

Aprendizagem por Reforço aplicada a Seleção de Comportamento em Robótica Móvel

Tiago Pimentel Martins da Silva

Universidade de Brasília

tiagopms@gmail.com

December 8, 2014

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

A automação pode ser resumida por uma pergunta:

A automação pode ser resumida por uma pergunta: *O que devo fazer a seguir, sabendo o que já vi e o que fiz?*

A automação pode ser resumida por uma pergunta: *O que devo fazer a seguir, sabendo o que já vi e o que fiz?*

Nesse trabalho, tentaremos responder essa pergunta utilizando:

A automação pode ser resumida por uma pergunta: *O que devo fazer a seguir, sabendo o que já vi e o que fiz?*

Nesse trabalho, tentaremos responder essa pergunta utilizando:

- Planejamento de tarefas;

A automação pode ser resumida por uma pergunta: *O que devo fazer a seguir, sabendo o que já vi e o que fiz?*

Nesse trabalho, tentaremos responder essa pergunta utilizando:

- Planejamento de tarefas;
- Redes bayesianas;

A automação pode ser resumida por uma pergunta: *O que devo fazer a seguir, sabendo o que já vi e o que fiz?*

Nesse trabalho, tentaremos responder essa pergunta utilizando:

- Planejamento de tarefas;
- Redes bayesianas;
- Aprendizagem por reforço.

1 Motivação

- Contextualização
- **Trabalhos anteriores**

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

Koike (2008) [1]

- Seleção de comportamentos;
- Redes bayesianas.

Koike (2008) [1]

- Seleção de comportamentos;
- Redes bayesianas.

Lidoris (2011) [2]

- Seleção de comportamentos;
- Redes bayesianas;
- Aprendizagem por demonstração.

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

- Estimar a distribuição conjunta de probabilidade:

$$P(M^{0:t} S^{0:t} Z^{0:t} | \pi_f)$$

- Estimar a distribuição conjunta de probabilidade;
- Pode ser descrita por:

$$P(M^{0:t} S^{0:t} Z^{0:t} \mid \pi_f) = P(M^0 S^0 Z^0 \mid \pi_f) \cdot \prod_{j=1}^t \left(\begin{array}{l} P(S^j \mid S^{j-1} M^{j-1} \pi_f) \\ \times P(Z^j \mid S^j \pi_f) \\ \times P(M^j \mid S^j M^{j-1} \pi_f) \end{array} \right)$$

Dividido em três passos:

- Predição:

$$P(S^t \mid z^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1}} \left(\begin{array}{l} P(S^t \mid S^{t-1} m^{t-1} \pi_f) \\ \times P(m^{t-1} \mid S^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} \mid z^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

Dividido em três passos:

- Predição:

$$P(S^t \mid z^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1}} \left(\begin{array}{l} P(S^t \mid S^{t-1} m^{t-1} \pi_f) \\ \times P(m^{t-1} \mid S^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} \mid z^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

- Observação:

$$P(S^t \mid z^{0:t} m^{0:t-1} \pi_f) \propto \left(\begin{array}{l} P(z^t \mid S^t \pi_f) \\ \times P(S^t \mid z^{0:t-1} m^{0:t-1} \pi_f) \end{array} \right)$$

Dividido em três passos:

- Predição:

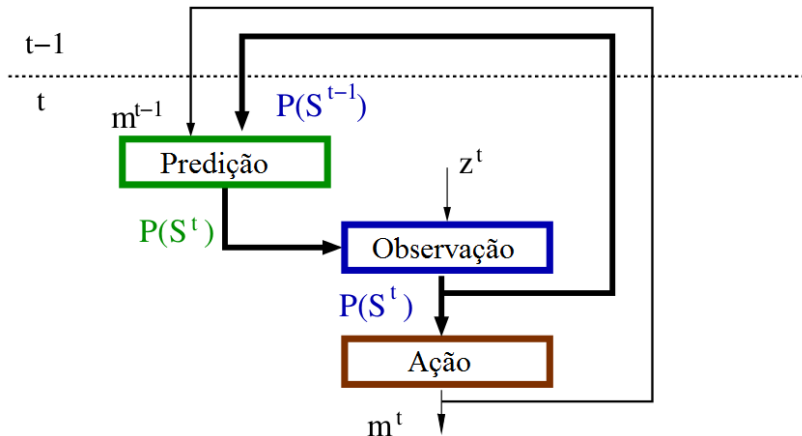
$$P(S^t | z^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1}} \left(\begin{array}{l} P(S^t | S^{t-1} m^{t-1} \pi_f) \\ \times P(m^{t-1} | S^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} | z^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

- Observação:

$$P(S^t | z^{0:t} m^{0:t-1} \pi_f) \propto \left(\begin{array}{l} P(z^t | S^t \pi_f) \\ \times P(S^t | z^{0:t-1} m^{0:t-1} \pi_f) \end{array} \right)$$

- Escolha de ação motora:

$$P(M^t | z^{0:t} m^{0:t-1} \pi_f) \propto \sum_{S^t} \left(\begin{array}{l} P(M^t | S^t m^{t-1} \pi_f) \\ \times P(S^t | z^{0:t} m^{0:t-1} \pi_f) \end{array} \right)$$



Filtro Bayesiano Recursivo. Fonte: [3]

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

- Base matemática para o modelamento de tomada de decisões;

- Base matemática para o modelamento de tomada de decisões;
- Extensão de cadeias de Markov;

- Base matemática para o modelamento de tomada de decisões;
- Extensão de cadeias de Markov;
- Possui ações e recompensas (escolha e motivação);

- Base matemática para o modelamento de tomada de decisões;
- Extensão de cadeias de Markov;
- Possui ações e recompensas (escolha e motivação);
- Cálculo recursivo.

Necessário:

- Modelo de atuação:

$$P(s'|u, s)$$

Necessário:

- Modelo de atuação:

$$P(s'|u, s)$$

- Modelo de recompensas:

$$r(s, u, s')$$

- Inicialmente, calcula-se um valor de ganho (analisando apenas a próxima ação):

$$V_1(s) = \max_u \left(\int r(s, u, s') \cdot P(s' | u, s) ds' \right)$$

- Inicialmente, calcula-se um valor de ganho (analisando apenas a próxima ação):

$$V_1(s) = \max_u \left(\int r(s, u, s') \cdot P(s' | u, s) ds' \right)$$

- E escolhe-se uma ação que maxime esse ganho:

$$\pi_1(s) = \operatorname{argmax}_u \left(\int r(s, u, s') \cdot P(s' | u, s) ds' \right)$$

- Depois, calcula-se esse valor de ganho iterativamente (analisando mais ações futuras):

$$V_j(s) = \max_u \left(\int (r(s, u, s') + \gamma \cdot V_{j-1}(s')) \cdot P(s' | u, s) \, ds' \right)$$

- Depois, calcula-se esse valor de ganho iterativamente (analisando mais ações futuras):

$$V_j(s) = \max_u \left(\int (r(s, u, s') + \gamma \cdot V_{j-1}(s')) \cdot P(s' | u, s) \, ds' \right)$$

- E escolhe-se uma ação que maxime esse ganho:

$$\pi_j(s) = \operatorname{argmax}_u \left(\int (r(s, u, s') + \gamma \cdot V_{j-1}(s')) \cdot P(s' | u, s) \, ds' \right)$$

- Depois, calcula-se esse valor de ganho iterativamente (analisando mais ações futuras):

$$V_j(s) = \max_u \left(\int (r(s, u, s') + \gamma \cdot V_{j-1}(s')) \cdot P(s' | u, s) \, ds' \right)$$

- E escolhe-se uma ação que maxime esse ganho:

$$\pi_j(s) = \operatorname{argmax}_u \left(\int (r(s, u, s') + \gamma \cdot V_{j-1}(s')) \cdot P(s' | u, s) \, ds' \right)$$

- Para $j \rightarrow \infty$ a política $\pi_j(s)$ tende a ser ótima.

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

- Estima o valor de $V(s)$;

- Estima o valor de $V(s)$;
- Não se tem o modelo de atuação $P(s'|u, s)$ ou de recompensa $r(s, u, s')$;

- Estima o valor de $V(s)$;
- Não se tem o modelo de atuação $P(s'|u, s)$ ou de recompensa $r(s, u, s')$;
- Aprende-se os valores a partir de experiências reais;

- Estima o valor de $V(s)$;
- Não se tem o modelo de atuação $P(s'|u, s)$ ou de recompensa $r(s, u, s')$;
- Aprende-se os valores a partir de experiências reais;
- É necessário ter uma política $\pi_{td}(s)$ pré definida e fixa.

Inicialmente:

$$V_{\pi_{td}}^0(s) = 0, \forall s \in S$$

Inicialmente:

$$V_{\pi_{td}}^0(s) = 0, \forall s \in S$$

Para cada experiência (s, u, s', r) :

Inicialmente:

$$V_{\pi_{td}}^0(s) = 0, \forall s \in S$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *amostra*:

$$amostra = r + \gamma \cdot V_{\pi_{td}}^{t-1}(s')$$

Inicialmente:

$$V_{\pi_{td}}^0(s) = 0, \forall s \in S$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *amostra*:

$$amostra = r + \gamma \cdot V_{\pi_{td}}^{t-1}(s')$$

- Atualiza-se o valor de ganho para o estado s :

$$V_{\pi_{td}}^t(s) = (1 - \alpha) \cdot V_{\pi_{td}}^{t-1}(s) + \alpha \cdot amostra$$

Inicialmente:

$$V_{\pi_{td}}^0(s) = 0, \forall s \in S$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *amostra*:

$$amostra = r + \gamma \cdot V_{\pi_{td}}^{t-1}(s')$$

- Atualiza-se o valor de ganho para o estado s :

$$V_{\pi_{td}}^t(s) = (1 - \alpha) \cdot V_{\pi_{td}}^{t-1}(s) + \alpha \cdot amostra$$

- Para $t \rightarrow \infty$, $V_{\pi_{td}}^t(s)$ tende ao valor obtido no MDP, para uma política π_{td} ótima.

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

- Estima o valor de $Q(s, u)$, sendo Q o valor de ganho para cada par estado-ação:

$$V(s) = \max_u (Q(s, u))$$

- Estima o valor de $Q(s, u)$, sendo Q o valor de ganho para cada par estado-ação:

$$V(s) = \max_u (Q(s, u))$$

- Não se tem o modelo de atuação $P(s'|u, s)$ ou de recompensa $r(s, u, s')$;

- Estima o valor de $Q(s, u)$, sendo Q o valor de ganho para cada par estado-ação:

$$V(s) = \max_u (Q(s, u))$$

- Não se tem o modelo de atuação $P(s'|u, s)$ ou de recompensa $r(s, u, s')$;
- Aprende-se os valores a partir de experiências reais;

- Estima o valor de $Q(s, u)$, sendo Q o valor de ganho para cada par estado-ação:

$$V(s) = \max_u (Q(s, u))$$

- Não se tem o modelo de atuação $P(s'|u, s)$ ou de recompensa $r(s, u, s')$;
- Aprende-se os valores a partir de experiências reais;
- Aprende-se uma política $\pi(s)$ ótima.

Inicialmente:

$$Q^0(s, u) = 0, \forall (s, u) \in (S, U)$$

Inicialmente:

$$Q^0(s, u) = 0, \forall (s, u) \in (S, U)$$

Para cada experiência (s, u, s', r) :

Inicialmente:

$$Q^0(s, u) = 0, \forall (s, u) \in (S, U)$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *amostra*:

$$amostra = r + \gamma \cdot \max_u (Q^{t-1}(s', u'))$$

Inicialmente:

$$Q^0(s, u) = 0, \forall (s, u) \in (S, U)$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *amostra*:

$$amostra = r + \gamma \cdot \max_u (Q^{t-1}(s', u'))$$

- Atualiza-se o valor de ganho para o par estado-ação (s, u) :

$$Q^t(s, u) = (1 - \alpha) \cdot Q^{t-1}(s, u) + \alpha \cdot amostra$$

Inicialmente:

$$Q^0(s, u) = 0, \forall (s, u) \in (S, U)$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *amostra*:

$$amostra = r + \gamma \cdot \max_u (Q^{t-1}(s', u'))$$

- Atualiza-se o valor de ganho para o par estado-ação (s, u) :

$$Q^t(s, u) = (1 - \alpha) \cdot Q^{t-1}(s, u) + \alpha \cdot amostra$$

- Para $t \rightarrow \infty$, uma política $\pi(s)$ é ótima se obtida com a função:

$$\pi^t(s) = \underset{u}{argmax} (Q^t(s, u))$$

Limitações:

Limitações:

- Podem existir muitos estados para se visitar: em um espaço contínuo, por exemplo, seriam infinitos;

Limitações:

- Podem existir muitos estados para se visitar: em um espaço contínuo, por exemplo, seriam infinitos;
- Podem existir muitas ações possíveis para cada estado: para o acionamento analógico de um motor, por exemplo, seriam infinitas;

Limitações:

- Podem existir muitos estados para se visitar: em um espaço contínuo, por exemplo, seriam infinitos;
- Podem existir muitas ações possíveis para cada estado: para o acionamento analógico de um motor, por exemplo, seriam infinitas;
- Se houverem muitos pares ação-estado (S, U) , mesmo que seja possível aprender por um tempo muito grande, é necessário armazenar o valor de $Q(S, U)$ para cada um desses pares;

Limitações:

- Podem existir muitos estados para se visitar: em um espaço contínuo, por exemplo, seriam infinitos;
- Podem existir muitas ações possíveis para cada estado: para o acionamento analógico de um motor, por exemplo, seriam infinitas;
- Se houverem muitos pares ação-estado (S, U) , mesmo que seja possível aprender por um tempo muito grande, é necessário armazenar o valor de $Q(S, U)$ para cada um desses pares;
- O algoritmo não consegue aplicar o que aprendeu em um estado para outros estados com características similares.

Pode-se generalizar os pares estado-ação utilizando-se características deles:

$$Q(S, U) = \omega_1 \cdot f_1(S, U) + \omega_2 \cdot f_2(S, U) + \dots + \omega_n \cdot f_n(S, U)$$

Pode-se generalizar os pares estado-ação utilizando-se características deles:

$$Q(S, U) = \omega_1 \cdot f_1(S, U) + \omega_2 \cdot f_2(S, U) + \dots + \omega_n \cdot f_n(S, U)$$

Para cada experiência (s, u, s', r) :

Pode-se generalizar os pares estado-ação utilizando-se características deles:

$$Q(S, U) = \omega_1 \cdot f_1(S, U) + \omega_2 \cdot f_2(S, U) + \dots + \omega_n \cdot f_n(S, U)$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *erro*:

$$erro = r + \gamma \cdot \max_u (Q^{t-1}(s', u')) - Q^{t-1}(s, u)$$

Pode-se generalizar os pares estado-ação utilizando-se características deles:

$$Q(S, U) = \omega_1 \cdot f_1(S, U) + \omega_2 \cdot f_2(S, U) + \dots + \omega_n \cdot f_n(S, U)$$

Para cada experiência (s, u, s', r) :

- Obtém-se um valor *erro*:

$$erro = r + \gamma \cdot \max_u (Q^{t-1}(s', u')) - Q^{t-1}(s, u)$$

- Atualiza-se o valor de ganho para o par estado-ação (s, u) :

$$\omega_i^t = \omega_i^{t-1} + \alpha \cdot erro \cdot f_i(s, u)$$

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- **Seleção de comportamentos**
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

Analogamente à aprendizagem com os pares estado-ação (S, U) , pode-se utilizar pares estado-comportamento (S, B) :

$$Q(S, B) = \omega_1 \cdot f_1(S, B) + \omega_2 \cdot f_2(S, B) + \dots + \omega_n \cdot f_n(S, B)$$

Analogamente à aprendizagem com os pares estado-ação (S, U) , pode-se utilizar pares estado-comportamento (S, B) :

$$Q(S, B) = \omega_1 \cdot f_1(S, B) + \omega_2 \cdot f_2(S, B) + \dots + \omega_n \cdot f_n(S, B)$$

Agora, para cada experiência (s, b, s', r) :

Analogamente à aprendizagem com os pares estado-ação (S, U) , pode-se utilizar pares estado-comportamento (S, B) :

$$Q(S, B) = \omega_1 \cdot f_1(S, B) + \omega_2 \cdot f_2(S, B) + \dots + \omega_n \cdot f_n(S, B)$$

Agora, para cada experiência (s, b, s', r) :

- Obtém-se um valor *erro*:

$$erro = r + \gamma \cdot \max_b (Q^{t-1}(s', b')) - Q^{t-1}(s, b)$$

Analogamente à aprendizagem com os pares estado-ação (S, U) , pode-se utilizar pares estado-comportamento (S, B) :

$$Q(S, B) = \omega_1 \cdot f_1(S, B) + \omega_2 \cdot f_2(S, B) + \dots + \omega_n \cdot f_n(S, B)$$

Agora, para cada experiência (s, b, s', r) :

- Obtém-se um valor *erro*:

$$erro = r + \gamma \cdot \max_b (Q^{t-1}(s', b')) - Q^{t-1}(s, b)$$

- Atualiza-se todos os pesos ω_i de acordo com o valor de suas características $f_i(s, b)$:

$$\omega_i^t = \omega_i^{t-1} + \alpha \cdot erro \cdot f_i(s, b)$$

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- **Sistema Parcialmente Observável**
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

- *Q Learning* foi criado para um sistema completamente observável;

- *Q Learning* foi criado para um sistema completamente observável;
- Não sabemos o estado atual;

- *Q Learning* foi criado para um sistema completamente observável;
- Não sabemos o estado atual;
- Essa definição então é expandida pra um sistema parcialmente observável;

- *Q Learning* foi criado para um sistema completamente observável;
- Não sabemos o estado atual;
- Essa definição então é expandida pra um sistema parcialmente observável;
- Utiliza-se $a \in A$ no lugar de $s \in S$;

- *Q Learning* foi criado para um sistema completamente observável;
- Não sabemos o estado atual;
- Essa definição então é expandida pra um sistema parcialmente observável;
- Utiliza-se $a \in A$ no lugar de $s \in S$;
- $a \in A$ é uma distribuição de probabilidades de se encontrar nos estados $s \in S$.

Analogamente à aprendizagem anterior, com os pares estado-ação (S, U) , pode-se utilizar pares (distribuição de probabilidade para os estados, comportamento) (A, B) :

$$Q(A, B) = \omega_1 \cdot f_1(A, B) + \omega_2 \cdot f_2(A, B) + \dots + \omega_n \cdot f_n(A, B)$$

Analogamente à aprendizagem anterior, com os pares estado-ação (S, U) , pode-se utilizar pares (distribuição de probabilidade para os estados, comportamento) (A, B) :

$$Q(A, B) = \omega_1 \cdot f_1(A, B) + \omega_2 \cdot f_2(A, B) + \dots + \omega_n \cdot f_n(A, B)$$

Agora, para cada experiência (a, b, a', r) :

Analogamente à aprendizagem anterior, com os pares estado-ação (S, U) , pode-se utilizar pares (distribuição de probabilidade para os estados, comportamento) (A, B) :

$$Q(A, B) = \omega_1 \cdot f_1(A, B) + \omega_2 \cdot f_2(A, B) + \dots + \omega_n \cdot f_n(A, B)$$

Agora, para cada experiência (a, b, a', r) :

- Obtém-se um valor *erro*:

$$erro = r + \gamma \cdot \max_b (Q^{t-1}(a', b')) - Q^{t-1}(a, b)$$

Analogamente à aprendizagem anterior, com os pares estado-ação (S, U) , pode-se utilizar pares (distribuição de probabilidade para os estados, comportamento) (A, B) :

$$Q(A, B) = \omega_1 \cdot f_1(A, B) + \omega_2 \cdot f_2(A, B) + \dots + \omega_n \cdot f_n(A, B)$$

Agora, para cada experiência (a, b, a', r) :

- Obtém-se um valor *erro*:

$$erro = r + \gamma \cdot \max_b (Q^{t-1}(a', b')) - Q^{t-1}(a, b)$$

- Atualiza-se todos os pesos ω_i de acordo com o valor de suas características $f_i(a, b)$:

$$\omega_i^t = \omega_i^{t-1} + \alpha \cdot erro \cdot f_i(a, b)$$

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- **Abordagem Bayesiana Final**
- Plataforma de testes

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

Para incorporar a seleção de comportamento no filtro bayesiano, expande-se as definições de:

Para incorporar a seleção de comportamento no filtro bayesiano, expande-se as definições de:

- Observação:

$$Z^+ = \begin{pmatrix} Z \\ Z_b \end{pmatrix} = \begin{pmatrix} Z \\ B \end{pmatrix}$$

Para incorporar a seleção de comportamento no filtro bayesiano, expande-se as definições de:

- Observação:

$$Z^+ = \begin{pmatrix} Z \\ Z_b \end{pmatrix} = \begin{pmatrix} Z \\ B \end{pmatrix}$$

- Estado:

$$S^+ = \begin{pmatrix} S \\ S_b \end{pmatrix}$$

O sistema final possui duas partes, cada uma dividida em passos:

O sistema final possui duas partes, cada uma dividida em passos:

- Treinamento
 - Predição;
 - Observação;
 - Seleção de comportamento;
 - Seleção de ação motora;
 - Aprendizagem.

O sistema final possui duas partes, cada uma dividida em passos:

- Treinamento
 - Predição;
 - Observação;
 - Seleção de comportamento;
 - Seleção de ação motora;
 - Aprendizagem.
- Pós treinamento
 - Predição;
 - Observação;
 - Seleção de comportamento;
 - Seleção de ação motora

Predição:

$$P(S^t S_b^t \mid z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1} S_b^{t-1}} \left(\begin{array}{l} P(S^t \mid S^{t-1} m^{t-1} \pi_f) \times P(S_b^t \mid \pi_f) \\ \times P(m^{t-1} \mid S^{t-1} S_b^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} S_b^{t-1} \mid z^{0:t-1} b^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

Predição:

$$P(S^t S_b^t \mid z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1} S_b^{t-1}} \left(\begin{array}{l} P(S^t \mid S^{t-1} m^{t-1} \pi_f) \times P(S_b^t \mid \pi_f) \\ \times P(m^{t-1} \mid S^{t-1} S_b^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} S_b^{t-1} \mid z^{0:t-1} b^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

Seleção de Comportamento:

$$Q^t(a^t, B^t) = \omega_1 \cdot f_1(a^t, B^t) + \omega_2 \cdot f_2(a^t, B^t) + \dots + \omega_n \cdot f_n(a^t, B^t)$$

Predição:

$$P(S^t S_b^t \mid z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1} S_b^{t-1}} \left(\begin{array}{l} P(S^t \mid S^{t-1} m^{t-1} \pi_f) \times P(S_b^t \mid \pi_f) \\ \times P(m^{t-1} \mid S^{t-1} S_b^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} S_b^{t-1} \mid z^{0:t-1} b^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

Seleção de Comportamento:

$$Q^t(a^t, B^t) = \omega_1 \cdot f_1(a^t, B^t) + \omega_2 \cdot f_2(a^t, B^t) + \dots + \omega_n \cdot f_n(a^t, B^t)$$

Observação:

$$P(S^t S_b^t \mid z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \propto \left(\begin{array}{l} P(z^t \mid S^t \pi_f) \times P(b^t \mid S_b^t \pi_f) \\ \times P(S^t S_b^t \mid z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \end{array} \right)$$

Predição:

$$P(S^t S_b^t | z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1} S_b^{t-1}} \left(\begin{array}{l} P(S^t | S^{t-1} m^{t-1} \pi_f) \times P(S_b^t | \pi_f) \\ \times P(m^{t-1} | S^{t-1} S_b^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} S_b^{t-1} | z^{0:t-1} b^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

Seleção de Comportamento:

$$Q^t(a^t, B^t) = \omega_1 \cdot f_1(a^t, B^t) + \omega_2 \cdot f_2(a^t, B^t) + \dots + \omega_n \cdot f_n(a^t, B^t)$$

Observação:

$$P(S^t S_b^t | z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \propto \left(\begin{array}{l} P(z^t | S^t \pi_f) \times P(b^t | S_b^t \pi_f) \\ \times P(S^t S_b^t | z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \end{array} \right)$$

Seleção de ação motora:

$$P(M^t | z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \propto \sum_{S^t S_b^t} \left(\begin{array}{l} P(M^t | S^t S_b^t m^{t-1} \pi_f) \\ \times P(S^t S_b^t | z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \end{array} \right)$$

Predição:

$$P(S^t S_b^t | z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \propto \sum_{S^{t-1} S_b^{t-1}} \left(\begin{array}{l} P(S^t | S^{t-1} m^{t-1} \pi_f) \times P(S_b^t | \pi_f) \\ \times P(m^{t-1} | S^{t-1} S_b^{t-1} m^{t-2} \pi_f) \\ \times P(S^{t-1} S_b^{t-1} | z^{0:t-1} b^{0:t-1} m^{0:t-2} \pi_f) \end{array} \right)$$

Seleção de Comportamento:

$$Q^t(a^t, B^t) = \omega_1 \cdot f_1(a^t, B^t) + \omega_2 \cdot f_2(a^t, B^t) + \dots + \omega_n \cdot f_n(a^t, B^t)$$

Observação:

$$P(S^t S_b^t | z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \propto \left(\begin{array}{l} P(z^t | S^t \pi_f) \times P(b^t | S_b^t \pi_f) \\ \times P(S^t S_b^t | z^{0:t-1} b^{0:t-1} m^{0:t-1} \pi_f) \end{array} \right)$$

Seleção de ação motora:

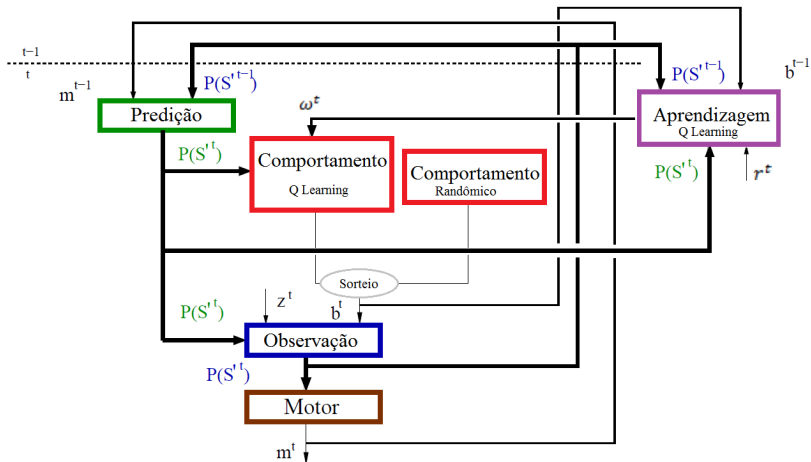
$$P(M^t | z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \propto \sum_{S^t S_b^t} \left(\begin{array}{l} P(M^t | S^t S_b^t m^{t-1} \pi_f) \\ \times P(S^t S_b^t | z^{0:t} b^{0:t} m^{0:t-1} \pi_f) \end{array} \right)$$

Aprendizagem:

$$erro = r + \gamma \cdot \max_{b'} (Q^{t-1}(a^t, b')) - Q^{t-1}(a^{t-1}, b^{t-1})$$

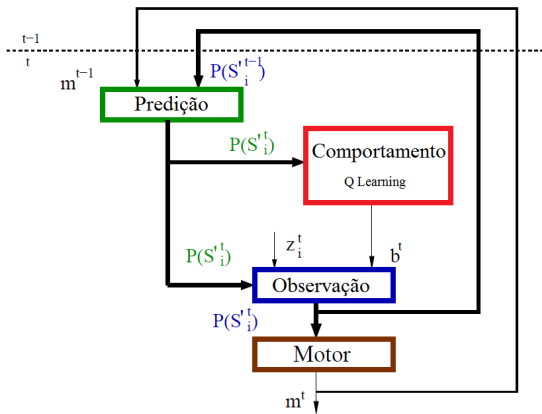
$$\omega_i^t = \omega_i^{t-1} + \alpha \cdot erro \cdot f_i(a^{t-1}, b^{t-1})$$

Treinamento:



Filtro Bayesiano utilizando *Q Learning* para Seleção de Comportamento. Durante treinamento.

Pós treinamento:



Filtro Bayesiano utilizando *Q Learning* para Seleção de Comportamento.
Após treinamento completo.

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

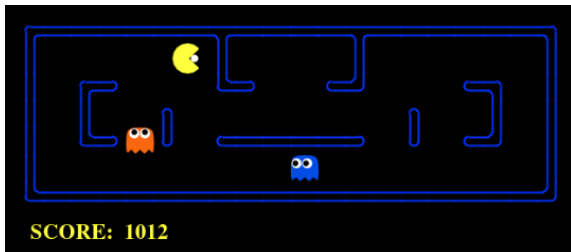
3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- **Plataforma de testes**

4 Resultados

- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

- Plataforma de Pacman¹ em Python;
- Código desenvolvido em C++;
- Integração feita utilizando ROS.



Plataforma do jogo Pacman.

¹Essa plataforma foi desenvolvida em Berkeley para aulas de IA e pode ser encontrada em http://ai.berkeley.edu/project_overview.html

Quatro mensagens trocadas:

Quatro mensagens trocadas:

- Posição do agente (Pacman) — $(x, y) \in \mathbb{R}^2$;

Quatro mensagens trocadas:

- Posição do agente (Pacman) — $(x, y) \in \mathbb{R}^2$;
- Distância para os fantasmas — $(d_x, d_y) = (\Delta x, \Delta y) \in \mathbb{R}^2$;

Quatro mensagens trocadas:

- Posição do agente (Pacman) — $(x, y) \in \mathbb{R}^2$;
- Distância para os fantasmas — $(d_x, d_y) = (\Delta x, \Delta y) \in \mathbb{R}^2$;
- Ação a ser executada —
 $m \in \{Norte, Sul, Leste, Oeste, Esperar\}$;

Quatro mensagens trocadas:

- Posição do agente (Pacman) — $(x, y) \in \mathbb{R}^2$;
- Distância para os fantasmas — $(d_x, d_y) = (\Delta x, \Delta y) \in \mathbb{R}^2$;
- Ação a ser executada —
 $m \in \{Norte, Sul, Leste, Oeste, Esperar\}$;
- Recompensa recebida do ambiente — $r \in \mathbb{R}$.

Erros inseridos:

Erros inseridos:

- Posição do agente (Pacman) — Erro gaussiano;

Erros inseridos:

- Posição do agente (Pacman) — Erro gaussiano;
- Distância para os fantasmas — Erro gaussiano;

Erros inseridos:

- Posição do agente (Pacman) — Erro gaussiano;
- Distância para os fantasmas — Erro gaussiano;
- Ação a ser executada — Chance $\nu_{atuacao}$ de ação desejada ser executada (se não, é executada ação aleatória);

Erros inseridos:

- Posição do agente (Pacman) — Erro gaussiano;
- Distância para os fantasmas — Erro gaussiano;
- Ação a ser executada — Chance $\nu_{atuacao}$ de ação desejada ser executada (se não, é executada ação aleatória);
- Recompensa recebida do ambiente — Nenhum erro inserido.

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

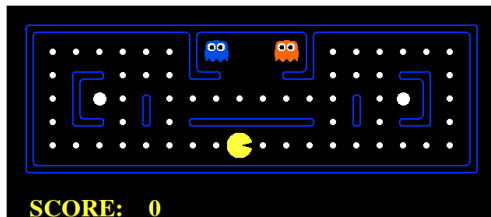
- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

Comportamentos:

- Ficar Parado;
- Comer;
- Fugir.

Características ($f_i(a, B)$):

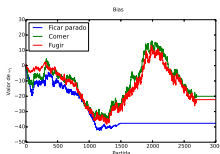
- Bias;
- Distância para Comida;
- Probabilidade de Fantasma.



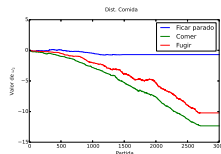
Mapa pequeno do jogo Pacman.

$$Q^t(a^t, B^t) = \omega_1 \cdot f_1(a^t, B^t) + \omega_2 \cdot f_2(a^t, B^t) + \omega_3 \cdot f_3(a^t, B^t)$$

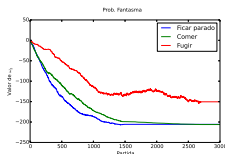
$$Q^t(a^t, B^t) = \omega_1 \cdot f_1(a^t, B^t) + \omega_2 \cdot f_2(a^t, B^t) + \omega_3 \cdot f_3(a^t, B^t)$$



(a) Bias (ω_1)

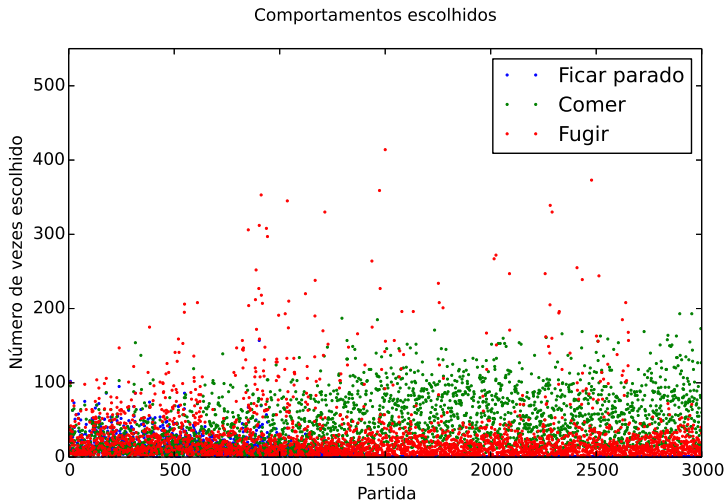


(b) Distância para Comida (ω_2)

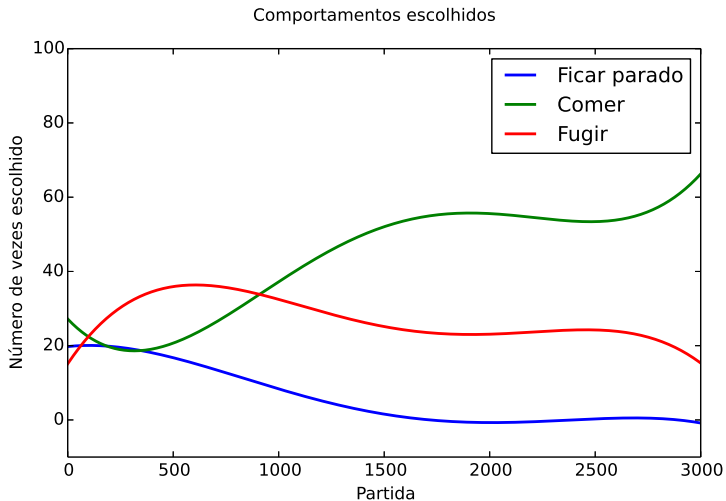


(c) Probabilidade de Fantasma

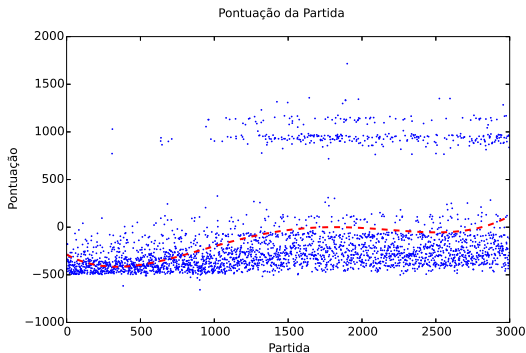
(ω_3)



Escolha dos comportamentos por partida.



Escolha dos comportamentos por partida.



Pontuação por partida.

Pontuação analisada após a conclusão do treinamento:

$$média(pontuação) = 50.04$$

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

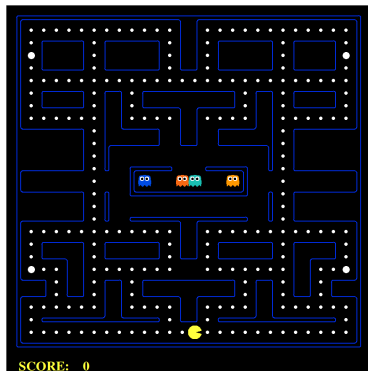
- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

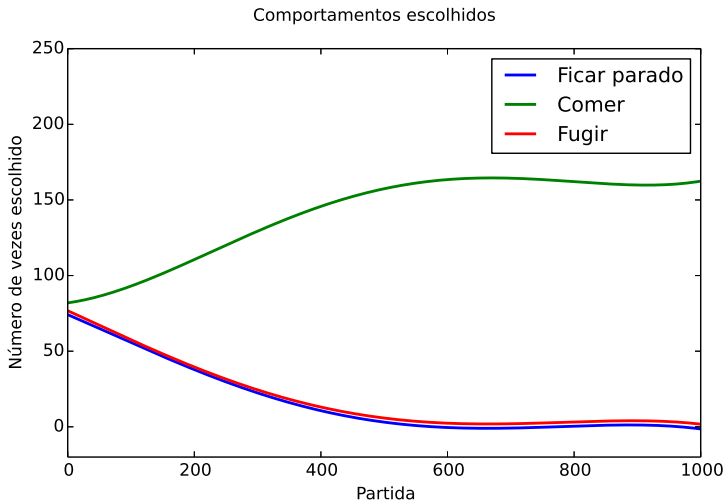
- 3 Comportamentos no mapa pequeno (Teste 1)
- **3 Comportamentos no mapa clássico (Teste 2)**
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

Comportamentos:

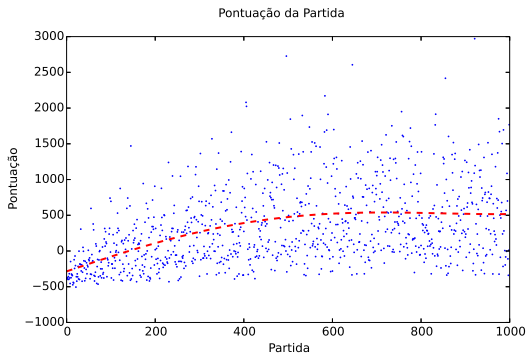
- Ficar Parado;
- Comer;
- Fugir.



Mapa clássico do jogo Pacman.



Escolha dos comportamentos por partida.



Pontuação por partida.

Pontuação analisada após a conclusão do treinamento:

$$média(pontuação) = 536.76$$

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

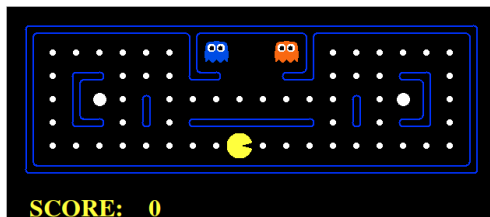
- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- **5 Comportamentos no mapa pequeno (Teste 3)**
- 5 Comportamentos no mapa clássico (Teste 4)

Comportamentos:

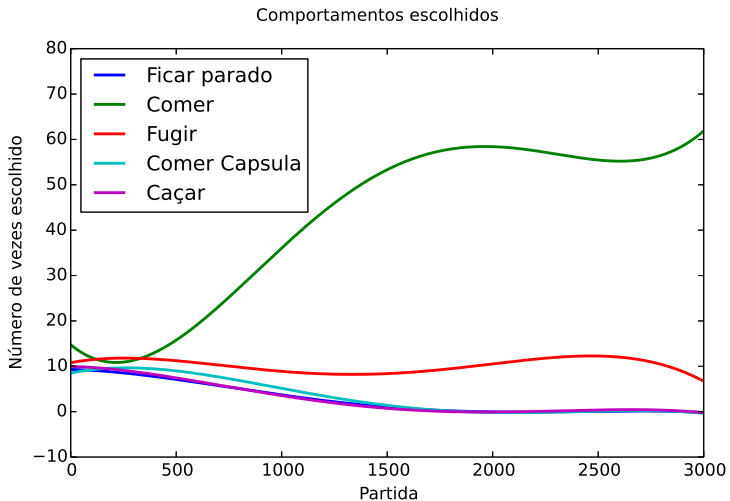
- Ficar Parado;
- Comer;
- Fugir;
- Comer Cápsula;
- Caçar.

Características ($f_i(a, B)$):

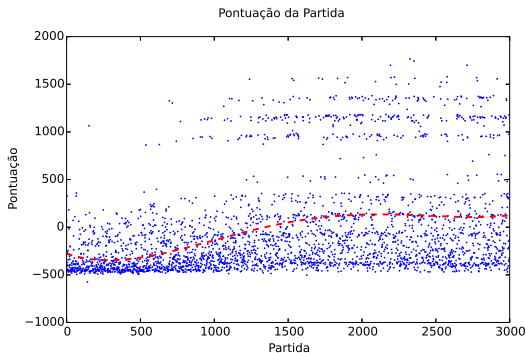
- Bias;
- Proximidade Comida;
- Proximidade Cápsula;
- Probabilidade de Fantasma;
- Probabilidade de Fantasma Branco.



Mapa pequeno do jogo Pacman.



Escolha dos comportamentos por partida.



Pontuação por partida.

Pontuação analisada após a conclusão do treinamento:

$$média(pontuação) = 143.93$$

1 Motivação

- Contextualização
- Trabalhos anteriores

2 Fundamentação Teórica

- Filtro Bayesiano
- MDP (Processo de Decisão de Markov)
- TD (Diferença Temporal)
- Q Learning

3 Solução Implementada

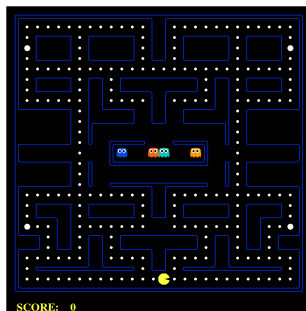
- Seleção de comportamentos
- Sistema Parcialmente Observável
- Abordagem Bayesiana Final
- Plataforma de testes

4 Resultados

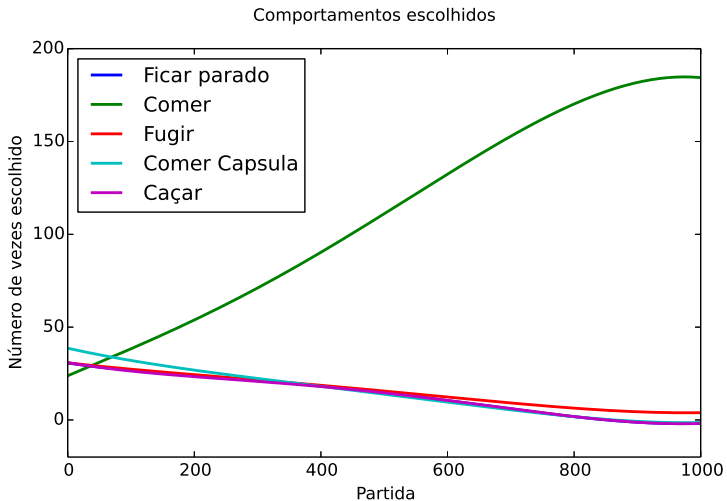
- 3 Comportamentos no mapa pequeno (Teste 1)
- 3 Comportamentos no mapa clássico (Teste 2)
- 5 Comportamentos no mapa pequeno (Teste 3)
- 5 Comportamentos no mapa clássico (Teste 4)

Comportamentos:

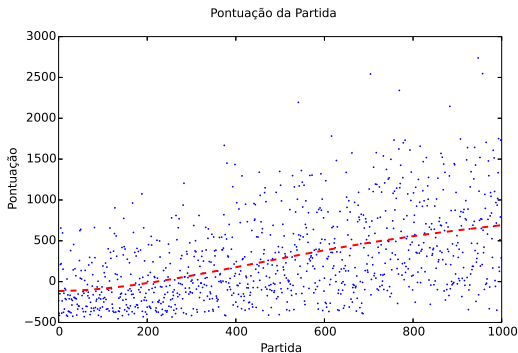
- Ficar Parado;
- Comer;
- Fugir;
- Comer Cápsula;
- Caçar.



Mapa clássico do jogo Pacman.



Escolha dos comportamentos por partida.






Pontuação por partida.

Pontuação analisada após a conclusão do treinamento:

$$média(pontuação) = 619.20$$

Conclusões

- O algoritmo consegue escolher comportamentos de forma lógica;
- Foi possível executar tarefas complexas utilizando modelos de seleção de ação simples;
- O vetor de características dos pares (A, B) deve ser escolhido com cuidado;
- O algoritmo possui certa dificuldade de superar máximos locais.

-  KOIKE, C. M. C. e C. Bayesian approach to action selection and attention focusing. In: BESSIÈRE, P.; LAUGIER, C.; SIEGWART, R. (Ed.). *Probabilistic reasoning and decision making in sensory motor systems*. Berlin: Springer, 2008, (Springer tracts in advanced robotics).
-  LIDORIS, G. *State Estimation, Planning, and Behavior Selection Under Uncertainty for Autonomous Robotic Exploration in Dynamic Environments*. Kassel University Press, 2011. ISBN 9783862190638. Disponível em:
<<http://books.google.com.br/books?id=3PjJwKvQcnYC>>.
-  KOIKE, C. M. C. e C. *Bayesian Approach to Action Selection and Attention Focusing*. Tese (Doutorado) — Institut National Polytechnique De Grenoble, 2005.

Obrigado