite 0.5	4272	4656	3324	4550	4214	4439	3978	5675	1705	**1971**	1983	1856	2043	1921
bipartite 1.0	4309	5152	3840	5124	4278	4780	4249	6155	1798	2310	2206	1933	2171	2071
bipartite 2.0	5023	5825	4402	5622	5553	6003	5250	8199	2594	2587	2831	2556	2372	2438
online	3531	4624	3108	3883	3797	4861	4261	5897	1898	2129	2224	1849	1842	1836
hybrid	3530	4652	3063	3885	3903	4889	4468	6107	1915	2155	2249	2044	1970	1899
greedy	3990	4493	3394	4478	4224	4342	3920	**5415**	1746	2029	2174	1855	1779	1902
local	**3363**	**4061**	**2819**	**3659**	**3556**	**3999**	3658	5468	**1703**	1982	**1935**	**1699**	**1726**	**1759**
adversarial	4211	4544	3335	4619	4280	4433	**3654**	5779	1906	2173	2062	2025	1999	1968
opt	3278	3723	2602	3617	3272	3552	3066	4819	1362	1490	1477	1372	1413	1389

Table 2: Total distance (km) traveled for various methods on the invasive species management problem by year. Less is better. We see that our method local typically outperforms alternatives, only being beaten by a small margin for three years.

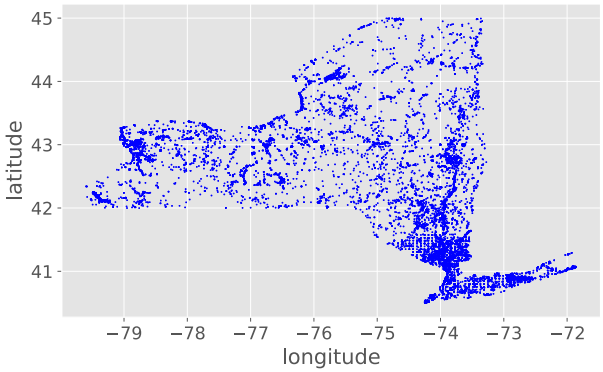

Figure 5: The geographical area that the invasive species dataset covers, each dot corresponds to an observation, on the ground, of an invasive species. Data are collected over 31 years.

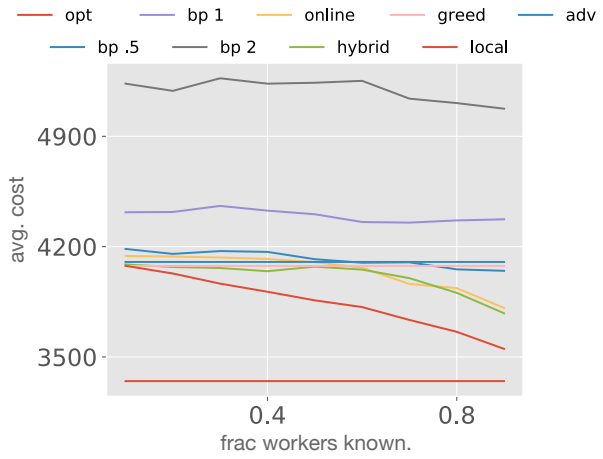

Figure 6: Ablation experiments showing the performance on the invasive species management matching problem of various algorithms, as the fraction of predicted nodes changes. Solution quality is given by average cost of matchings, measured in kilometers. Our proposed methods improves upon the alternatives, and improves as the number of predicted nodes grows.

ployees are known ahead of time. Observations are typically strongly correlated at the small scale, representing observations conducted the same day a couple of meters apart. We divide the landscape into patches corresponding to 0.2 degrees latitudes/longitudes and subsample the set of observations so that each patch has at most one observation per year, approximately corresponding to a day's work. Then, for each year, we construct the matching graph where the jobs correspond to the observations conducted that year, and the workers correspond to the observers. As we have not been able to obtain employment data, for a given year, observers that only conduct one observation and have no previous observations are treated as citizen scientists (modeled as unpredicted workers), others as employees (predicted workers). We weight an edge by the distance from the centroid of the employee's/citizen scientist's observations to the observation location. The results are given in Table 2, where we see that our method outperforms the alternatives. We also consider an ablation experiment to evaluate the impact of the number of predicted observers. Instead of treating the observers as unknown based upon historical data, we randomly pick the observer to be predicted or not. We then plot the average distance traveled, averaged over all years and five repetitions (with five fixed seeds), as a function of the fraction of known nodes. The results are given in Figure 6, with performance improving as with predicted observers.

Related Work

Matching problems are ubiquitous in sustainability applications, examples include health interventions (Wilder et al. 2018) and organ donor matching (Roth, Sönmez, and Ünver 2004), fair division of goods (Aleksandrov et al. 2015) and supply-demand matching in energy storage (Pickard, Shen, and Hansing 2009). The idea of learning augmented algorithms goes back at least to (Lykouris and Vassilvitskii 2018), which studies the problem in the context of machine-learned advice for caching policies. Other applications include frequency estimation in data streams (Hsu et al. 2018), low-rank estimation (Indyk, Vakilian, and Yuan 2019), scheduling (Lattanzi et al. 2020) among others (Purohit, Svitkina, and Kumar 2018). Within matching, (Kumar et al. 2018) introduces semi-supervised maximum matching, online matching where a certain subset of the nodes is known before. We instead consider min-cost perfect matching, which introduces additional complications as the externalities of a single bad choice can grow substantially. Pure online algorithms have a long history, see, e.g., (Karp, Vazirani, and Vazirani 1990), but is still an active area of research (Buchbinder, Segev, and Tkach 2019; Buchbinder et al. 2020; Devanur and Huang 2017). Both applications we consider here have extensive literature owing to their practical importance. In the context of invasive species management, citizen science has proven to be a promising strategy for conservation work (Crall et al. 2015; Rutledge et al. 2013; McKinley et al. 2017). On the computational side, considered methods include reinforcement learning (Taleghan et al. 2015), mixed integer programming solvers (Büyüktahtakın, Feng, and Szidarovszky 2014), stochastic dynamic programming and others (Shea and Possingham 2000). From a theoretical perspective (Bjorck et al. 2018) considers a predator-prey model for biocontrol, (Gupta et al. 2018) uses Hawkes processes for modelling and (Spencer 2012) considers an extension of the firefighter problem. Within the domain of urban mobility (Lowalekar, Varakantham, and Jaillet 2018; Freund et al. 2019) various dimensions of the problem have been considered, examples include pricing (Qiu et al. 2017), finding the minimum fleet size (Vazifeh et al. 2018) or allocating multiple passengers to the same ride (Santi et al. 2014). Again there is a heterogeneous set of strategies considered, from traditional combinatorial optimization (Nair and Miller-Hooks 2011) to reinforcement learning (Schultz and Sokolov 2018).

Conclusion

Motivated by problems in computational sustainability, we have introduced a novel learning augmented method for matching problems with partially unknown nodes. Our algorithm interpolates between a completely known setting and an adversarial online setting, and we provide a theoretical bound that improves with accurate predictions. We evaluate our method on two large scale datasets covering urban mobility and invasive species management and find that our method consistently outperforms alternatives. We believe that this research direction is broadly applicable to problems in computational sustainability, and hope that it can inspire future work outside of our two applications.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number CCF-1522054. This material is also based upon work supported by the Air Force Office of Scientific Research under award number FA9550-18-1-0136. We are also grateful from generous support from the TTS foundation. We are also thankful for funding from the New York State Environmental Protection Fund as administered by the New York State Department of Environmental Conservation. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Ethics Statement

Managing invasive species is a pervasive and economically important challenge in a globalized world, and finding appropriate strategies for combating them is an outstanding challenge. Similarly, urban mobility is a promising route towards reducing greenhouse gas emission, and improving its efficacy is important for both consumers and companies. We believe that our work furthers both these vital areas in computational sustainability. We do not see any specific negative societal impact resulting from this work and do not believe that we leverage bias from either dataset.

References

Aleksandrov, M.; Aziz, H.; Gaspers, S.; and Walsh, T. 2015. Online fair division: Analysing a food bank problem. *arXiv preprint arXiv:1502.07571* .

Bjorck, J.; Bai, Y.; Wu, X.; Xue, Y.; Whitmore, M.; and Gomes, C. 2018. Scalable Relaxations of Sparse Packing Constraints: Optimal Biocontrol in Predator-Prey Networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Bondi, E.; Fang, F.; Hamilton, M.; Kar, D.; Dmello, D.; Choi, J.; Hannaford, R.; Iyer, A.; Joppa, L.; Tambe, M.; et al. 2018. Spot poachers in action: Augmenting conservation drones with automatic detection in near real time. In *AAAI*, 7741–7746.

Bonney, R.; Cooper, C. B.; Dickinson, J.; Kelling, S.; Phillips, T.; Rosenberg, K. V.; and Shirk, J. 2009. Citizen science: a developing tool for expanding science knowledge and scientific literacy. *BioScience* 59(11): 977–984.

Buchbinder, N.; Feldman, M.; Filmus, Y.; and Garg, M. 2020. Online submodular maximization: Beating 1/2 made simple. *Mathematical Programming* 1–21.

Buchbinder, N.; Segev, D.; and Tkach, Y. 2019. Online algorithms for maximum cardinality matching with edge arrivals. *Algorithmica* 81(5): 1781–1799.

Büyüktahtakın, İ. E.; Feng, Z.; and Szidarovszky, F. 2014. A multi-objective optimization approach for invasive species control. *Journal of the Operational Research Society* 65(11): 1625–1635.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cox, J.; Oh, E. Y.; Simmons, B.; Lintott, C.; Masters, K.; Greenhill, A.; Graham, G.; and Holmes, K. 2015. Defining and measuring success in online citizen science: A case study of Zooniverse projects. *Computing in Science & Engineering* 17(4): 28–41.

Crall, A. W.; Jarnevich, C. S.; Young, N. E.; Panke, B. J.; Renz, M.; and Stohlgren, T. J. 2015. Citizen science contributes to our knowledge of invasive plant species distributions. *Biological invasions* 17(8): 2415–2427.

Devanur, N. R.; and Huang, Z. 2017. Primal dual gives almost optimal energy-efficient online algorithms. *ACM Transactions on Algorithms (TALG)* 14(1): 1–30.

Dilkina, B.; and Gomes, C. P. 2010. Solving connected subgraph problems in wildlife conservation. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, 102–116. Springer.

Follett, R.; and Strezov, V. 2015. An analysis of citizen science based research: usage and publication patterns. *PloS one* 10(11): e0143687.

Freund, D.; Henderson, S. G.; O'Mahony, E.; and Shmoys, D. B. 2019. Analytics and bikes: Riding tandem with motivate to improve mobility. *INFORMS Journal on Applied Analytics* 49(5): 310–323.

Gupta, A.; Farajtabar, M.; Dilkina, B.; and Zha, H. 2018. Discrete Interventions in Hawkes Processes with Applications in Invasive Species Management. In *IJCAI*, 3385–3392.

Hsu, C.-Y.; Indyk, P.; Katabi, D.; and Vakilian, A. 2018. Learning-based frequency estimation algorithms. In *International Conference on Learning Representations*.

Indyk, P.; Vakilian, A.; and Yuan, Y. 2019. Learning-based low-rank approximations. In *Advances in Neural Information Processing Systems*, 7402–7412.

Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, 352–358.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2(1-2): 83–97.

Kumar, R.; Purohit, M.; Schild, A.; Svitkina, Z.; and Vee, E. 2018. Semi-online bipartite matching. *arXiv preprint arXiv:1812.00134* .

Lattanzi, S.; Lavastida, T.; Moseley, B.; and Vassilvitskii, S. 2020. Online scheduling via learned weights. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1859–1877. SIAM.

Lowalekar, M.; Varakantham, P.; and Jaillet, P. 2018. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence* 261: 71–112.

Lykouris, T.; and Vassilvitskii, S. 2018. Competitive caching with machine learned advice. *arXiv preprint arXiv:1802.05399* .

McKinley, D. C.; Miller-Rushing, A. J.; Ballard, H. L.; Bonney, R.; Brown, H.; Cook-Patton, S. C.; Evans, D. M.; French, R. A.; Parrish, J. K.; Phillips, T. B.; et al. 2017. Citizen science can improve conservation science, natural resource management, and environmental protection. *Biological Conservation* 208: 15–28.

Nair, R.; and Miller-Hooks, E. 2011. Fleet management for vehicle sharing operations. *Transportation Science* 45(4): 524–540.

NatureServe. 2020 (Accessed 2020-07-01). *iMapInvasives: NatureServe's online data system supporting strategic invasive species management*. URL http://www.imapinvasives.org.

Pickard, W. F.; Shen, A. Q.; and Hansing, N. J. 2009. Parking the power: Strategies and physical limitations for bulk energy storage in supply–demand matching on a grid whose input power is provided by intermittent sources. *Renewable and Sustainable Energy Reviews* 13(8): 1934–1945.

Pimentel, D.; Zuniga, R.; and Morrison, D. 2005. Update on the environmental and economic costs associated with alien-invasive species in the United States. *Ecological economics* 52(3): 273–288.

Purohit, M.; Svitkina, Z.; and Kumar, R. 2018. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems*, 9661–9670.

Qiu, H.; et al. 2017. *Dynamic pricing in shared mobility on demand service and its social impacts*. Ph.D. thesis, Massachusetts Institute of Technology.

Raghvendra, S. 2016. A robust and optimal online algorithm for minimum metric bipartite matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Roth, A. E.; Sönmez, T.; and Ünver, M. U. 2004. Kidney exchange. *The Quarterly journal of economics* 119(2): 457–488.

Rutledge, C.; Fierke, M.; Careless, P.; and Worthley, T. 2013. First detection of Agrilus planipennis in Connecticut made by monitoring Cerceris fumipennis (Crabronidae) colonies. *Journal of Hymenoptera Research* 32: 75.

Santi, P.; Resta, G.; Szell, M.; Sobolevsky, S.; Strogatz, S. H.; and Ratti, C. 2014. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences* 111(37): 13290–13294.

Schultz, L.; and Sokolov, V. 2018. Deep reinforcement learning for dynamic urban transportation problems. *arXiv preprint arXiv:1806.05310* .

Shea, K.; and Possingham, H. P. 2000. Optimal release strategies for biological control agents: an application of stochastic dynamic programming to population management. *Journal of Applied ecology* 37(1): 77–86.

Spencer, G. 2012. Robust cuts over time: Combatting the spread of invasive species with unreliable biological control. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Sullivan, B. L.; Wood, C. L.; Iliff, M. J.; Bonney, R. E.; Fink, D.; and Kelling, S. 2009. eBird: A citizen-based bird observation network in the biological sciences. *Biological conservation* 142(10): 2282–2292.

Taleghan, M. A.; Dietterich, T. G.; Crowley, M.; Hall, K.; and Albers, H. J. 2015. PAC optimal MDP planning with application to invasive species management. *The Journal of Machine Learning Research* 16(1): 3877–3903.

Vazifeh, M. M.; Santi, P.; Resta, G.; Strogatz, S. H.; and Ratti, C. 2018. Addressing the minimum fleet problem in on-demand urban mobility. *Nature* 557(7706): 534–538.

Wilder, B.; Ou, H.-C.; de la Haye, K.; and Tambe, M. 2018. Optimizing Network Structure for Preventative Health. In *AAMAS*, 841–849.

Xue, Y.; Dilkina, B.; Damoulas, T.; Fink, D.; Gomes, C.; and Kelling, S. 2013. Improving your chances: Boosting citizen science discovery. In *First AAAI Conference on Human Computation and Crowdsourcing*.