Reinforcement Learning in Buchberger's Algorithm

Dylan Peifer

Cornell University

5 April 2019

Outline

The efficiency of Buchberger's algorithm strongly depends on a choice of selection strategy. By phrasing Buchberger's algorithm as a reinforcement learning problem and applying standard reinforcement learning techniques we can learn new selection strategies that can match or beat the existing state-of-the-art.

1. Gröbner Bases and Buchberger's Algorithm
2. Reinforcement Learning and Policy Gradient
3. Preliminary Results

1. Gröbner Bases and Buchberger's Algorithm

$R = K[x_1, \ldots, x_n]$ a polynomial ring over some field K

$I = \langle f_1, \ldots, f_k \rangle \subseteq R$ an ideal generated by $f_1, \ldots, f_k \in R$

$R = K[x_1, \ldots, x_n]$ a polynomial ring over some field K

$I = \langle f_1, \ldots, f_k \rangle \subseteq R$ an ideal generated by $f_1, \ldots, f_k \in R$

Example

$$\begin{aligned} R &= \mathbb{Q}[x, y] \\ &= \{\textit{polynomials in } x \textit{ and } y \textit{ with rational coefficients}\} \end{aligned}$$

$$\begin{aligned} I &= \langle x^2 - y^3, xy^2 + x \rangle \\ &= \{a(x^2 - y^3) + b(xy^2 + x) : a, b \in R\} \end{aligned}$$

$R = K[x_1, \ldots, x_n]$ a polynomial ring over some field K

$I = \langle f_1, \ldots, f_k \rangle \subseteq R$ an ideal generated by $f_1, \ldots, f_k \in R$

Example

$$\begin{aligned} R &= \mathbb{Q}[x, y] \\ &= \{\textit{polynomials in } x \textit{ and } y \textit{ with rational coefficients}\} \end{aligned}$$

$$\begin{aligned} I &= \langle x^2 - y^3, xy^2 + x \rangle \\ &= \{a(x^2 - y^3) + b(xy^2 + x) : a, b \in R\} \end{aligned}$$

Question

In the above example, is $x^5 + x$ an element of I?

Question

Consider the ideal $I = \langle x^2 + x - 2 \rangle$ *in the ring* $\mathbb{Q}[x]$. *Is* $x^3 + 3x^2 + 5x + 4$ *an element of* I*?*

Question

Consider the ideal $I = \langle x^2 + x - 2 \rangle$ *in the ring* $\mathbb{Q}[x]$. *Is* $x^3 + 3x^2 + 5x + 4$ *an element of* I*?*

$$
\begin{array}{rrrrrrrrrrrr}
 & & & & & x & + & 2 & & & & \\
\cline{5-12}
x^2 & + & x & - & 2 & \big| & x^3 & + & 3x^2 & + & 5x & + & 4 \\
 & & & & & - & (x^3 & + & x^2 & - & 2x) & & \\
\cline{6-11}
 & & & & & & & & 2x^2 & + & 7x & + & 4 \\
 & & & & & & - & (2x^2 & + & 2x & - & 4) \\
\cline{7-12}
 & & & & & & & & & & 5x & + & 8
\end{array}
$$

Question

Consider the ideal $I = \langle x^2 + x - 2 \rangle$ *in the ring* $\mathbb{Q}[x]$. *Is* $x^3 + 3x^2 + 5x + 4$ *an element of* I*?*

$$
\begin{array}{rrrrrrrrrrrr}
 & & & & x & + & 2 & & & & & \\
\cline{5-12}
x^2 & + & x & - & 2 & \big| & x^3 & + & 3x^2 & + & 5x & + & 4 \\
 & & & & & - & (x^3 & + & x^2 & - & 2x) & & \\
\cline{6-11}
 & & & & & & & & 2x^2 & + & 7x & + & 4 \\
 & & & & & & & - & (2x^2 & + & 2x & - & 4) \\
\cline{8-12}
 & & & & & & & & & & 5x & + & 8
\end{array}
$$

$$x^3 + 3x^2 + 5x + 4 \quad = \quad (x + 2)(x^2 + x - 2) + (5x + 8)$$

Question

Consider the ideal $I = \langle x^2 + x - 2 \rangle$ *in the ring* $\mathbb{Q}[x]$. *Is* $x^3 + 3x^2 + 5x + 4$ *an element of* I*?*

$$
\begin{array}{rl}
 & x + 2 \\
\hline
x^2 + x - 2 \;\big|\; & x^3 + 3x^2 + 5x + 4 \\
 & -(x^3 + x^2 - 2x) \\
\hline
 & 2x^2 + 7x + 4 \\
 & -(2x^2 + 2x - 4) \\
\hline
 & 5x + 8
\end{array}
$$

$$x^3 + 3x^2 + 5x + 4 = (x + 2)(x^2 + x - 2) + (5x + 8)$$

$$\Longrightarrow \quad x^3 + 3x^2 + 5x + 4 \notin \langle x^2 + x - 2 \rangle$$

Definition

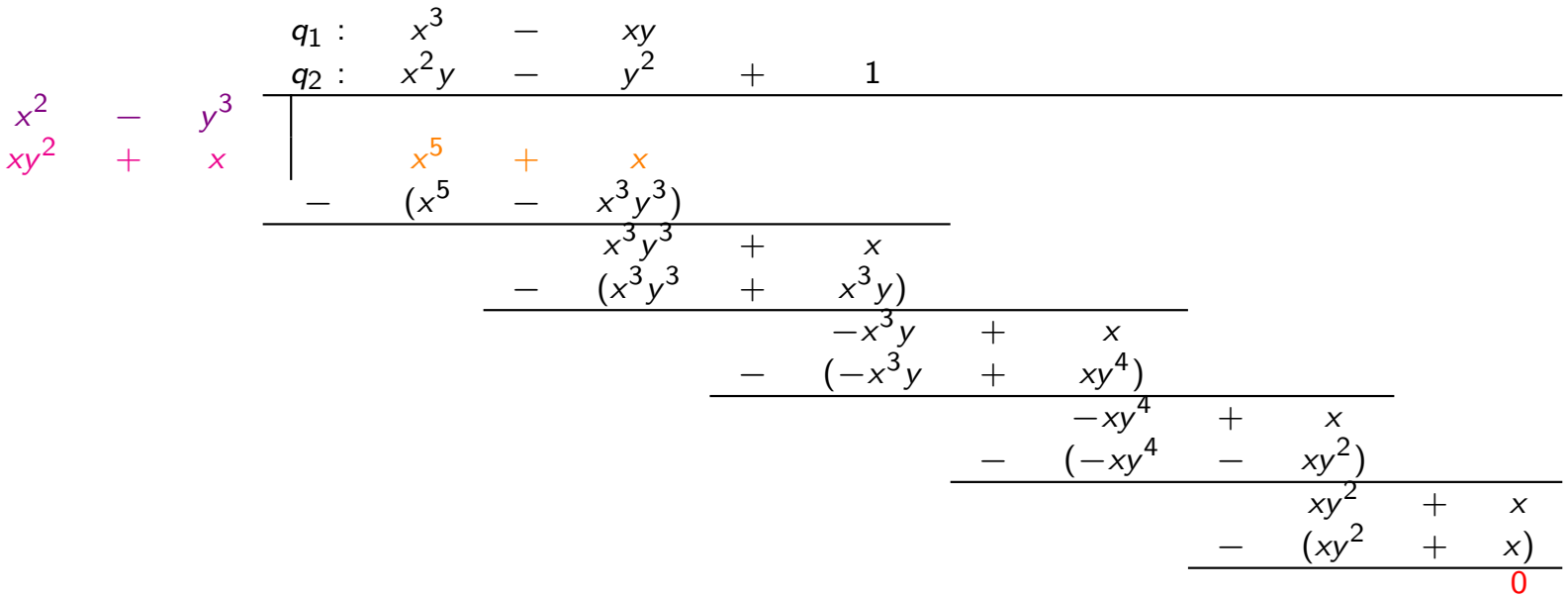
Let x^α denote an arbitrary monomial where α is the vector of exponents. A monomial order on $R = k[x_1, \ldots, x_n]$ is a relation $>$ on the monomials of R such that

1. *$>$ is a total ordering*
2. *$>$ is a well-ordering*
3. *if $x^\alpha > x^\beta$ then $x^\gamma x^\alpha > x^\gamma x^\beta$ for any x^γ (i.e., $>$ respects multiplication).*

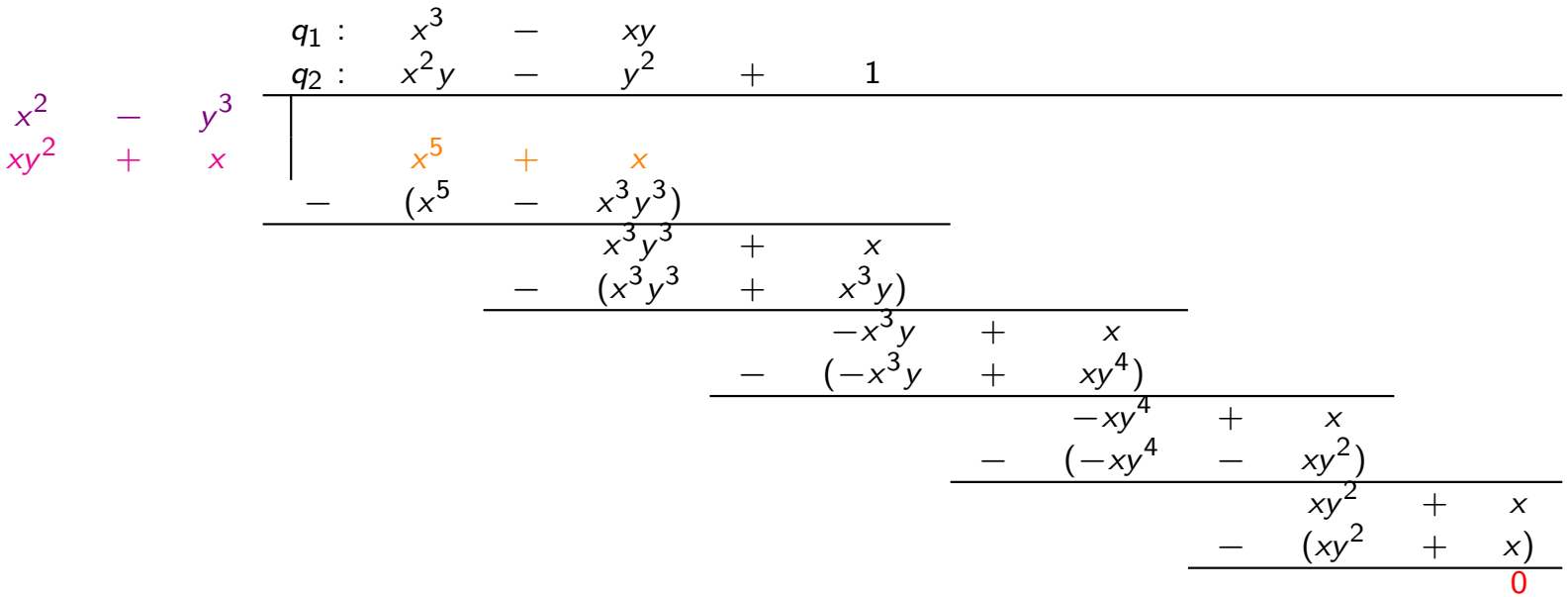
Example

Lexicographic order (lex) is defined by $\alpha > \beta$ if the leftmost nonzero component of $\alpha - \beta$ is positive. For example, $x > y > z$, $xy > y^4$, and $xz > y^2$.

Divide $x^5 + x$ by the generators $x^2 - y^3$ and $xy^2 + x$

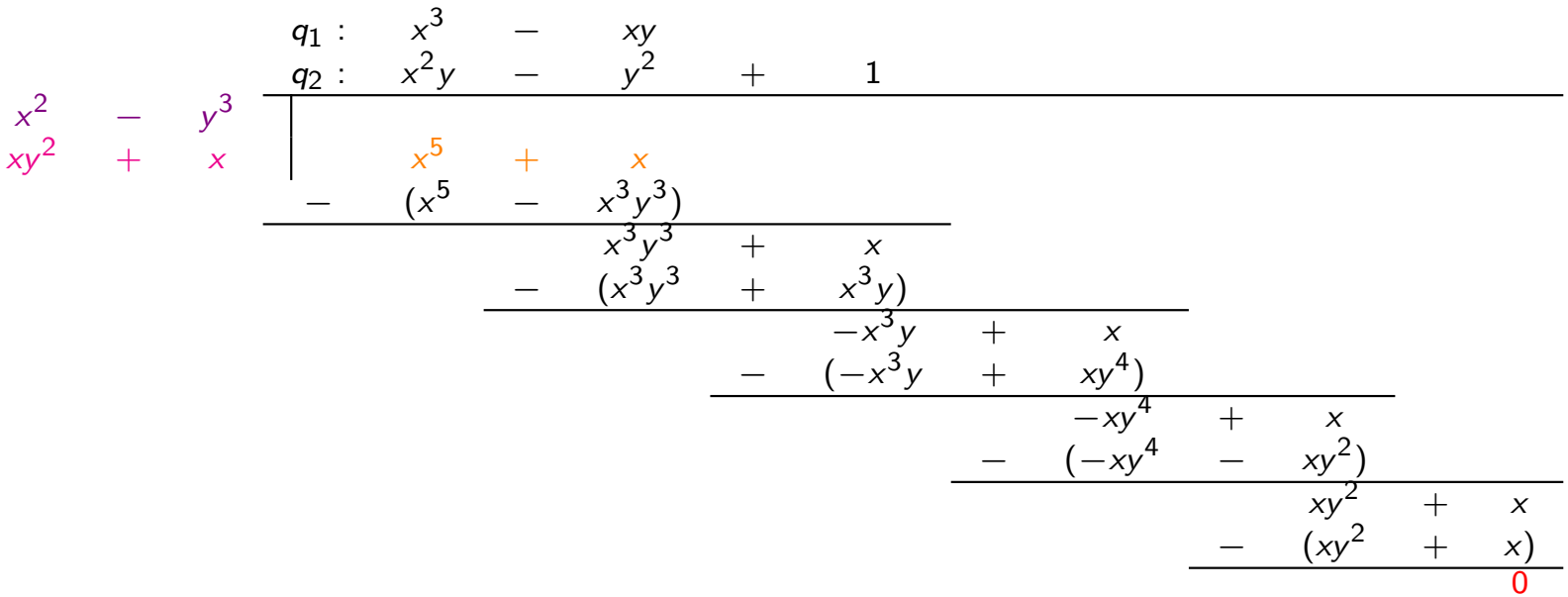


Divide $x^5 + x$ by the generators $x^2 - y^3$ and $xy^2 + x$



$$x^5 + x \quad = \quad (x^3 - xy)(x^2 - y^3) + (x^2y - y^2 + 1)(xy^2 + x)$$

Divide $x^5 + x$ by the generators $x^2 - y^3$ and $xy^2 + x$



$$x^5 + x = (x^3 - xy)(x^2 - y^3) + (x^2y - y^2 + 1)(xy^2 + x)$$

$$\Longrightarrow \quad x^5 + x \in \langle x^2 - y^3, xy^2 + x \rangle$$

Definition

When F is set of polynomials and dividing h by the $f_i \in F$ using the division algorithm leads to the remainder r we write $h^F \to r$ or say h reduces to r.

Definition

When F is set of polynomials and dividing h by the $f_i \in F$ using the division algorithm leads to the remainder r we write $h^F \to r$ or say h reduces to r.

Lemma

If $h^F \to 0$ then h is in the ideal generated by F.

Definition

When F is set of polynomials and dividing h by the $f_i \in F$ using the division algorithm leads to the remainder r we write $h^F \to r$ or say h reduces to r.

Lemma

If $h^F \to 0$ then h is in the ideal generated by F.

Unfortunately, the converse is false.

Example

Using the same ideal $I = \langle x^2 - y^3, xy^2 + x \rangle$, note that

$$y^2(x^2 - y^3) - x(xy^2 + x) \quad = \quad -x^2 - y^5 \quad \in \quad I$$

However, multivariate division produces the nonzero remainder $-y^5 - y^3$.

Definition

Given a monomial order, let $\mathrm{LT}(f)$ *be the leading term of* f*. Similarly, let* $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(f) \mid f \in I \rangle$ *be the ideal generated by all leading terms of* I*.*

Definition

Given a monomial order, a Gröbner basis G *of a nonzero ideal* I *is a subset* $\{g_1, g_2, \ldots, g_s\} \subseteq I$ *such that any of the following equivalent conditions hold:*

$$
\begin{aligned}
&(i) && f^G \to 0 \iff f \in I \\
&(ii) && f^G \text{ is unique for all } f \in R \\
&(iii) && \langle \mathrm{LT}(g_1), \mathrm{LT}(g_2), \ldots, \mathrm{LT}(g_s) \rangle = \langle \mathrm{LT}(I) \rangle
\end{aligned}
$$

Definition

Let $S(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)} f - \frac{x^\gamma}{\mathrm{LT}(g)} g$ *where* x^γ *is the least common multiple of the leading monomials of* f *and* g*. This is the* S-polynomial *of* f *and* g*, where* S *stands for subtraction or syzygy.*

Definition

Let $S(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)} f - \frac{x^\gamma}{\mathrm{LT}(g)} g$ *where* x^γ *is the least common multiple of the leading monomials of* f *and* g*. This is the* S-polynomial *of* f *and* g*, where* S *stands for subtraction or syzygy.*

Example

$$\begin{aligned} S(x^2 - y^3, xy^2 + x) &= \frac{x^2y^2}{x^2}(x^2 - y^3) - \frac{x^2y^2}{xy^2}(xy^2 + x) \\ &= y^2(x^2 - y^3) - x(xy^2 + x) \\ &= -x^2 - y^5 \end{aligned}$$

Definition

Let $S(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)} f - \frac{x^\gamma}{\mathrm{LT}(g)} g$ *where* x^γ *is the least common multiple of the leading monomials of* f *and* g*. This is the S-polynomial of* f *and* g*, where* S *stands for subtraction or syzygy.*

Example

$$\begin{aligned} S(x^2 - y^3, xy^2 + x) &= \frac{x^2y^2}{x^2}(x^2 - y^3) - \frac{x^2y^2}{xy^2}(xy^2 + x) \\ &= y^2(x^2 - y^3) - x(xy^2 + x) \\ &= -x^2 - y^5 \end{aligned}$$

Theorem (Buchberger's Criterion)

Let $G = \{g_1, g_2, \ldots, g_s\}$ *generate some ideal* I*. If* $S(g_i, g_j)^G \to 0$ *for all pairs* g_i, g_j *then* G *is a Gröbner basis of* I*.*

Algorithm 1 Buchberger's Algorithm

input a set of polynomials $\{f_1, \ldots, f_k\}$
output a Gröbner basis G of $I = \langle f_1, \ldots, f_k \rangle$
procedure $\textsc{Buchberger}(\{f_1, \ldots, f_k\})$
    $G \leftarrow \{f_1, \ldots, f_k\}$ ▷ the current basis
    $P \leftarrow \{(f_i, f_j) \mid 1 \leq i < j \leq k\}$ ▷ the remaining pairs
    while $|P| > 0$ **do**
        $(f_i, f_j) \leftarrow \text{select}(P)$
        $P \leftarrow P \setminus \{(f_i, f_j)\}$
        $r \leftarrow S(f_i, f_j)^G$
        if $r \neq 0$ **then**
            $P \leftarrow P \cup \{(f, r) : f \in G\}$
            $G \leftarrow G \cup \{r\}$
        end if
    end while
    return G
end procedure

Example

$I = \langle x^2 - y^3, xy^2 + x \rangle$

Example

$I = \langle x^2 - y^3, xy^2 + x \rangle$

initialize G *to* $\{x^2 - y^3, xy^2 + x\}$
initialize P *to* $\{(x^2 - y^3, xy^2 + x)\}$

Example

$I = \langle x^2 - y^3, xy^2 + x \rangle$

initialize G *to* $\{x^2 - y^3, xy^2 + x\}$
initialize P *to* $\{(x^2 - y^3, xy^2 + x)\}$

select $(x^2 - y^3, xy^2 + x)$ *and compute* $S(x^2 - y^3, xy^2 + x)^G \to -y^5 - y^3$
update G *to* $\{x^2 - y^3, xy^2 + x, -y^5 - y^3\}$
update P *to* $\{(x^2 - y^3, -y^5 - y^3), (xy^2 + x, -y^5 - y^3)\}$

Example

$I = \langle x^2 - y^3, xy^2 + x \rangle$

initialize G *to* $\{x^2 - y^3, xy^2 + x\}$
initialize P *to* $\{(x^2 - y^3, xy^2 + x)\}$

select $(x^2 - y^3, xy^2 + x)$ *and compute* $S(x^2 - y^3, xy^2 + x)^G \to -y^5 - y^3$
update G *to* $\{x^2 - y^3, xy^2 + x, -y^5 - y^3\}$
update P *to* $\{(x^2 - y^3, -y^5 - y^3), (xy^2 + x, -y^5 - y^3)\}$

select $(x^2 - y^3, -y^5 - y^3)$ *and compute* $S(x^2 - y^3, -y^5 - y^3)^G \to 0$

Example

$I = \langle x^2 - y^3, xy^2 + x \rangle$

initialize G *to* $\{x^2 - y^3, xy^2 + x\}$
initialize P *to* $\{(x^2 - y^3, xy^2 + x)\}$

select $(x^2 - y^3, xy^2 + x)$ *and compute* $S(x^2 - y^3, xy^2 + x)^G \to -y^5 - y^3$
update G *to* $\{x^2 - y^3, xy^2 + x, -y^5 - y^3\}$
update P *to* $\{(x^2 - y^3, -y^5 - y^3), (xy^2 + x, -y^5 - y^3)\}$

select $(x^2 - y^3, -y^5 - y^3)$ *and compute* $S(x^2 - y^3, -y^5 - y^3)^G \to 0$

select $(xy^2 + x, -y^5 - y^3)$ *and compute* $S(xy^2 + x, -y^5 - y^3)^G \to 0$

Example

$I = \langle x^2 - y^3, xy^2 + x \rangle$

initialize G *to* $\{x^2 - y^3, xy^2 + x\}$
initialize P *to* $\{(x^2 - y^3, xy^2 + x)\}$

select $(x^2 - y^3, xy^2 + x)$ *and compute* $S(x^2 - y^3, xy^2 + x)^G \to -y^5 - y^3$
update G *to* $\{x^2 - y^3, xy^2 + x, -y^5 - y^3\}$
update P *to* $\{(x^2 - y^3, -y^5 - y^3), (xy^2 + x, -y^5 - y^3)\}$

select $(x^2 - y^3, -y^5 - y^3)$ *and compute* $S(x^2 - y^3, -y^5 - y^3)^G \to 0$

select $(xy^2 + x, -y^5 - y^3)$ *and compute* $S(xy^2 + x, -y^5 - y^3)^G \to 0$

return $G = \{x^2 - y^3, xy^2 + x, -y^5 - y^3\}$

Algorithm 2 Buchberger's Algorithm

input a set of polynomials $\{f_1, \ldots, f_k\}$
output a Gröbner basis G of $I = \langle f_1, \ldots, f_k \rangle$
procedure BUCHBERGER($\{f_1, \ldots, f_k\}$)
    $G \leftarrow \{f_1, \ldots, f_k\}$ ▷ the current basis
    $P \leftarrow \{(f_i, f_j) \mid 1 \leq i < j \leq k\}$ ▷ the remaining pairs
    while $|P| > 0$ **do**
        $(f_i, f_j) \leftarrow \text{select}(P)$
        $P \leftarrow P \setminus \{(f_i, f_j)\}$
        $r \leftarrow S(f_i, f_j)^G$
        if $r \neq 0$ **then**
            $P \leftarrow P \cup \{(f, r) : f \in G\}$
            $G \leftarrow G \cup \{r\}$
        end if
    end while
    return G
end procedure

In general, we should select "small" pairs (f_i, f_j) first.

In general, we should select "small" pairs (f_i, f_j) first.

- First:
  among the pairs with minimal j, pick the pair with smallest i
- Degree:
  pick the pair with smallest degree of $\mathrm{lcm}(\mathrm{LT}(f_i), \mathrm{LT}(f_j))$
- Normal:
  pick the pair with smallest $\mathrm{lcm}(\mathrm{LT}(f_i), \mathrm{LT}(f_j))$ in the monomial order
- Sugar:
  pick the pair with smallest sugar degree of $\mathrm{lcm}(\mathrm{LT}(f_i), \mathrm{LT}(f_j))$, which is the degree it would have had if we had homogenized at the beginning

The number of pair reductions performed is a rough estimate of how much time was spent. Smaller numbers are better.

example	First	Normal	Sugar	Random	Last	Strange	Spice
cyclic4	11	11	11	14	21	23	23
reimer3	25	23	25	25	25	28	28
katsura5	28	28	28	44	76	86	86
eco6	67	61	64	97	149	295	295
noon4	71	71	71	100	103	375	375
cyclic6	366	620	343	793			
katsura7	164	164	164	285			
katsura4-lex	25	46	19	29	44	30	59
eco5-lex	30	22	26	28	91	32	97
cyclic5-lex	104	1602	108				

Summary

- A Gröbner basis of an ideal in a polynomial ring is a special generating set that is useful for many computational problems. Buchberger's algorithm can produce a Gröbner basis from any initial generating set of an ideal.
- Buchberger's algorithm works by repeatedly choosing pairs (f_i, f_j) of the current generating set and adding the reduction of the s-polynomial of f_i and f_j to the generating set if it is not zero.
- The selection strategy used to pick which pair to choose next can make a big difference in the efficiency of Buchberger's algorithm.

2. Reinforcement Learning and Policy Gradient

Reinforcement learning is the study of methods for learning how to act in order to maximize reward.

Alternatively, reinforcement learning tries to understand and optimize goal-directed behavior driven by interaction with the world.

- playing games (backgammon, chess, Go, StarCraft, ...)
- flying a helicopter or driving a car
- controlling a power station or data center
- managing a portfolio of stocks or other financial assets
- allocating resources to research projects

Key features of a reinforcement learning problem include

- no supervisor, instead we learn from our own experience
- rewards can be sparse or delayed
- current actions influence future conditions and options

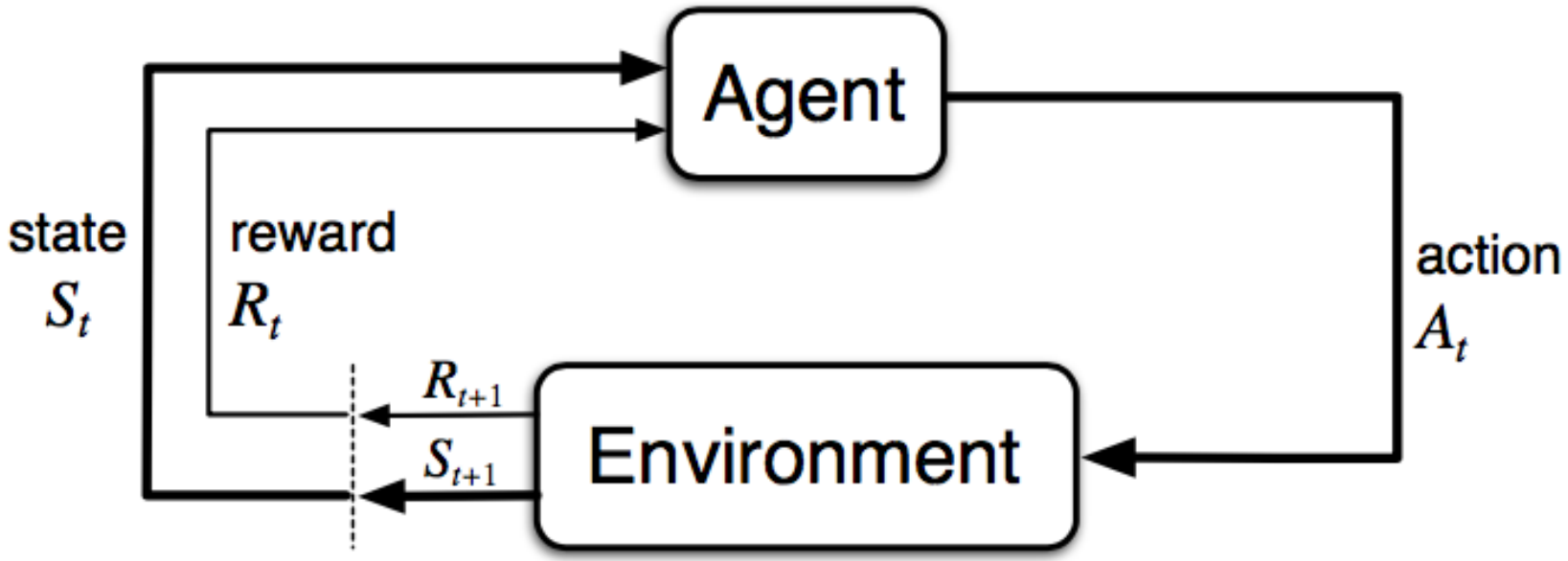
Key features of a reinforcement learning problem include

- no supervisor, instead we learn from our own experience
- rewards can be sparse or delayed
- current actions influence future conditions and options

and key issues are

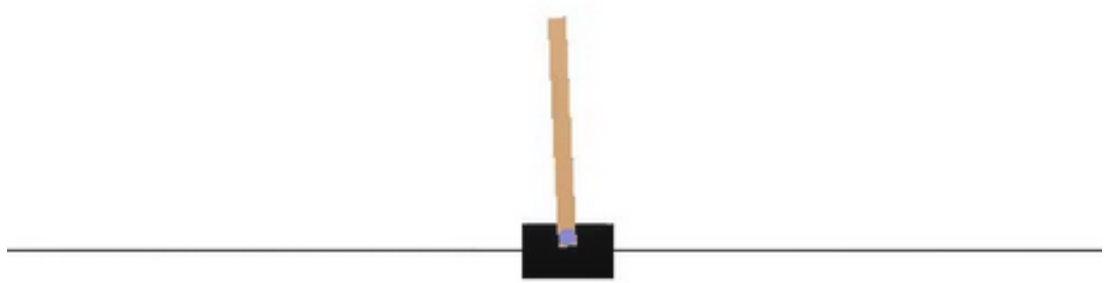
- Credit Assignment: After we receive a reward, how do we determine which action was responsible for it?
- Explore/Exploit Tradeoff: How do we balance trying new things and taking advantage of what we already know?

Reinforcement learning problems can be phrased as the interaction of an agent and an environment.



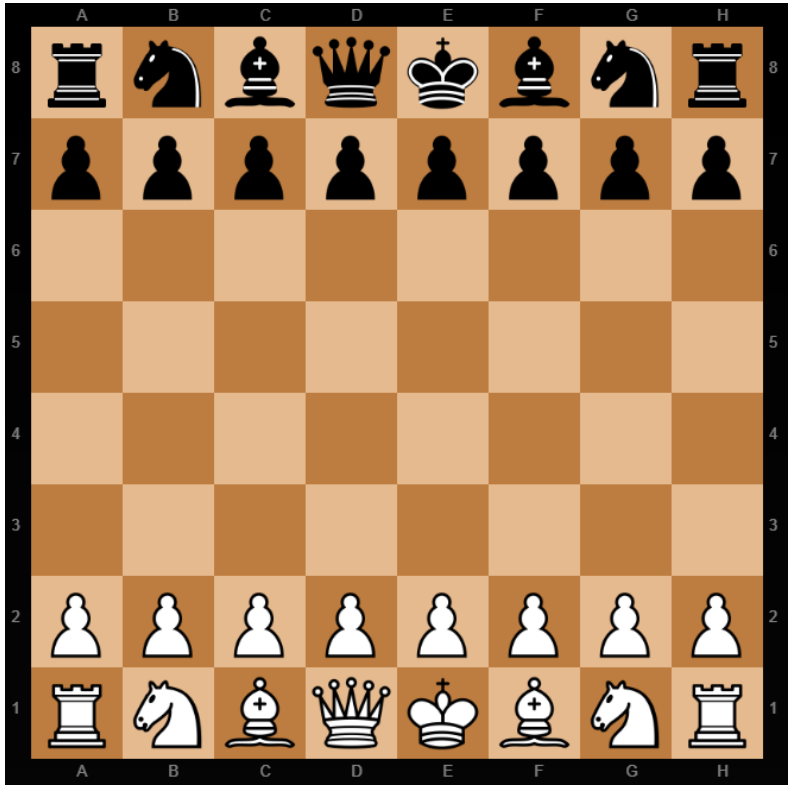
The agent chooses actions and the environment processes actions and gives back the updated state and a reward. The agent wants to maximize its return, which is the amount of reward it gets in the long run.

CartPole



State: the cart and pole positions and velocities
Action: push the cart left or right
Reward: 1 for every transition the pole is still upright

Chess



State: the positions of all pieces on the board

Action: a valid move of one of your pieces

Reward: 1 if you win immediately after the transition, otherwise 0

Buchberger

$$G = \{x^2 - y^3, xy^2 + x, -y^5 - y^3\}$$
$$P = \{(x^2 - y^3, -y^5 - y^3), (xy^2 + x, -y^5 - y^3)\}$$

State: the current basis and pair set
Action: a pair from the pair set
Reward: -1 for every transition until the pair set is empty

Definition

A policy π is a function

$$\pi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$$
$$\pi(a|s) = Pr(A_t = a|S_t = s)$$

which maps state-action pairs to the probability of choosing the given action in the given state.

Definition

A policy π is a function

$$\pi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$$
$$\pi(a|s) = Pr(A_t = a|S_t = s)$$

which maps state-action pairs to the probability of choosing the given action in the given state.

Policies are often viewed as functions that take in a state and return a probability distribution on actions. An agent follows a policy by applying the policy to its current state and sampling from the returned probability distribution to choose the next action.

Definition

A trajectory or rollout τ of a policy π is a series of states, actions, and rewards $(S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots, R_T, S_T)$ obtained by following the policy π one time through the environment.

Definition

The return of a trajectory is the sum of rewards

$$\sum_{t=1}^{T} R_t$$

along the trajectory.

Definition

A trajectory or rollout τ of a policy π is a series of states, actions, and rewards $(S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots, R_T, S_T)$ obtained by following the policy π one time through the environment.

Definition

The return of a trajectory is the sum of rewards

$$\sum_{t=1}^{T} R_t$$

along the trajectory.

Given an environment, the goal of reinforcement learning is to find a policy π that maximizes the expected return

$$\mathop{\mathbb{E}}_{\tau \sim \pi} \left[\sum_{t=1}^{T} R_t \right].$$

Consider a parametrized policy function π_θ which maps states to probability distributions on actions. The expected return is now a function

$$J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} R_t \right]$$

of the parameters θ of the policy.

Consider a parametrized policy function π_θ which maps states to probability distributions on actions. The expected return is now a function

$$J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} R_t \right]$$

of the parameters θ of the policy.

Starting from any value of the parameters θ_1, we can improve the policy by repeatedly moving the parameters in the direction of $\nabla_\theta J(\theta)$.

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta)|_{\theta_k}$$

Theorem (Policy Gradient Theorem)

Suppose π_θ is a parametrized policy that is differentiable with respect to its parameters θ. Then the gradient of

$$J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} R_t \right]$$

is

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t|S_t) \sum_{t'=t+1}^{T} R_{t'} \right].$$

Theorem (Policy Gradient Theorem)

Suppose π_θ is a parametrized policy that is differentiable with respect to its parameters θ. Then the gradient of

$$J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} R_t \right]$$

is

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(A_t | S_t) \sum_{t'=t+1}^{T} R_{t'} \right].$$

Intuitively, we should increase the probability of taking the action we chose proportional to the future reward we received and the derivative of the log probability of choosing that action again.

Summary

- Reinforcement learning can be phrased as the interaction of an agent and an environment, where an agent picks actions and is trying to maximize the total reward it receives from the environment over a full trajectory.
- Buchberger's algorithm is a reinforcement learning problem with state the current basis/pairs and action a choice of pair to reduce (i.e., each pass through the `while` loop is a state transition).
- Policy gradient methods improve a parametrized policy by moving the parameters in the direction of the gradient of expected return.

3. Preliminary Results

Example 1: 5 Binomial Quadrics

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random quadrics.

Example 1: 5 Binomial Quadrics

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random quadrics.

Example

$I = \langle xy + 31398y^2, x^2 + 15976y^2, xy + 3328xz, y^2 + 18836z^2, yz + 10816z^2 \rangle$

Example 1: 5 Binomial Quadrics

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random quadrics.

Example

$I = \langle xy + 31398y^2, x^2 + 15976y^2, xy + 3328xz, y^2 + 18836z^2, yz + 10816z^2 \rangle$

In this setting we know Random selection has average return -37. Degree, Normal, and Sugar selection all have average return -21.

Example 1: 5 Binomial Quadrics

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random quadrics.

Example

$I = \langle xy + 31398y^2, x^2 + 15976y^2, xy + 3328xz, y^2 + 18836z^2, yz + 10816z^2 \rangle$

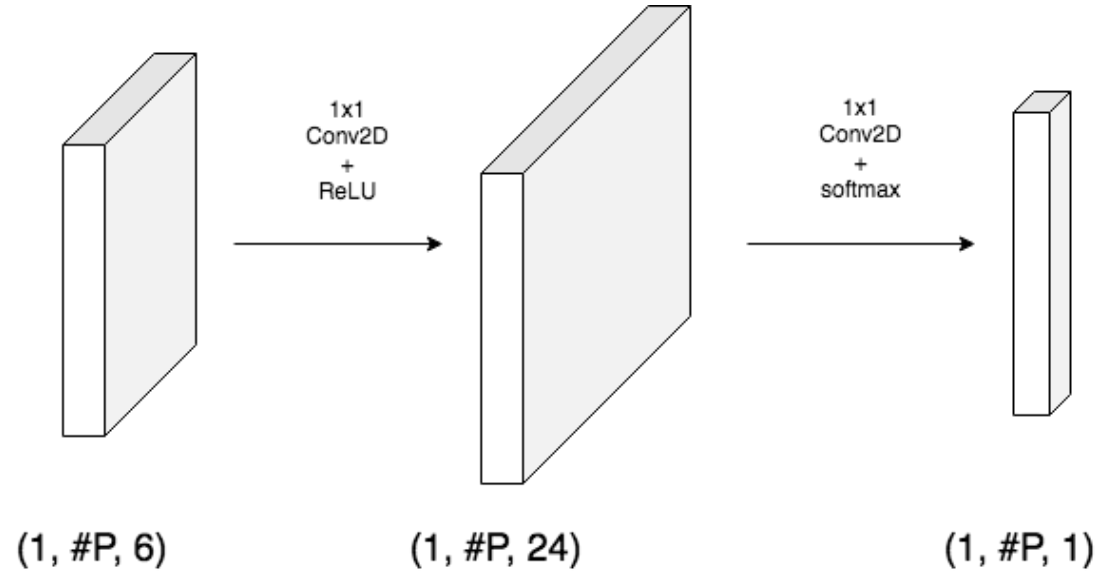
In this setting we know Random selection has average return -37. Degree, Normal, and Sugar selection all have average return -21. WARNING: These returns are without any pair elimination.

$$G = \{xy+31398y^2, x^2+15976y^2, xy+3328xz, y^2+18836z^2, yz+10816z^2\}$$
$$P = \{(1,2),(1,3),(2,3),(1,4),(2,4),(3,4),(1,5),(2,5),(3,5),(4,5)\}$$

$G = \{xy+31398y^2, x^2+15976y^2, xy+3328xz, y^2+18836z^2, yz+10816z^2\}$
$P = \{(1,2),(1,3),(2,3),(1,4),(2,4),(3,4),(1,5),(2,5),(3,5),(4,5)\}$

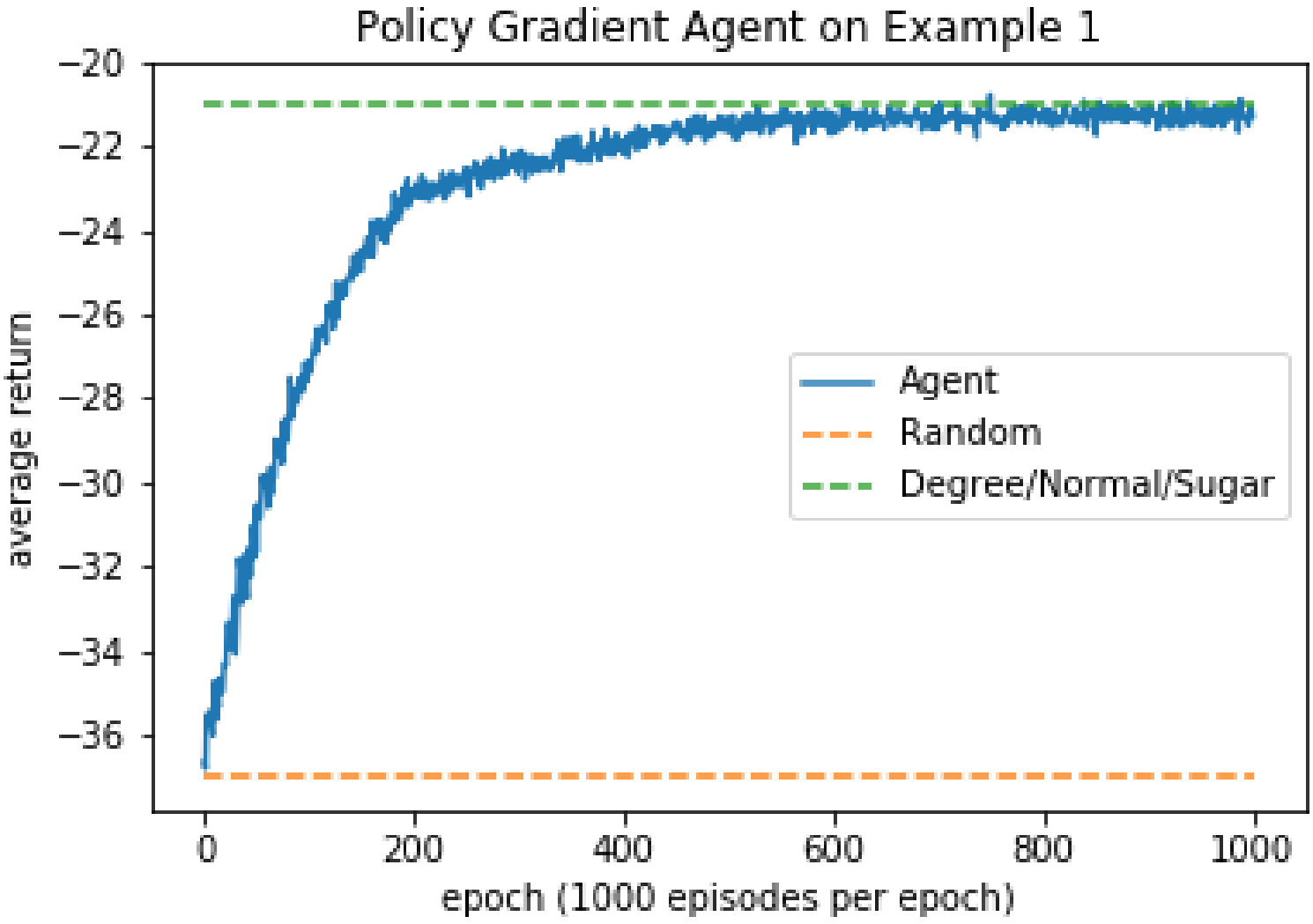
Concatenate exponent vectors of the lead terms in each pair. Place each pair in the row of a matrix.

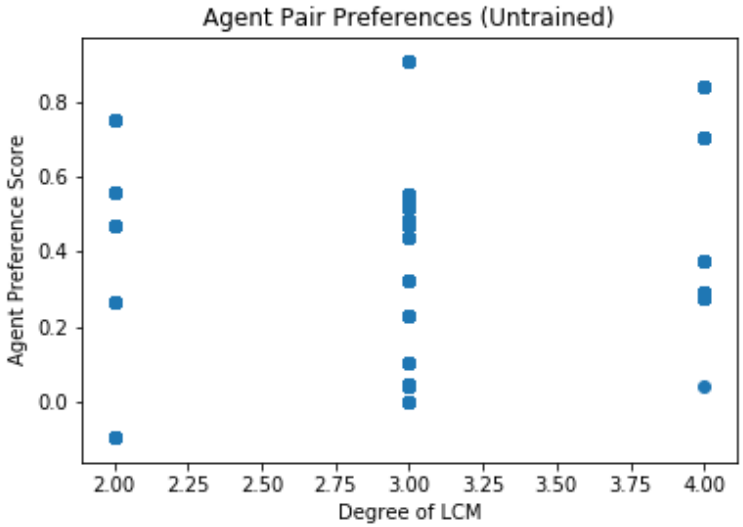
$$\rightarrow \begin{bmatrix} 1 & 1 & 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 2 & 0 & 0 & 1 & 1 \end{bmatrix}$$

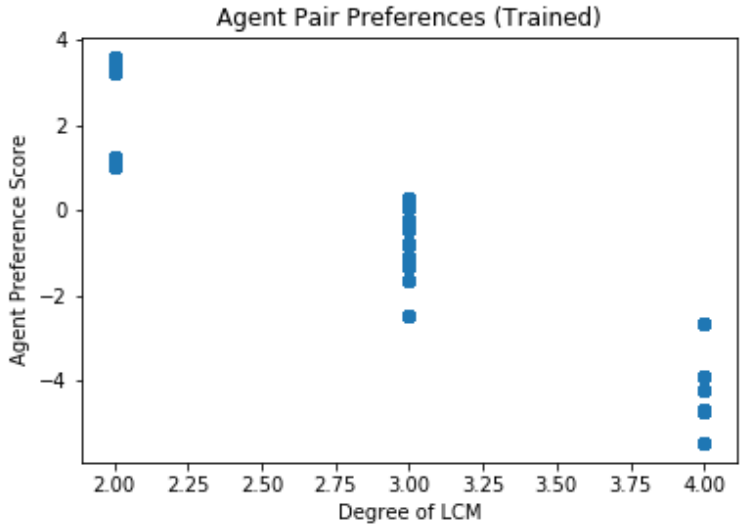


Trainable Parameters: 193
Optimizer: Adam with learning rate 0.00001
Training Time: 8 hours

In each epoch we perform 1000 rollouts, compute future rewards for each state on each trajectory, baseline by the size of the current pair set in each state, and normalize these scores before performing the policy gradient step.

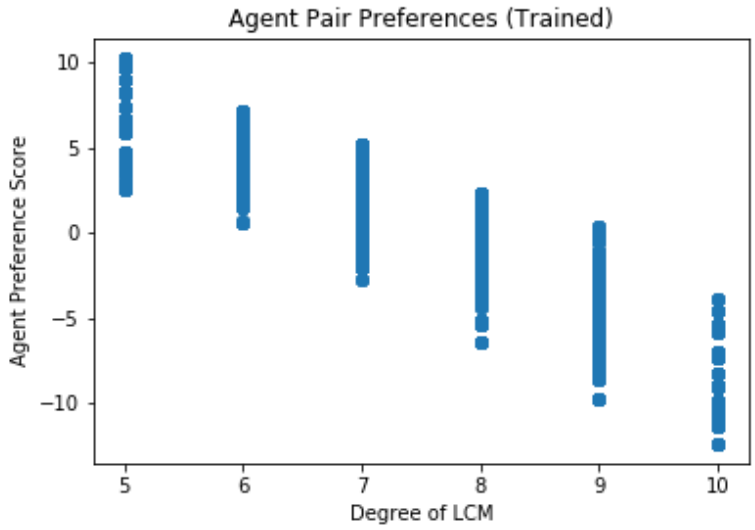

Policy Gradient Agent on Example 1
average return
−20
−22
−24
−26
−28
−30
−32
−34
−36
0
200
400
600
800
1000
epoch (1000 episodes per epoch)
Agent
Random
Degree/Normal/Sugar

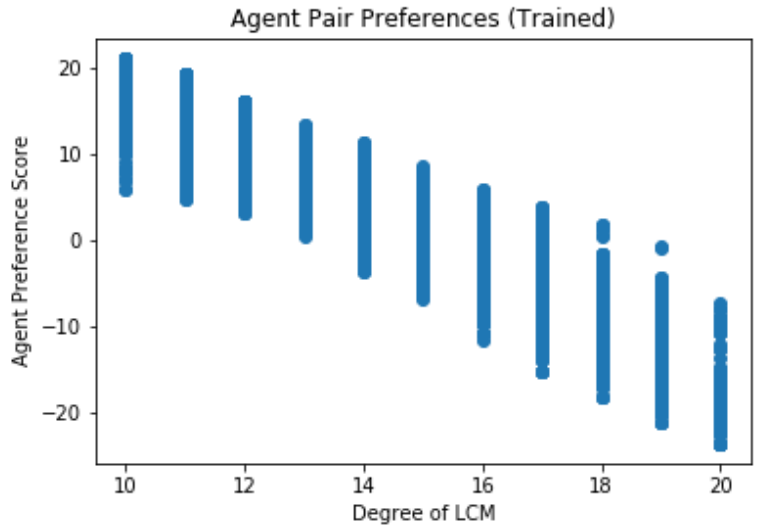




Before training there is no relation between the degree of a pair and the agent's preference.

After training the agent clearly prefers pairs that have smaller degree.





5 Binomials of Degree 5:
Random: -221
Degree/Normal/Sugar: -54.7
Agent: -71.5

5 Binomials of Degree 10:
Random: -1260
Degree/Normal/Sugar: -125
Agent: -319

Example 2: 5 Inhomogeneous Binomials of degree ≤ 7

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random binomials of degree less than or equal to 7.

Example 2: 5 Inhomogeneous Binomials of degree ≤ 7

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random binomials of degree less than or equal to 7.

Example

$I =$
$\langle xy^6 + 9y^2z^4, z^4 + 1212z, xy^3 + 961xy^2, x^4yz + 12518xz, xyz^2 + 20y \rangle$

Example 2: 5 Inhomogeneous Binomials of degree ≤ 7

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random binomials of degree less than or equal to 7.

Example

$I =$
$\langle xy^6 + 9y^2z^4, z^4 + 1212z, xy^3 + 961xy^2, x^4yz + 12518xz, xyz^2 + 20y \rangle$

In this setting we know Random selection has average return -175. Degree, Normal, and Sugar selection have average returns -117, -126, -130, respectively.

Example 2: 5 Inhomogeneous Binomials of degree ≤ 7

Consider $R = \mathbb{Z}/32003[x, y, z]$, grevlex ordering, and ideals I generated by 5 random binomials of degree less than or equal to 7.

Example

$I =$
$\langle xy^6 + 9y^2z^4, z^4 + 1212z, xy^3 + 961xy^2, x^4yz + 12518xz, xyz^2 + 20y \rangle$

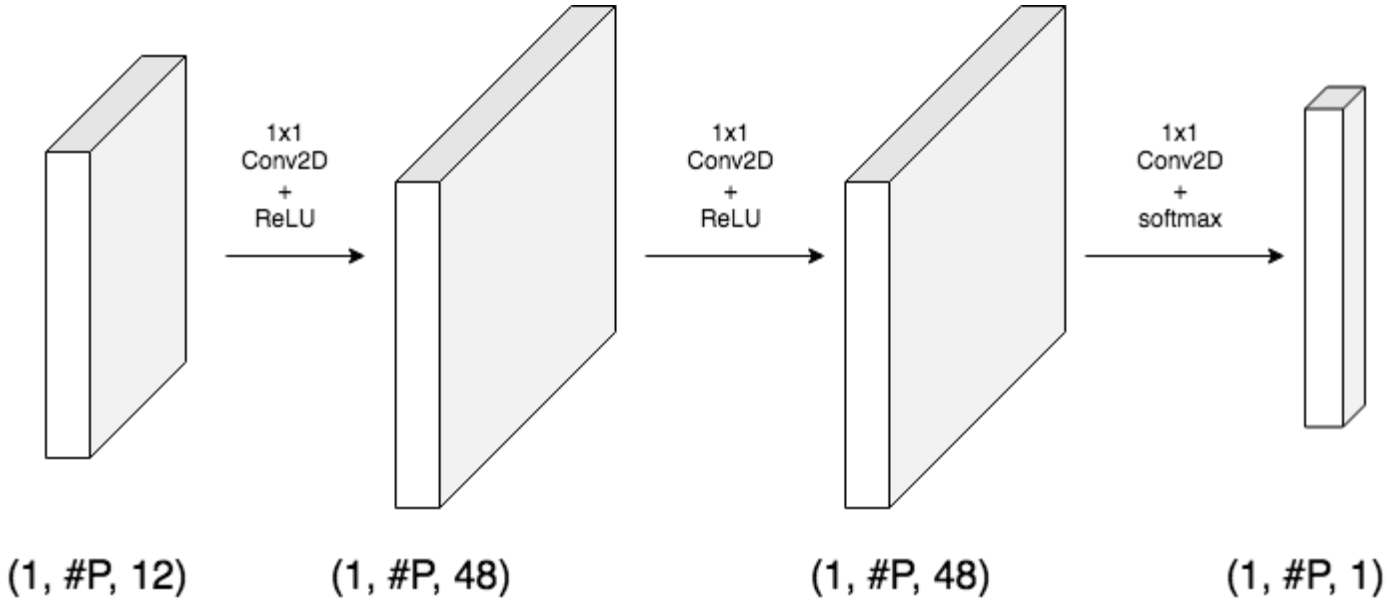
In this setting we know Random selection has average return -175. Degree, Normal, and Sugar selection have average returns -117, -126, -130, respectively.
WARNING: These returns are without any pair elimination.

$$G = \{xy^6 + 9y^2z^4, z^4 + 1212z, xy^3 + 961xy^2, x^4yz + 12518xz, xyz^2 + 20y\}$$
$$P = \{(1,2),(1,3),(2,3),(1,4),(2,4),(3,4),(1,5),(2,5),(3,5),(4,5)\}$$

$$G = \{xy^6 + 9y^2z^4, z^4 + 1212z, xy^3 + 961xy^2, x^4yz + 12518xz, xyz^2 + 20y\}$$
$$P = \{(1,2),(1,3),(2,3),(1,4),(2,4),(3,4),(1,5),(2,5),(3,5),(4,5)\}$$

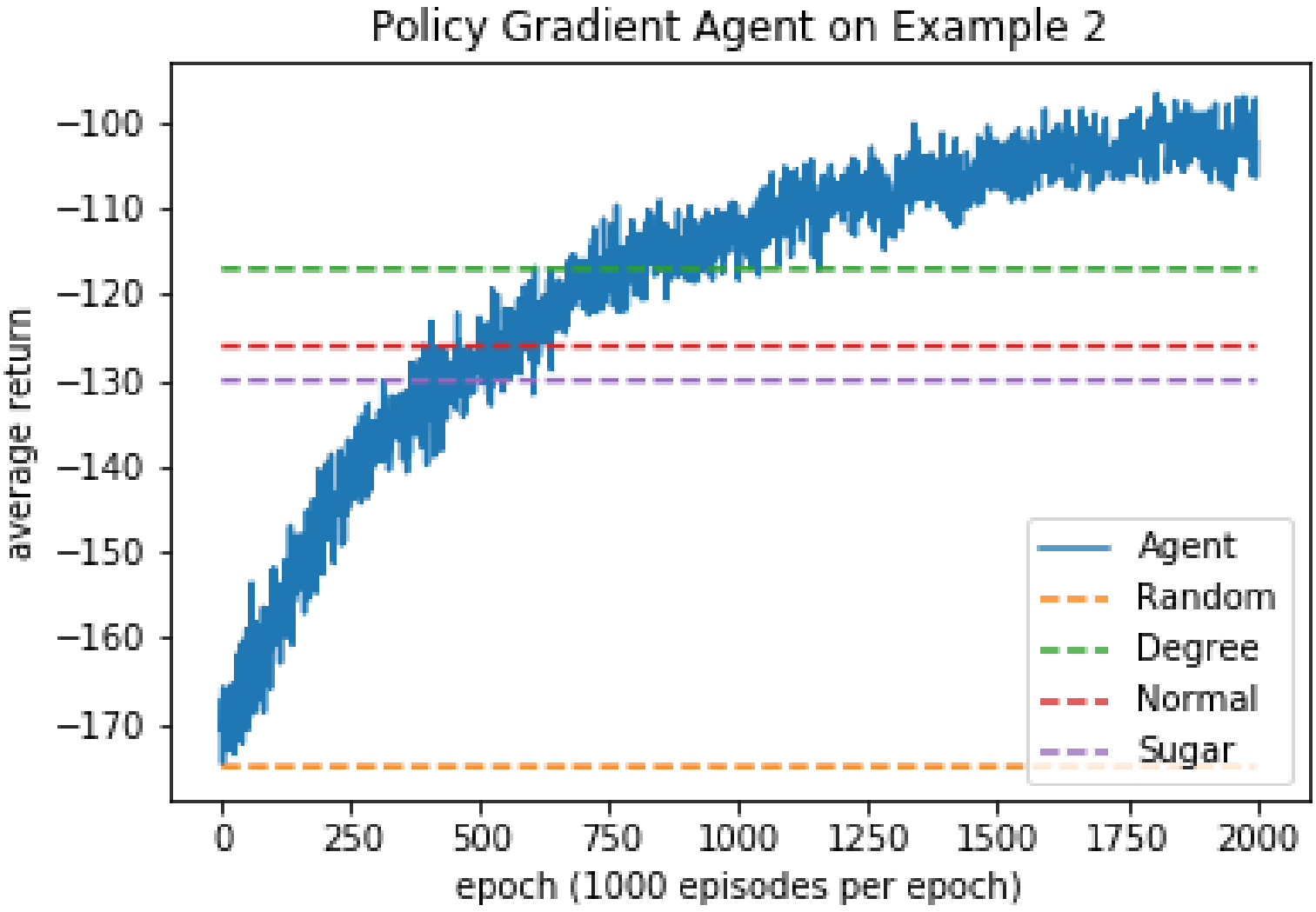
Concatenate exponent vectors of **all** terms in each pair. Place each pair in the row of a matrix.

$$\rightarrow \begin{bmatrix}
1 & 6 & 0 & 0 & 2 & 4 & 0 & 0 & 4 & 0 & 0 & 1 \\
1 & 6 & 0 & 0 & 2 & 4 & 1 & 3 & 0 & 1 & 2 & 0 \\
0 & 0 & 4 & 0 & 0 & 1 & 1 & 3 & 0 & 1 & 2 & 0 \\
1 & 6 & 0 & 0 & 2 & 4 & 4 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 4 & 0 & 0 & 1 & 4 & 1 & 1 & 1 & 0 & 1 \\
1 & 3 & 0 & 1 & 2 & 0 & 4 & 1 & 1 & 1 & 0 & 1 \\
1 & 6 & 0 & 0 & 2 & 4 & 1 & 1 & 2 & 0 & 1 & 0 \\
0 & 0 & 4 & 0 & 0 & 1 & 1 & 1 & 2 & 0 & 1 & 0 \\
1 & 3 & 0 & 1 & 2 & 0 & 1 & 1 & 2 & 0 & 1 & 0 \\
4 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 2 & 0 & 1 & 0
\end{bmatrix}$$



Trainable Parameters: 3025
Optimizer: Adam with learning rate 0.00001
Training Time: 72 hours

In each epoch we perform 1000 rollouts, compute future rewards for each state on each trajectory, baseline by the size of the current pair set in each state, and normalize these scores before performing the policy gradient step.


Policy Gradient Agent on Example 2
average return
−100
−110
−120
−130
−140
−150
−160
−170
0
250
500
750
1000
1250
1500
1750
2000
epoch (1000 episodes per epoch)
Agent
Random
Degree
Normal
Sugar

Summary

- In the binomial quadric example, a policy gradient agent that only saw lead terms learned a strategy that approximates degree selection.
- In the general binomial example, a policy gradient agent that saw the full binomials learned a strategy that performs better than Degree, Normal, or Sugar.
- While some performance does transfer, policy gradient agents do have trouble generalizing to ideals significantly different from those seen in training.