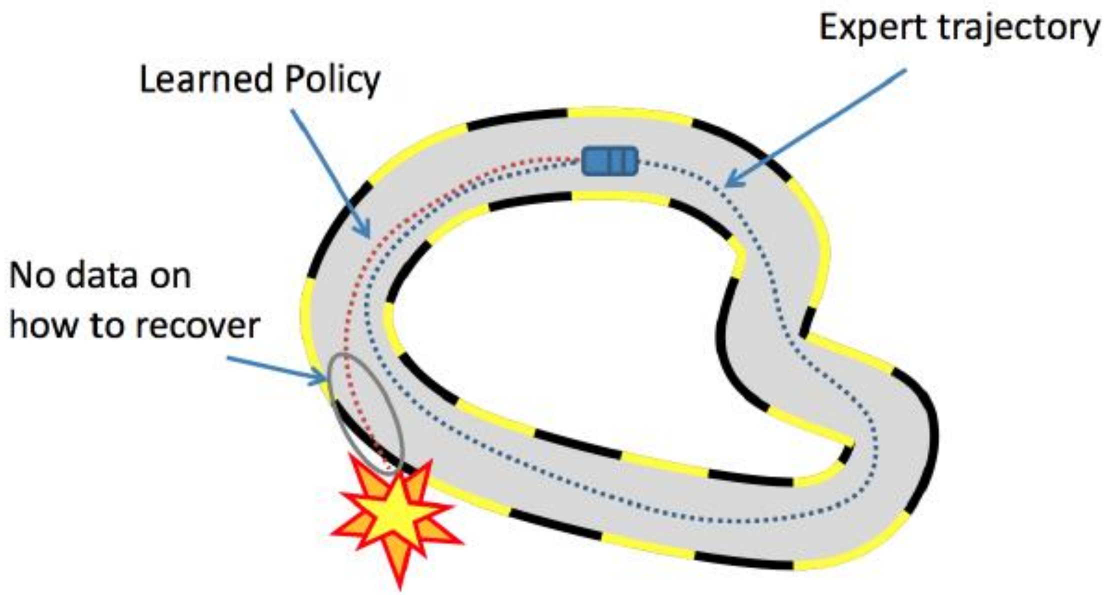
Effective Reduction for Imitation Learning

Ross and Bagnell

Summarized by Hyecheol (Jerry) Jang

Introduction – Problem Formulation

- **Imitation Learning**
  Train classifier to replicate an expert's policy given training data
- **Problem of Imitation Learning Algorithms**
  - Compounding of errors (e.g. driving performance)
  - Supervised learner making mistakes with some small probability (ϵ) making total cost growing quadratically $O(\epsilon T^2)$



Introduction – Problem Formulation

- **Goal**
  - Optimize the total T-step cost of the learning policy
- **Problem**
  - Without knowing the policy in-advance, not possible to generate samples from the induced distribution of states
  - When optimizing the learning policy, we do not know how the state distribution will be affected
  - *Learners are nearly always too optimistic about performance*
- **Solution**
  - Chane the current policy slowly
  - Making new policy's state distribution like the old policy *(Kakade, 2002)*

Introduction – What This Paper About

- **Goal**
  - *Analyze how the approach leads to better performance in imitation learning*
- **Key ideas**
  - Policy starts from querying and executing the expert's action (mimicking)
  - **Slowly replace it with learning policy**
- **Constraint**
  - Require more interactive setting than traditional imitation learning setups
  - Learner need to be allowed to interact with system and
    can query expert policy at any given state
  - *Such interactivity is possible in many real-world imitation learning problem*
    - *Teleoperated robot*

Introduction – What This Paper About

- **Ultimate Goal**
  - *Present simple, sequential algorithm of training non-stationary policy by iterating over time steps*
  - *Show that the algorithms have better performance bounds*
- **Plan**
  - Reduction-based analysis
  - Connect to Conservative Policy Iteration (CPI / RL) and **SEARN** (structured prediction, https://arxiv.org/abs/0907.0786)
  - *SMILe (Stochastic Mixing Iterative Learning) algorithm*
    - Simple, iterative approach providing the benefits of SEARN with simpler implementation and less interaction with an expert

Notation

- π^*: expert's policy to mimic, deterministic
- π_s: distribution over actions of policy π in state s
- T: task horizon
- $C(s, a)$: immediate cost of doing action a in state s, $[0, 1]$
- $C_\pi(s) = \mathbb{E}_{a \sim \pi_s}(C(s, a))$:
  expected immediate cost of performing policy π in state s
- $e(s, a) = I(a \neq \pi^*(s))$:
  0-1 loss of executing action a in state s, comparing to π^*
- $e_\pi(s) = \mathbb{E}_{a \sim \pi_s}(e(s, a))$

Notation

- d_π^i: state distribution at time i following the policy π from initial time
- $d_\pi = \frac{1}{T}\sum_{i=1}^{T} d_\pi^i$:
  encoding state visitation frequency over T time following policy π
- $J(\pi) = T\mathbb{E}_{s\sim d_\pi}(C_\pi(s))$: T-step cost of executing policy π
- $\mathcal{R}_\Pi(\pi) = J(\pi) - \min_{\pi'\in\Pi} J(\pi')$:
  regret bound of a policy π w.r.t the best policy in policy class Π

Traditional Approach

- Trains a classifier that learns to replicate the expert's policy under the state distribution induced by expert
- Minimize 0-1 loss under distribution

$$d_{\pi^*}: \hat{\pi} = \underset{\pi \in \Pi}{\operatorname{argmin}} \mathbb{E}_{s \sim d_\pi}(e_\pi(s))$$

- If the resulting classifier(policy) $\hat{\pi}$ makes mistake with small probability of ϵ, then we can guarantee Theorem 2.1

Theorem 2.1. *Let $\hat{\pi}$ be such that $\mathbb{E}_{s \sim d_{\pi^*}}[e_{\hat{\pi}}(s)] \leq \epsilon$. Then $J(\hat{\pi}) \leq J(\pi^*) + T^2\epsilon$. (Proof in Supplementary Material)*

Traditional Approach

Theorem 2.1. *Let $\hat{\pi}$ be such that $\mathbb{E}_{s \sim d_{\pi^*}}[e_{\hat{\pi}}(s)] \leq \epsilon$. Then $J(\hat{\pi}) \leq J(\pi^*) + T^2\epsilon$. (Proof in Supplementary Material)*

- **Reason for Quadratic Growth**
  - As $\hat{\pi}$ makes mistake, it ends up in the new state that are not visible in π^*
  - Incurring maximal cost of 1
- Fails to give good performance bound due to
  - mismatching between test and training distribution ($\hat{\pi} \neq \pi^*$)
  - Learner does not learn how to recover from mistakes

Forward Training Algorithm

- **Solution for Poor Performance Bound of Traditional Approach**

- Allow training to occur over several iteration
- Train one policy for one particular time step
- If learner makes mistake, the expert demonstrates how to recover at future steps
- Stops when once it learned a policy of all T time step

Forward Training Algorithm

- **Some more notations**
  - π_i^n: policy executed at time step i after n-th iteration
  - π^n: non-stationary policy denoted by π_i^n for $i = 1, \dots, T$
  - $J^{\pi}(\pi', t)$: expected T-step cost of executing π' at time t and π on other time
  - $\mathbb{A}\left(\pi^{i-1}, \pi_i^i\right) = J^{\pi^{i-1}}\left(\pi_i^i, i\right) - J(\pi^{i-1})$: policy disadvantage of π_i^i
    Indicating increment of T-step cost due to changed current policy at single step
- **Steps**
  - Initialize $\pi_1^0, \pi_2^0, \dots, \pi_n^0$ with expert's action (querying expert)
  - At time i, train π_i^i / others remain unchanged
  - After T iteration, π^T does not query from expert, end of training

Forward Training Algorithm

Theorem 3.1. $J(\pi^n) = J(\pi^*) + n\bar{\mathbb{A}}$, *where* $\bar{\mathbb{A}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{A}(\pi^{i-1}, \pi_i^i)$. *(Proof in Supplementary Material)*

- **Analysis**
  - Need to minimize $\mathbb{A}(\pi^{i-1}, \pi_i^i)$
  - Same as minimizing $J^{\pi^{i-1}}(\pi_i^i, i)$
  - i.e. minimizing cost-to-go from step i
- **Problem**
  - Impractical in imitation learning
  - Require ability to try several actions from the same state
    - Need to restart the system in any particular states, unrealistic

Forward Training Algorithm

- **Problem**
  - **Impractical in imitation learning**
  - Require ability to try several actions from the same state
    - Need to restart the system in any particular states, unrealistic
  - Learn value-estimation
    - Requiring more samples
    - Less robust
  - Interaction requirements with an expert is unnatural

Forward Training Algorithm

- **Solution**
  - Use agreement with expert to bound the loss w.r.t. an arbitrary cost
  - When $\pi_i^i = \pi_i^{i-1}$, then it is π^*
  - $\mathbb{A}(\pi^{i-1}, \pi_i^i) = 0$

Initialize $\pi_1^0, \ldots, \pi_T^0$ to query and execute π^*.
for $i = 1$ **to** T **do**
  Sample T-step trajectories by following π^{i-1}.
  Get dataset $\mathcal{D} = \{(s_i, \pi^*(s_i))\}$ of states, actions taken by expert at step i.
  Train classifier $\pi_i^i = \mathrm{argmin}_{\pi \in \Pi} \mathbb{E}_{s \sim \mathcal{D}}(e_\pi(s))$.
  $\pi_j^i = \pi_j^{i-1}$ for all $j \neq i$
end for
Return $\pi_1^T, \ldots, \pi_T^T$

Algorithm 3.1: Forward Training Algorithm.

  - Choose π_i^i to minimize 0-1 loss w.r.t. π^* under $d_{\pi^{i-1}}^i$

Forward Training Algorithm

- **Analysis**
  - $J^{\pi}(\pi', t, s)$: expected T-step cost of π conditioned + at state s at time t, executing π'
  - Suppose $\sup_{\pi \in \Pi, s | d^i_{\pi_{i-1}}(s)>0} [J^{\pi^{i-1}}(\pi, i, s) - J^{\pi^{i-1}}(\pi^*, i, s)] \leq u_i.$ Making u_i being the maximal increases in expected cost-to-go at time i
  - If we choose π^i_i s.t. $\mathbb{E}_{s \sim d^i_{\pi^{i-1}}}\left(e_{\pi^i_i}(s)\right) = \epsilon_i$, the policy disadvantage is bounded by $\mathbb{A}\left(\pi^{i-1}, \pi^i_i\right) \leq u_i \epsilon_i$
  - Implied $J\left(\pi^T\right) \leq J(\pi^*) + T\overline{\epsilon u}$, where $\overline{\epsilon u} = \frac{1}{T}\sum_{i=1}^{T} \epsilon_i$

Forward Training Algorithm

- **Analysis**
  - Implied $J\left(\pi^T\right) \leq J(\pi^*) + T\overline{\epsilon u}$, where $\overline{\epsilon u} = \frac{1}{T}\sum_{i=1}^{T} \epsilon_i$
  - Worst Case: $\overline{\epsilon u} \leq T\bar{\epsilon}$ ($\bar{\epsilon}$= $\frac{1}{T}\sum_{i=1}^{T} \epsilon_i$)
    same guarantee as the traditional approach
- **Some Good Properties?**
  - Changing only one action in current policy will not increase the cost by much more than small constant k on average
  - When π^* is stable, we can expect that only few steps will be on worst case

Forward Training Algorithm

- **Generalization**
  - If $\sup_{t \leq T, \pi_i^i \in \Pi \forall i \leq t, s | d_{\pi^{i-1}}(s) > 0} [J^{\pi^{t-1}}(\pi_t^t, t, s) - J^{\pi^{t-1}}(\pi^*, t, s)] \leq k$
  - Then we have $u_i \leq k$ for all i, making $J(\pi^T) \leq J(\pi^*) + kT\bar{\epsilon}$
- **Drawbacks**
  - Impractical when T is large
  - Does not extend to infinite horizon tasks (only training non-stationary policy)

Stochastic Mixing Iterative Learning Algorithm

- **Motivation**
  - Forward Algorithm: Guarantee smaller regret by changing policy slowly
  - Using stationary stochastic policy over iteration makes same effect
- **Details**
  - $\pi^{n+1} = (1-\alpha)\pi^n + \alpha\hat{\pi}^{n+1}$
  - After n iteration, the probability of querying expert becomes $(1-\alpha)^n$, when n becomes infinite, the probability becomes 0
- **What to do**
  - How to train policy $\hat{\pi}^{n+1}$
  - Choose the proper α
  - The number of iteration N, so that we can ensure good performance of $\hat{\pi}^N$

Stochastic Mixing Iterative Learning Algorithm

- **Idea**
  - Starts with CPI/SEARN
    - Impractical if applying directly to imitation learning
    - Requiring optimization of cost-to-go at each iteration
    - Previously work proved that the performance is worse than traditional approach by log factor
- **Alternatives / Constraint**
  - Policy disadvantage is bounded by minimizing the immediate 0-1 classification loss

Stochastic Mixing Iterative Learning Algorithm

- **New Notation**
  - $J_k^{\pi}(\pi', t_1, \dots t_k)$: expected T-step cost of executing π' at steps $\{t_1, \dots, t_k\}$
  - $\mathbb{A}_k(\pi, \pi')$= $J_k^{\pi}\left(\pi'\right) - J(\pi)$: k-th order policy disadvantage of π' w.r.t. π

  - $J_k^{\pi}(\pi') = \frac{1}{\binom{T}{k}} \sum_{t_1=1}^{T-k+1} \dots \sum_{t_k=t_{k-1}+1}^{T} J_k^{\pi}(\pi', t_1, \dots t_k)$

    expected T-step cost of executing π' k times and policy π at all other steps

Stochastic Mixing Iterative Learning Algorithm

- **Analysis**

Lemma 4.1. *If* $\alpha \leq \frac{1}{T}$, *then for any* $k \in \{1, 2, \ldots, T-1\}$, $J(\pi^n) \leq J(\pi^0) + n\sum_{i=1}^{k} \alpha^i \binom{T}{i}(1-\alpha)^{T-i}\bar{\mathbb{A}}_i + n\alpha^{k+1}T\binom{T}{k+1}$, *where* $\bar{\mathbb{A}}_i = \frac{1}{n}\sum_{j=1}^{n} \mathbb{A}_i(\pi^{j-1}, \hat{\pi}^j)$. *(Proof in Supplementary Material)*

- Probability of choosing expert (π^0) at time n is $p_n = (1-\alpha)^n$
- Want to have strong performance guarantee of unsupervised policy $\tilde{\pi}^n$
  - Never querying expert

Lemma 4.2. $J(\tilde{\pi}^n) \leq J(\pi^n) + p_n T^2$. *(Proof in Supplementary Material)*

Stochastic Mixing Iterative Learning Algorithm

Lemma 4.2. $J(\tilde{\pi}^n) \leq J(\pi^n) + p_n T^2$. *(Proof in Supplementary Material)*

- **Analysis**
  - Lemma 4.2 implies
    - If $n \geq \frac{2}{\alpha} \ln T$, then $p_n \leq \frac{1}{T^2}$ s.t. $J(\tilde{\pi}^n) \leq J(\pi^n) + 1$
    - Additional cost of 1 becomes negligible for large T
- **SEARN**
  - Effectively seek to minimize directly the previous bound for $k = 1$
  - Using $N = \frac{2}{\alpha} \ln T$, $\alpha = T^{-3}$
  - Guaranteeing $J\left(\tilde{\pi}^N\right) \leq J(\pi^*) + O(T \ln T \overline{\mathbb{A}_1} + \ln T)$
  - Require estimation of cost-to-go under the current policy **for each action**
    - Impractical

Stochastic Mixing Iterative Learning Algorithm

- **Problem of SEARN**
  - T-step roll-outs requires a lot of interaction
    - For A actions, k trajectories per estimation, m sampled state per iteration
    - $O\left(mAkT^4 \log T\right)$ queries to complete all $O(T^3 \log T)$
  - Cost function to minimize is unknown
  - Not possible to obtain the cost-to-estimates

Stochastic Mixing Iterative Learning Algorithm

- **Alternative**
  - $\hat{\pi}^n$ mimicking π^{n-1}
  - As for all k, we got $\mathbb{A}_k(\pi^{n-1}, \hat{\pi}^n) = 0$
  - Only unknown part π^{n-1} is expert's policy, we can focus on learning expert's
  - If $\hat{\pi}^{*n} = \operatorname{argmin}_{\pi\in\Pi} \mathbb{E}_{s\sim d_{\pi^{n-1}}}(e_\pi(s))$,
    then policy $\hat{\pi}^n = p_{n-1}\hat{\pi}^{*n} + (1 - p_{n-1})\hat{\pi}^{n-1}$
    - Approximate π^{n-1} with new estimate $\hat{\pi}^{*n}$ from π^*

Initialize $\pi^0 \leftarrow \pi^*$ to query and execute expert.
for $i = 1$ **to** N **do**
  Execute π^{i-1} to get $\mathcal{D} = \{(s, \pi^*(s))\}$.
  Train classifier $\hat{\pi}^{*i} = \operatorname{argmin}_{\pi\in\Pi} \mathbb{E}_{s\sim\mathcal{D}}(e_\pi(s))$.
  $\pi^i = (1-\alpha)^i\pi^* + \alpha\sum_{j=1}^{i}(1-\alpha)^{j-1}\hat{\pi}^{*j}$.
end for
Remove expert queries: $\tilde{\pi}^N = \frac{\pi^N - (1-\alpha)^N\pi^*}{1-(1-\alpha)^N}$
Return $\tilde{\pi}^N$

Algorithm 4.1: The SMILe Algorithm.

Stochastic Mixing Iterative Learning Algorithm

- **Analysis**
  - SMILe can use any classification algorithm
  - Weights of each learned policy remaining constant over iteration
    - SEARN: old policies have much smaller weights than new policies
  - **Theorem 4.1.** *For* $\alpha = \frac{\sqrt{3}}{T^2\sqrt{\log T}}$, *and* $N = 2T^2(\ln T)^{3/2}$, *then* $J(\tilde{\pi}^N) \leq J(\pi^*) + O(T(\tilde{\mathbb{A}}_1 + \tilde{\epsilon}) + 1)$. *(Proof in Supplementary Material)*
    - Improve from SEARN
  - Upper bound in policy disadvantage
    - Favorable mixing properties of dynamic system
    - Recovery behavior of the policy to be mimicked
    - In practice:
policy disadvantage is typically bounded on average due to recoverability of problem setting

Experimental Results

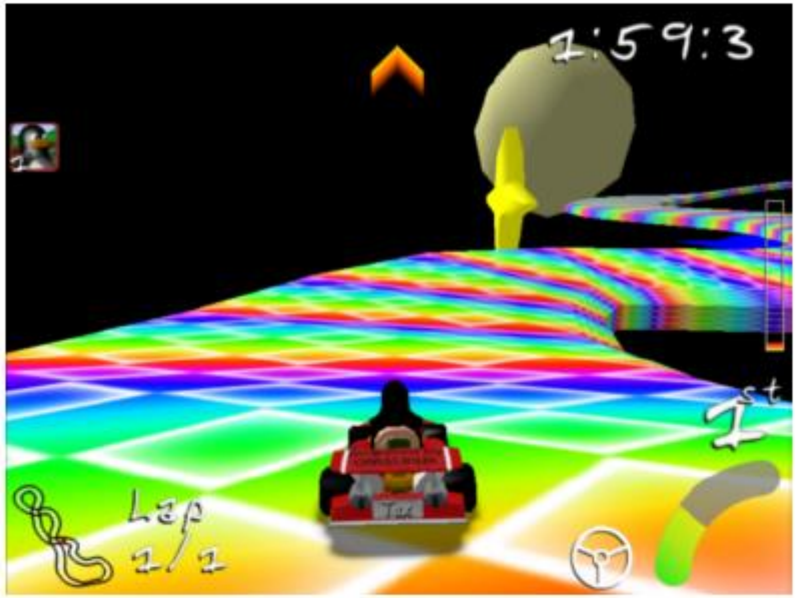
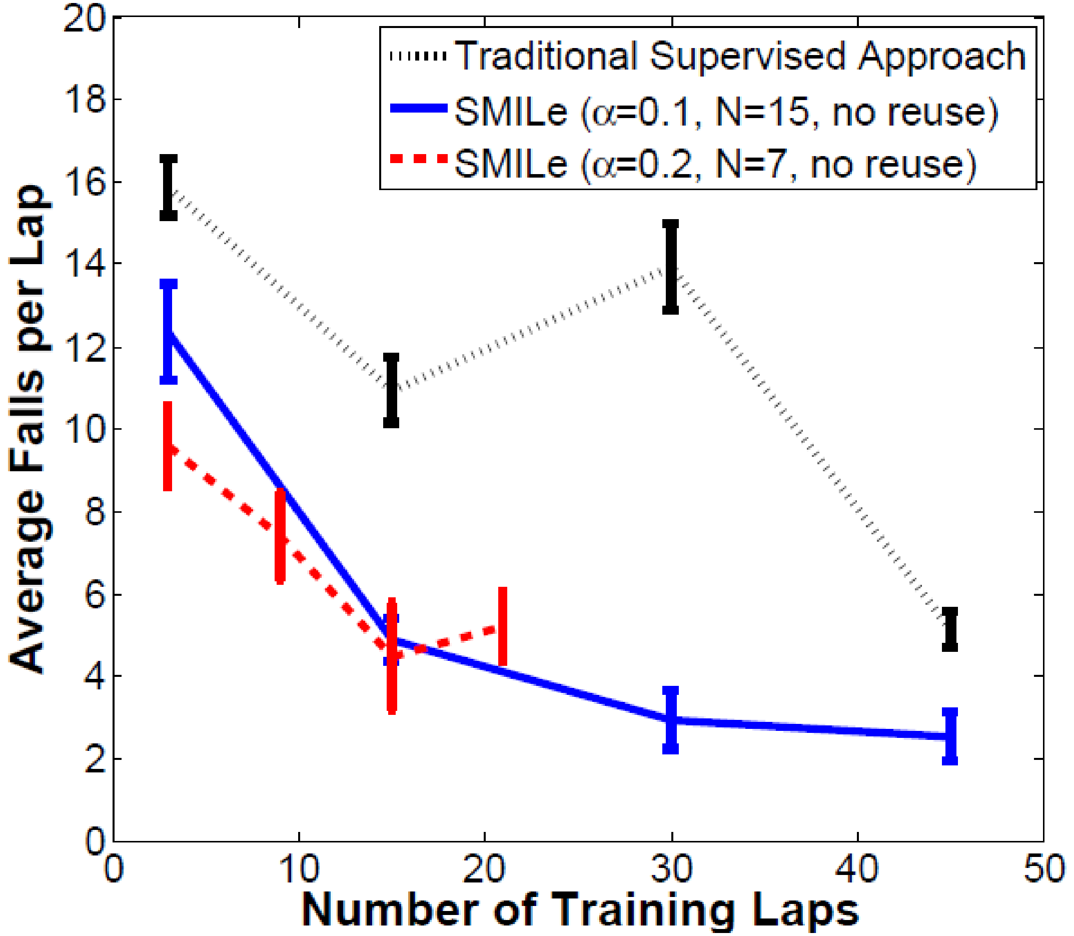
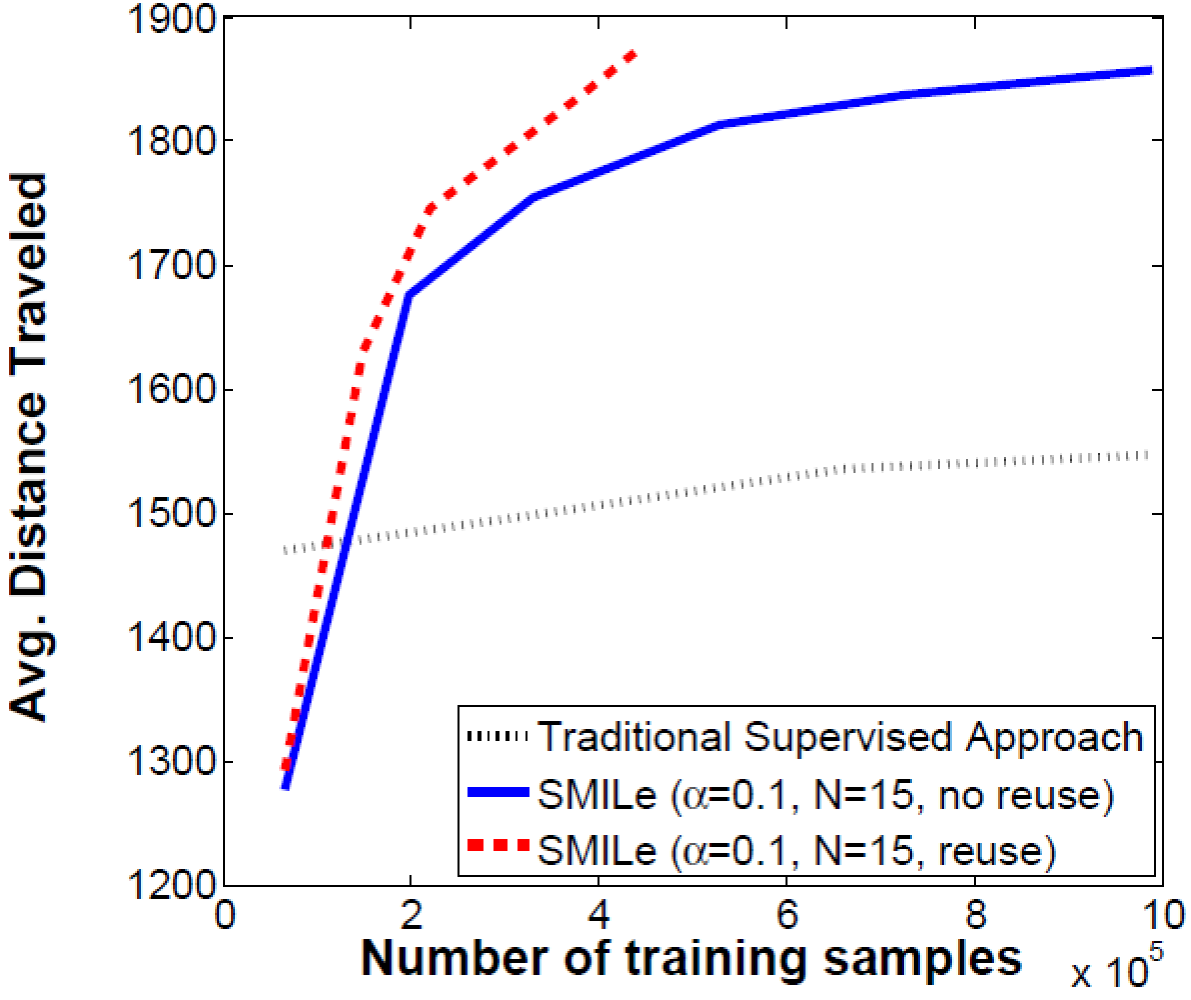


Figure 1: Image from Super Tux Kart's Star Track.



Experimental Results

Figure 3: Captured image from Mario Bros.



Summary

- Focused on the problem of Imitation Learning
  - Learning policy influences the future test input
  - Leading compounding errors / Regret bound grows quadratically
- Suggesting two alternative algorithm
  - Learner’s policy is slowly modified from executing expert’s policy