CS598NJ: Statistical Reinforcement **Fall 2018**

Theoretical analysis of DAgger

Unnat Jain *Handed In: January 28, 2019*

1 Introduction

This project presents the proofs introduced in Ross et al. [2011] from the perspective of the author. Since these results have already been proved before, we would want to present them in a top-down approach. We start with the most relevant proofs in Sec. 2. This should aid the reader to quickly evaluate the author's understanding. Later we will concisely present all the relevant theorems, lemmas and definitions in Sec. 3. Afterwards, in Sec. 3.5, we present the main proof for the finite sample setting of DAgger. Finally in Sec. 4, we touch upon some recent works in Computer Vision which leverage imitation learning for improvements and rely mostly on DAgger.

2 No-regret and reduction approach

2.1 Table of notations

Since there quite a few notations, we compiled all of them in Tab. 1. This helps us (and hopefully the reader too) to explain (and understand) the theoretical analysis more easily.

2.2 DAgger algorithm

The algorithm is concisely presented in the original paper Ross et al. [2011], hence I'll directly link that here. The most important points of the algorithm are:

- Builds a dataset of expert demonstrations *i.e.* $(s, \pi^*(s))$ pairs to iteratively train policies $\hat{\pi}_1, \hat{\pi}_2 ... \hat{\pi}_N$
- The novelty is in what should be the distribution to sample states from, to create the above dataset.
- Algorithm suggests to iteratively sample trajectories by weighing the learned policy. This weight $1 - \beta_i$ increases with iteration i.
- If the best policy in $\hat{\pi}_i$ (*i.e.* $\hat{\pi}$) is deployed at test time (obviously not π_i, as we don't have the expert π^* at test time!), the number of mistakes are linear in T and ϵ_N (true loss of best policy).
- The paper mentions that it can also be shown that the same guarantees hold if we deploy a meta-policy which randomly picks one of the policies from $\hat{\pi}_{1:N}$ (instead of the best one from $\hat{\pi}_i$)

This form of the DAgger algorithm (Algorithm 1) utilizes the expert at every iteration and slowly reduces it's dependence on the expert. This is ensured by decaying the value of β_i with i.

Π	class of all policies considered
T	horizon
π	a particular policy from Π
d_π^t	distribution over states if π was followed from time= 1 to time= $t-1$
d_π	$\frac{1}{T}\sum_{t=1}^{t=T} d_\pi^t$ *i.e.* the averaged distribution over states if π was followed from time= 1 to time= $t-1$
$C(s,a)$	the cost of executing a at state s
$C_\pi(s)$	$\mathbb{E}_{a\sim\pi(s)}\{C(s,a)\}$ the average cost of executing π at state s
$J(\pi)$	$\sum_{t=1}^{t=T}\{$cost of executing π under it's own distribution of states$\}$ $=\sum_{t=1}^{t=T}\mathbb{E}_{s\sim d_\pi^t}\{C_\pi(s)\}$ $=T\mathbb{E}_{s\sim d_\pi}\{C_\pi(s)\}$ Since we usually do not know $C(s,a)$, we need a surrogate loss which is a measure of how far off is π from π^* (expert's policy)
$l(s,\pi)$	the surrogate loss, could be $0-1$ error, MSE *etc.* between π and π^*
$\pi_{1:N}$	sequence of policies π_1, π_2 ... π_N, such that π_i was executed at i^{th} iteration
$\hat{\pi}_{1:N}$	sequence of policies $\hat{\pi}_1$, $\hat{\pi}_2$ ... $\hat{\pi}_N$, such that $\hat{\pi}_i$ is the learned policy at iteration i
$\{\beta_i\}$	real number sequence, s.t. $\frac{1}{N}\sum_{i=1}^{N}\beta_i \to 0$ as $N\to\infty$. The particular case that we use for proofs is $\beta_i \le (1-\alpha)^{i-1}$ where α is a constant independent of T
$\hat{\pi}$	best policy amongst the sequence $\hat{\pi}_{1:N}$ (when test under it's own distribution of states). Mathematically, $\hat{\pi}=\text{argmin}_{\pi\in\hat{\pi}_{1:N}}\mathbb{E}_{s\sim d_\pi}\{l(s,\pi)\}$
ϵ_N	loss of the best policy in hindsight. Mathematically, $\epsilon_N=\min_{\pi\in\Pi}\frac{1}{N}\sum_{i=1}^{i=N} l_i(\pi)=\min_{\pi\in\Pi}\frac{1}{N}\sum_{i=1}^{i=N}\mathbb{E}_{s\sim d_{\pi_i}}\{l(s,\pi)\}$ (Eq. (2))
l_i	loss at the i^{th} iteration. For DAgger analysis of $l_i(\pi)=\mathbb{E}_{s\sim d_{\pi_i}}\{l(s,\pi)\}$ (see Sec. 2.5)
$l_{\max}$	upper bound on l_i. Mathematically, $l_i(s,\hat{\pi}_i)\le l_{\max}$ for all policies $\hat{\pi}_i$, s s.t. $d_{\hat{\pi}_i}(s)>0$
γ_N	average regret as defined by Eq. (3).
$\hat{\gamma}_N$	average regret of policy sequence $\hat{\pi}_{1:N}$
n_β	largest $n\le N$ such that $\beta_n>\frac{1}{T}$. Since $\{\beta_i\}$ is non-decreasing a n_β exists.

.

Table 1: A compilation of important notations used in the theoretical analysis of DAgger.

A simpler form of this algorithm can be obtained by setting $\beta_i = 1$ for $i = 1$ and $\beta_i = 0$ for $i \geq 2$. This is equivalent to sampling the expert under a distribution of states to obtain $\mathcal{D}_1$. In later iterations the states are sample from learned policies $\hat{\pi}_i$ for $i \geq 2$, instead of a mixture of expert and learned policy.

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

Algorithm 1: Dagger algorithm (credits Ross et al. [2011])

2.3 Assumptions

It is important to understand the assumptions under which we derive the bounds for DAgger.

- loss is strongly convex [helps to establish result in Sec. 3.3]
- the underlying cost function is upper bounded by the surrogate loss l [helps to establish Theorem 2 from Theorem 1]
- real number sequence, s.t. $\frac{1}{N}\sum_{i=1}^{N} \beta_i \to 0$ as $N \to \infty$. The particular case that we use for proofs is $\beta_i \leq (1 - \alpha)^{i-1}$ where α is a constant independent of T.

2.4 Guarantees

With most notations and assumptions explained, we now present the most crucial result of the paper. Theorem 1 detailed below simply follow from Theorem 3 (see Sec. 2.5).

Theorem 1 *There exists a policy $\hat{\pi}$ in sequence $\hat{\pi}_{1:N}$ s.t. $\mathbb{E}_{s \sim d_{\hat{\pi}}}\{l(s, \hat{\pi})\} \leq \epsilon_N + O(\frac{1}{T})$*

The particular policy can be obtained by choosing the best of $\hat{\pi}_{1:N}$. More specifically, $\hat{\pi} = \text{argmin}_{\pi \in \hat{\pi}_{1:N}} \mathbb{E}_{s \sim d_\pi}\{l(s, \pi)\}$. As mentioned in Sec. 2.2, a meta-policy which randomly picks one of the policies from $\hat{\pi}_{1:N}$ could also be shown to have the same guarantees. From Theorem 1, and under the assumption the underlying cost function is upper bounded by the surrogate loss l, we get $J(\hat{\pi}) \leq T\epsilon_N + O(1)$. Since this hold for any cost function bounded in $[0, 1]$, l also upper bounds the $0 - 1$ loss with respect to π^*. Hence, combining this with Theorem 4, we get Theorem 2

Theorem 2 *If N is $\tilde{O}(uT)$[1], there exists a policy $\hat{\pi}$ in sequence $\hat{\pi}_{1:N}$ s.t.* $J(\hat{\pi}) \leq J(\pi^*) + uT\epsilon_N + O(1)$

2.5 Analysis

In this section, we wish to show that DAgger can be used to find a policy which has the guarantees stated in Sec. 2.4 under its own distribution of states. To this end, we choose the loss functions l_i as the loss under the distribution of states dictated of the i^{th} policy for the i^{th} iteration:

$$l_i(\pi) = \mathbb{E}_{s\sim d_{\pi_i}}\{l(s,\pi)\} \tag{1}$$

For N iteration, let the loss of the best policy in hindsight be defined as:

$$\epsilon_N = \min_{\pi\in\Pi} \frac{1}{N}\sum_{i=1}^{i=N} l_i(\pi) = \min_{\pi\in\Pi} \frac{1}{N}\sum_{i=1}^{i=N} \mathbb{E}_{s\sim d_{\pi_i}}\{l(s,\pi)\} \tag{2}$$

Let $l_{\max}$ be the upper bound on l_i *i.e.* $l_i(s,\hat{\pi}_i) \le l_{\max}$ for all policies $\hat{\pi}_i$, s s.t. $d_{\hat{\pi}_i}(s) > 0$. We define n_β as the largest $n \le N$ such that $\beta_n > \frac{1}{T}$. Since $\{\beta_i\}$ is non-decreasing a n_β exists. $\hat{\gamma}_N$ is the average regret (defined by Eq. (3)) of sequence $\hat{\pi}_{i:N}$.

Theorem 3 *There exists a policy $\hat{\pi}$ in sequence $\hat{\pi}_{1:N}$ s.t.*

$$\mathbb{E}_{s\sim d_{\hat{\pi}}}\{l(s,\hat{\pi})\} \le \epsilon_N + \hat{\gamma}_N + \tfrac{2l_{\max}}{N}[n_\beta + T\textstyle\sum_{i=n_\beta+1}^{i=N}\beta_i]$$

Proof.

$$\begin{aligned}
&\min_{\hat{\pi}\in\hat{\pi}_{1:N}} \mathbb{E}_{s\sim d_{\hat{\pi}}}\{l(s,\hat{\pi})\} \\
&\le \frac{1}{N}\sum_{i=1}^{i=N} \mathbb{E}_{s\sim d_{\hat{\pi}_i}}\{l(s,\hat{\pi}_i)\} \quad \text{(since minimum is less than average)} \\
&\le \frac{1}{N}\sum_{i=1}^{i=N} [\mathbb{E}_{s\sim d_{\pi_i}}\{l(s,\hat{\pi}_i)\} + 2l_{\max}\min(1,T\beta_i)] \quad \text{(from Corollary 2)} \\
&= \frac{1}{N}\sum_{i=1}^{i=N} [l_i(\hat{\pi}_{1:N})\} + 2l_{\max}\min(1,T\beta_i)] \quad \text{(from Eq. (1))} \\
&\le \hat{\gamma}_N + \min_{\pi\in\Pi}\frac{1}{N}\sum_{i=1}^{i=N} l_i(\pi) + \frac{1}{N}\sum_{i=1}^{i=N} 2l_{\max}\min(1,T\beta_i)] \quad \text{(from Eq. (3))} \\
&= \hat{\gamma}_N + \epsilon_N + \frac{1}{N}\sum_{i=1}^{i=N} 2l_{\max}\min(1,T\beta_i)] \quad \text{(from Eq. (2))} \\
&\le \hat{\gamma}_N + \epsilon_N + \frac{2l_{\max}}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] \quad \text{(from Corollary 1)}
\end{aligned}$$

Proving Theorem 1 from Theorem 3:

1. $\hat{\gamma}_N$ term:
   For any no-regret algorithm, $\hat{\gamma}_N$ term is already $\tilde{O}(\frac{1}{N})$ under the assumption of l being strongly convex. So we need N to be $\tilde{O}(T)$ to make this term $O(\frac{1}{T})$.

[1] this (*soft-O* notation) was new to me, hence (informally) defining it here: $f(n)$ is $\tilde{O}(g(n))$ means $f(n)$ is $O(g(n)\log^k(g(n))$ for some k

2. $\frac{1}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i]$ term:
   We work under the assumption stated in Sec. 2.3, $\beta_i \leq (1-\alpha)^{i-1}$ where α is a constant independent of T. Hence, $\frac{1}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] \leq \frac{1}{N\alpha}[\log T + 1]$. So if N is $\tilde{O}(T)$ (as assumed in Theorem 1), then the above simplifies to $O(\frac{1}{T})$.

From point 1 above, we need N to be $\tilde{O}(T)$ for any no-regret algorithm, hence the second point incurs no extra cost.

3 Relevant background

3.1 Bound on cost-to-go $J(\pi)$

We present a proof similar to the one derived in [Ross and Bagnell, 2010]. This is a general proof, which applies to any policy π which has a surrogate loss ϵ under its own distribution of states. We make use of this in Sec. 2.4 to derive Theorem 2 from Theorem 1. Also, $l(s,\pi)$ is an upper bound cost C (for any C in $[0,1]$). Hence $l(s,\pi)$ is also an upper bound on $0-1$ loss. Let $Q_t^{\pi'}$ be the t-step cost of executing π in state s.

Theorem 4 *Let π be such that $\mathbb{E}_{s\sim d_\pi}\{l(s,\pi)\} = \epsilon$, and $Q^{\pi^*}_{T-t+1}(s,a) - Q^{\pi^*}_{T-t+1}(s,\pi^*) \leq u$ for all actions a, $t \in 1,2,...T$ and $d_\pi^t(s) > 0$, then $J(\pi) \leq J(\pi^*) + uT\epsilon$.*

Proof. For this proof we make use of a construction $\pi_{1:t}$. This is the policy which deploys π for the first t steps and then executes the expert policy π^* for the remaining steps. Then:

$$
\begin{aligned}
& J(\pi) \\
& = J(\pi_{1:T}) \\
& = J(\pi_{1:0}) + \sum_{t=0}^{t=T-1}[J(\pi_{1:T-t}) - J(\pi_{1:T-t-1})] \quad \text{(a standard add+subtract redundant terms)} \\
& = J(\pi^*) + \sum_{t=0}^{t=T-1}[J(\pi_{1:T-t}) - J(\pi_{1:T-t-1})] \quad \text{(by defining of construction } \pi_{1:t}) \\
& = J(\pi^*) + \sum_{t=0}^{t=T-1}[\text{Avg. cost of executing } \pi \textit{ vs. } \pi^* \text{ at time}=(T-t)] \quad \text{(by defining of } J(\pi)) \\
& = J(\pi^*) + \sum_{t=0}^{t=T-1}\mathbb{E}_{s\sim d_\pi^t}[Q^{\pi^*}_{T-t}(s,\pi) - Q^{\pi^*}_{T-t}(s,\pi^*)] \quad \text{(by defining of } Q_t^{\pi^*}(s,\pi)) \\
& = J(\pi^*) + \sum_{t=1}^{t=T}\mathbb{E}_{s\sim d_\pi^t}[Q^{\pi^*}_{T-t+1}(s,\pi) - Q^{\pi^*}_{T-t+1}(s,\pi^*)] \quad \text{(manipulating index of summation)} \\
& \leq J(\pi^*) + u\sum_{t=1}^{t=T}\mathbb{E}_{s\sim d_\pi^t}[l(s,\pi)] \\
& (\because Q^{\pi^*}_{T-t+1}(s,\pi) - Q^{\pi^*}_{T-t+1}(s,\pi^*) = 1 \text{ with P}(\pi \text{ not choosing the same action as } \pi^*) \\
& = J(\pi^*) + uT\epsilon \quad \text{(by definition of } \epsilon)
\end{aligned}
$$

3.2 Online Learning

An algorithm for online learning provides a policy π_i for iteration i, which has a loss of $l_i(\pi_i)$. At the next iteration, with new information of $l_i(\pi_i)$, the algorithm can provide a new policy π_{i+1}. This would lead to a loss $l_{i+1}(\pi_{i+1})$, and so on. The losses l_i are unknown.

3.3 No-regret algorithms

A no regret algorithm outputs a policy sequence $\pi_1, \pi_2 ... \pi_N$ so that the average regret becomes 0 as N $\to \infty$. We would refer to the average regret for policy $\hat{\pi}_{1:N}$:

$$\frac{1}{N}\sum_{i=1}^{i=N} l_i(\pi_i) - \min_{\pi \in \Pi} \frac{1}{N}\sum_{i=1}^{i=N} l_i(\pi) \leq \gamma_N \tag{3}$$

For no-regret: $\lim_{N\to\infty} \gamma_N = 0$. As shown by [Hazan et al., 2006, Kakade and Shalev-Shwartz, 2009, Kakade and Tewari, 2009], if l_i is strongly convex, then γ_N is $\tilde{O}(\frac{1}{N})$

3.4 Total variational distance between $d_{\hat{\pi}_i}$ and d_{π_i}

We would need a bound over the total variational distance between $d_{\hat{\pi}_i}$ and d_{π_i}. Lemma 1 helps establish this bound.

Lemma 1 $\|d_{\hat{\pi}_i} - d_{\pi_i}\|_1 \leq 2T\beta_i$
Straightforward proof, same as (lemma 4.1 in [Ross et al., 2011])

Note that a trivial bound would be $\|d_{\hat{\pi}_i} - d_{\pi_i}\|_1 \leq 2$, and Lemma 1 is tighter when $\beta_i \leq \frac{1}{T}$. Since β_i are non-decreasing there exists a n_β which is the largest $n \leq N$ such that $\beta_n > \frac{1}{T}$. From the definition of n_β, it can be concluded:

Corollary 1 $\frac{1}{N}\sum_{i=1}^{i=N} \min(1, T\beta_i) \leq \frac{1}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N} \beta_i]$

Also, from Lemma 1 we get:

Corollary 2 $\mathbb{E}_{s\sim d_{\hat{\pi}_i}}\{l_i(s, \hat{\pi}_i)\} \leq \mathbb{E}_{s\sim d_{\pi_i}}\{l_i(s, \hat{\pi}_i)\} + 2l_{\max}\min(1, T\beta_i)$

3.5 Finite sample extension

Theorem 3 holds in the probabilistic setting, where the online learning algorithm observes infinite samples. While this isn't true in practice, since we would only observe finite samples.

In the finite sample setting, at each iteration i, assume we sample m trajectories. This sample forms $\mathcal{D}_i$. Hence, $l_i(\pi)$ for this setting would be $\mathbb{E}_{s\sim\mathcal{D}_i}\{l(s,\pi)\}$. The no-regret online learner would come with a guarantee analogous to Eq. (3). Particularly, $\frac{1}{N}\sum_{i=1}^{i=N} \mathbb{E}_{s\sim\mathcal{D}_i}\{l(s,\pi)\} - \min_{\pi\in\Pi}\frac{1}{N}\sum_{i=1}^{i=N} \mathbb{E}_{s\sim\mathcal{D}_i}\{l(s,\pi)\} \leq \gamma_N$. Analogously, $\hat{\epsilon}_N = \min_{\pi\in\Pi}\frac{1}{N}\sum_{i=1}^{i=N} \mathbb{E}_{s\sim\mathcal{D}_i}\{l(s,\pi)\}$ is the training loss of the best policy in hindsight (in this setting). With these slightly different notations, the following theorem holds.

Theorem 5 *With probability* $1-\delta$*, there exists a policy* $\hat{\pi} \in \hat{\pi}_{1:N}$ *s.t.* $\mathbb{E}_{s\sim d_{\hat{\pi}}}\{l(s,\hat{\pi})\} \leq \hat{\epsilon}_N + \hat{\gamma}_N + \frac{2l_{\max}}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] + l_{\max}\sqrt{\frac{2\log(\frac{1}{\delta})}{mN}}$

Proof. For this proof we use a construction of Y_{ij}. The support is over $i \in \{1...N\}$ and $j \in \{1...m\}$. Y_{ij} is defined to be the difference between the expected per step loss of π_i under state distribution d_{π_i} and the average per step loss of π_i under the j^{th} sample trajectory with π_i at iteration i. Y_{ij} variables are zero mean and bounded in $[-l_{\max}, l_{\max}]$. When considered in the order $Y_{11}...Y_{1m}, Y_{21}...Y_{2m}...Y_{n1}...Y_{nm}$, they form a martingale[2]. Now we need to apply the a more general concentration inequality than the one we commonly used in class (Hoeffding's). This is Azuma–Hoeffding inequality, which gives concentration result for the values of martingales that have bounded differences (which is the case here). Hence, the following holds: $\frac{1}{mN}\sum_{i=1}^{i=N}\sum_{j=1}^{j=M} Y_{ij} \leq l_{\max}\sqrt{\frac{2\log(\frac{1}{\delta})}{mN}}$ with probability at least $(1-\delta)$. Hence with probability at least $(1-\delta)$:

$$
\begin{aligned}
&\min_{\hat{\pi}\in\hat{\pi}_{1:N}} \mathbb{E}_{s\sim d_{\hat{\pi}}}\{l(s,\hat{\pi})\} \\
&\leq \frac{1}{N}\sum_{i=1}^{i=N}[\mathbb{E}_{s\sim d_{\pi_i}}\{l(s,\hat{\pi}_i)\}] + \frac{2l_{\max}}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] \quad \text{(same as Theorem 3)} \\
&= \frac{1}{N}\sum_{i=1}^{i=N}[\mathbb{E}_{s\sim \mathcal{D}_i}\{l(s,\hat{\pi}_i)\}] + \frac{1}{mN}\sum_{i=1}^{i=N}\sum_{j=1}^{j=M} Y_{ij} + \frac{2l_{\max}}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] \quad \text{(by def. of } Y_{ij}) \\
&\leq \frac{1}{N}\sum_{i=1}^{i=N}[\mathbb{E}_{s\sim \mathcal{D}_i}\{l(s,\hat{\pi}_i)\}] + l_{\max}\sqrt{\frac{2\log(\frac{1}{\delta})}{mN}} + \frac{2l_{\max}}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] \quad \text{(from Azuma-Hoeffding's)} \\
&\leq \hat{\epsilon}_N + \hat{\gamma}_N + l_{\max}\sqrt{\frac{2\log(\frac{1}{\delta})}{mN}} + \frac{2l_{\max}}{N}[n_\beta + T\sum_{i=n_\beta+1}^{i=N}\beta_i] \quad \text{(similar to Theorem 3)}
\end{aligned}
$$

Azuma-Hoeffding's inequality indicates that $N \times m$ should be of $O(T^2 log(\frac{1}{\delta}))$ for generalization error to be of $O(\frac{1}{T})$. This result implied another the finite sample guarantee (theorem 3.3 in [Ross et al., 2011]).

4 DAgger in Computer Vision

In computer vision, recent research on visual navigation and embodied applications Gupta et al. [2017a,b], Das et al. [2018a,b] have found imitation learning to be particularly helpful. Deep net models train robustly by reinforcement learning methods when "warm started" by imitation learning. Variants of DAgger have been employed in this regard Gupta et al. [2017a], Das et al. [2018b]. Hence understanding the guarantees and bounds of DAgger is particularly interesting to me, which lays the motivation for this report.

[2] this was new to me, hence (informally) defining it here: a martingale is a sequence of random variables (i.e., a stochastic process) for which, at a particular time, the conditional expectation of the next value in the sequence, given all prior values, is equal to the present value [ref: wikipedia]

References

A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied Question Answering. In *Proc. CVPR*, 2018a.

A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Neural Modular Control for Embodied Question Answering. In *Proc. ECCV*, 2018b.

S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive Mapping and Planning for Visual Navigation. In *Proc. CVPR*, 2017a.

Saurabh Gupta, David Fouhey, Sergey Levine, and Jitendra Malik. Unifying map and landmark based representations for visual navigation. *arXiv preprint arXiv:1712.08125*, 2017b.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *International Conference on Computational Learning Theory*, 2006.

Sham M Kakade and Shai Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *NeurIPS*, 2009.

Sham M Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *NeurIPS*, 2009.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. pages 661–668, 2010.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.