# Aprendizado de affordances aplicado à navegação de robôs autônomos

Rosana Veroneze

***Abstract*—A aprendizagem e a percepção da transitabilidade é uma competência fundamental para os robôs autônomos móveis. Por sua vez, isso requer que o robô seja capaz de predizer affordances com base nos dados de seus sensores. Destarte, este trabalho objetiva fazer um estudo de algumas pesquisas que trabalharam com o aprendizado de affordances aplicado à navegação de robôs autônomos. Este estudo foca nos modelos de aprendizagem utilizados e não nos resultados obtidos por essas pesquisas.**

***Palavras-Chave*—affordance, robôs autônomos, transitabilidade, percepção.**

## I. Introdução

O conceito *affordance* foi cunhado por Gibson (1966), o qual foi um dos maiores psicólogos do século XX e trabalhou com o desenvolvimento de uma nova teoria de percepção (Sahin *et al.*, 2007). Em seu livro *The ecological approach to visual perception*, Gibson (1979) diz que os affordances do ambiente são aquilo que o ambiente oferece ao animal. Uma questão importante é que os affordances podem variar de espécie para espécie (e até mesmo de animal para animal), pois dependem da capacidade que a espécie tem de perceber e usar tais affordances. Embora o termo affordance tenha surgido dentro da psicologia, esse conceito influenciou pesquisas em diversas áreas (Sahin *et al.*, 2007), como: psicologia ecológica; ciência cognitiva; neurofisiologia; neuropsicologia; interface humano – computador; e robôs autônomos. Dentre essas áreas citadas, esse trabalho foca na aplicação do conceito affordance à robótica, mais especificamente, no aprendizado de affordances aplicado à navegação de robôs autônomos.

Para que um robô possa realizar tarefas, é fundamental que ele seja capaz de interagir com o ambiente e com os objetos contidos nesse ambiente, sendo capaz de predizer o efeito de suas ações sobre os objetos e o ambiente (Sun *et al.*, 2010). Por sua vez, isso requer que o robô seja capaz de predizer affordances com base nos dados de seus sensores. Atualmente, na maioria dos sistemas robóticos práticos, a predição de affordances é feita implicitamente por meio de regras que são inseridas no software do robô (Sun *et al.*, 2010). Mas, outra abordagem possível é tornar os robôs capazes de aprender a realizar tarefas em um ambiente desconhecido sem a orientação explícita de humanos, tornando o robô capaz de aprender affordances.

Em se tratando da navegação de robôs, um conceito importante é o da *transitabilidade*, o qual se refere ao affordance de ser capaz de transitar sobre determinado local. Segundo Warren (1984), para determinar se um caminho é transitável, as propriedades comportamentais relevantes do ambiente devem ser analisadas em relação às propriedades relevantes e o sistema de ação do animal. A aprendizagem e a percepção da transitabilidade é uma competência fundamental para os organismos vivos e para os robôs autônomos, porque a maioria de suas ações depende de sua mobilidade (Ugur & Sahin, 2010). Apesar da importância desse conceito, muitas pesquisas se dedicam simplesmente a evitar qualquer tipo de obstáculo e a guiar os robôs a espaços abertos (Ugur & Sahin, 2010). O problema dessa abordagem é que ela não difere, por exemplo, uma rocha de uma bexiga. Além disso, algo pode ser intransponível para um robô e não ser para outro. Portanto, o aprendizado automático de affordances de transitabilidade (por meio de experimentação) possibilita a criação de estratégias de navegação efetivas e ajustadas para as propriedades locais do terreno e para as propriedades do robô.

Um dos primeiros trabalhos a tratar a estimação de transitabilidade como um problema de aprendizado on-line de affordances foi desenvolvido no *Instituto de Tecnologia da Georgia* por Kim *et al.* (2006). Para isso, os autores utilizaram uma abordagem chamada percepção direta (PD), a qual é utilizada na maioria dos trabalhos de aprendizagem de affordances (Sun *et al.*, 2010). Ainda em 2006, pesquisadores desse instituto publicaram mais um artigo utilizando a PD para o aprendizado de affordances (Sun *et al.*, 2006). Os resultados desses dois primeiros trabalhos mostraram os potenciais benefícios de uma proposta baseada em aprendizado para a predição de affordances. Entretanto, tentativas de escalar tal proposta a um conjunto mais complexo de affordances revelaram que o modelo PD tem limitações de escalabilidade (Sun *et al.*, 2010). Buscando solucionar essa limitação, um novo modelo, chamado Categoria-Affordance (CA), foi proposto recentemente (Sun *et al.*, 2010).

Visando entender como o conceito de affordances pode ser aplicado à navegação de robôs autônomos, este artigo faz um estudo dos três trabalhos desenvolvidos pelo grupo da Georgia e também do recente trabalho de Ugur & Sahin (2010). Este último também utiliza o modelo PD, mas de forma diferenciada do proposto pelo grupo da Georgia.

Este trabalho está organizado da seguinte maneira. A

Seção 2 apresenta os três trabalhos baseados no modelo PD. A Seção 3 apresenta o modelo CA. A Seção 4 faz uma comparação entre os modelos PD e CA. E, finalmente, a Seção 5 apresenta algumas considerações finais.

## II. MODELO PERCEPÇÃO DIRETA

### A. *O Modelo PD desenvolvido no Instituto de Tecnologia da Georgia*

O trabalho de KIM *et al.* (2006) adotada uma abordagem de treinamento on-line, no qual o robô aprende os affordances de transitabilidade por meio de interações com o ambiente. O processo de aprendizagem produz um classificador capaz de fazer predições de transitabilidade para novas regiões do terreno.

O robô utilizado nos experimentos de KIM *et al.* (2006) é exibido na Fig. 1. Vale ressaltar que a utilização de um robô idêntico a esse não é necessária. Entretanto, o robô escolhido deve possuir: sistema de GPS, sensores de visão, e sensores de navegação (como o para-choque).

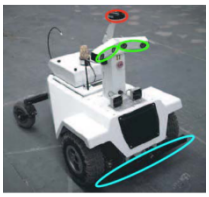

Fig. 1: Robô utilizado em KIM *et al.* (2006). Ele é equipado com um sistema de GPS (vermelho), 2 cabeças estéreo (verde) e para-choque (ciano).
Fonte: KIM *et al.* (2006).

A abordagem adotada para a coleta autônoma de dados do ambiente é ilustrada na Fig. 2. O robô faz um retrato do terreno em sua frente no instante $t - N$. Como pode ser visto, o mapa do terreno é baseado em grade. Cada pedaço de imagem (*image patch*) é a observação de uma única célula da grade. O robô armazena os *image patches* resultantes de sua visualização do terreno em um conjunto de dados. Inicialmente, esses dados não são rotulados porque o robô ainda não interagiu com o terreno que ele visualizou, portanto a transitabilidade não é conhecida. No instante $t$, o robô tentará navegar sob o terreno que ele previamente havia visualizado, descobrindo, então, as propriedades de transitabilidade do terreno. As células sob o robô que são transitáveis resultam em exemplos de treinamento positivos, e vice-versa. Os dados de treinamento são vetores de características visuais e geométricas quantizadas dos *image patches*. Cada *image patch* é codificado por um vetor de características $x \in \Re^{13}$ (e como já foi dito, atribuído a uma célula da grade do mapa local).

Para estabelecer a correspondência entre os *image patches* coletados e as células no mapa do terreno, foi utilizado um mapa local. Esse mapa local considera apenas as observações que estão em certa janela de tempo de tamanho $N$ (veja Fig. 3). O sistema constrói um mapa local $M_t$ com as características da interação $t$. Cada novo mapa local, $M_t$, alimenta uma fila de tamanho $N$. Apenas os dados que são rotulados nesse período são utilizados para o treinamento (os dados não rotulados são, simplesmente, descartados). É importante salientar que os vetores de características armazenados nos mapas locais são rotulados (como transitável ou não) com base no sistema de GPS e nos sensores de navegação.

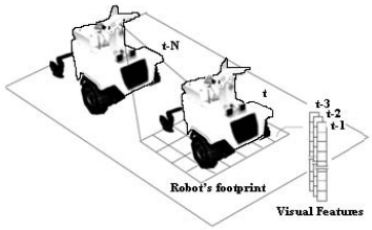


Fig. 2: Ilustração da coleta autônoma de dados. O robô faz o retrato de células em um mapa do terreno no instante $t - N$. Essas células estarão sob o robô no instante $t$.
Fonte: KIM *et al.* (2006).

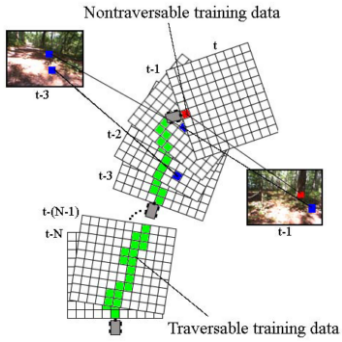


Fig. 3: Rotulação autônoma de células do terreno e características visuais correspondentes às células. As células verdes foram rotuladas como transitáveis e a célula vermelha encontrada no instante $t$ foi rotulada como não-transitável.
Fonte: KIM *et al.* (2006).

Algoritmo 1: Algoritmo de aprendizado on-line

1. Para cada $x$ de um *image patch*, se o número de cluster é zero, então faça $x$ o centro de um novo cluster.
2. Se o número de clusters atingir um dado limite, então faça o raio fixo $r = 2r$.
3. Se $D(x, c_m) \leq r$ e o rótulo de $x$ e $c_m$ for o mesmo, então atualize o centro do cluster $c_m$.
4. Se o rótulo de $x$ e $c_m$ não for o mesmo, então reduza uma amostra do cluster $c_m$. Se o número de amostram em $c_m$ atingir zero, então remova o cluster $c_m$.
5. Se $D(x, c_m) > r$, então crie um novo cluster com centro $x$.

onde $c_m$ é o centro mais próximo a $x$.

Uma vez que as características estejam rotuladas, elas são utilizadas como entrada para o módulo de aprendizagem on-line. O módulo de aprendizagem agrupa os vetores $x$ rotulados com base em um raio de hiperesfera fixo ($r$). São mantidos conjuntos de clusters separados para exemplos positivos e negativos. O Algoritmo 1 exibe o método de clusterização. A medida de distância adotada foi:

$$D(x, c_i) = w\sum_{j=1}^{8} \frac{(c_i^j - x^j)^2}{2(c_i^j + x^j)} + (1-w)\sum_{j=9}^{13} \frac{(c_i^j - x^j)^2}{2(c_i^j + x^j)}. \quad (1)$$

O vetor de características está dividido em dois grupos – características de aparência (de 1 a 8) e geométricas (de 9 a 13) – os quais são ponderados por $w$ e $(1 - w)$, respectivamente, com $w \in [0,1]$.

Com base nos modelos aprendidos, um módulo de classificação pode classificar os novos *image patches*. O classificador on-line é dado por:

$$H(g) = I\left[\frac{\sum_{i=1}^{n} h(x_i)}{n} \geq \theta\right], \quad (2)$$

onde $g$ é uma célula da grade do mapa local, a qual é associada a $n$ vetores $x$; e $h(x)$ é dado por:

$$h(x) = I\left[\frac{D(x, c_{NT})}{D(x, c_T)} \geq \theta'\right], \quad (3)$$

onde $c_{NT}$ é o centro do cluster não-transitável mais próximo a $x$; e $c_T$ é o centro do cluster transitável mais próximo a $x$.

Uma extensão desse trabalho foi realizado por SUN *et al.* (2006). Nessa extensão, o objetivo é que o robô aprenda a distinguir entre terrenos preferíveis e não preferíveis, além de transitáveis e não transitáveis. Para isso, um "professor" guia o robô no ambiente e as sequências de imagens e estados do robô são armazenados em um arquivo de log. Então, o robô precisa aprender a associar as cores das imagens ao custo de percorrer o caminho. Isso é feito considerando-se que as cores por onde o "professor" transitou são provavelmente melhores do que aquelas por onde ele não transitou.

### B. *O trabalho de UGUR & SALIM (2010)*

No trabalho de UGUR & SALIM (2010), os affordances são representados por uma tripla aninhada: **(*efeito*, (*entidade*, *comportamento*))**, indicando que quando um agente aplica um *comportamento* em certa *entidade*, o *efeito* é gerado. Os comportamentos que o robô pode realizar são pré-determinados, sendo eles: andar 70 centímetros em um ângulo de 0º, ±20º, ±40º ou ±60 º.

O treinamento do robô começa com uma fase de exploração realizada em ambiente virtual, a qual é exibida no Algoritmo 2.

Após a fase de exploração, acontece a de aprendizado (Algoritmo 3). Nessa fase, para um dado comportamento, o robô descobre as características relevantes do ambiente para transitabilidade (ou não- transitabilidade) e aprende a mapear tais características a seus affordances. Para encontrar as características relevantes, foi utilizado o algoritmo ReliefF (KIRA & RENDELL, 1992). Nesse algoritmo, a relevância de uma característica é aumentada se a característica assume valores similares para situações que possuem os mesmos resultados de execução e vice-versa. As características com relevância acima de um limiar são consideradas relevantes. O algoritmo SVM (VAPNIK, 1998) foi utilizado para classificar a transitabilidade com base nas características relevantes.

Na fase de execução, os classificadores SVM são utilizados para predizer a existência de affordances. Nessa fase, o modulo de predição de affordances recebe um comportamento $b^i$ como entrada e prediz o affordance desse comportamento com base na percepção do ambiente, e $H^i$ e $X^i$ encontrados na fase de aprendizado.

Algoritmo 2: Fase de exploração

**Para** cada tentativa *k* **faça**
  Coloque o robô em um ambiente aleatoriamente construído
  O robô deve sensoriar o ambiente, criando o vetor de características $x_k$.
  **Para** cada comportamento $b^i$ **faça**
    Execute $b^i$
    Armazene $< b^i, x_k, r_{k,i}>$ no repositório
    Volte a posição do robô e dos objetos
  **Fim Para**
**Fim Para**

onde $r_{k,i}$ é o resultado do comportamento $b^i$ aplicado ao vetor de características $x_k$, o qual armazena dados sobre toda a grade (e não apenas de uma célula da grade como nos dois trabalhos anteriormente apresentados).

Algoritmo 3: Fase de aprendizado

**Para** cada comportamento $b^i$ **faça**
  Busque amostras $<x_k, r_{k,i}>$ do repositório para o comportamento $b^i$
  Encontre um conjunto de características relevantes $X^i$
  Treine o modelo SVM, $H^i$, com $X^i$
  Armazene $H^i$ e $X^i$ para a percepção de affordances no modo de execução
**Fim Para**

## III. MODELO CATEGORIA-AFFORDANCE

### A. *Descrição do Modelo Categoria-Affordance*

Seja $c$ uma variável discreta representando uma de $C$ categorias possíveis de um objeto. Em geral, é suposto que $C$ é conhecido a priori, e pode ser determinado pela distribuição de $x$ (por exemplo, por meio de clusterização). A distribuição $p(x|c)$ é um modelo generativo probabilístico das características de um objeto (por exemplo, uma mistura de gaussianas). Seja $a$ um vetor de tamanho $K$, que codifica as propriedades de affordance de um dado objeto. Então, $a^k$ é uma variável binária aleatória associada ao $k$-ésimo affordance. Em outras palavras, $a$ é um vetor binário indicando quais affordances um dado objeto possui. Note que esse modelo foi desenvolvido para um agente robô específico,

com capacidades físicas fixas. Assim, os affordances podem ser vistos como uma propriedade do objeto.

É sabido que existem dependências probabilísticas entre os atributos de affordance. Entretanto, uma vez que existem $2^K$ vetores binários possíveis, não é tratável representar diretamente a distribuição completa de probabilidade conjunta dos affordances. Então, considerou-se que $p(a^k \mid a^{\bar{k}}, c, x) = p(a^k \mid c, x)$, onde $a^{\bar{k}}$ é o conjunto dos $K-1$ affordances restantes. Intuitivamente, considera-se que qualquer acoplamento entre affordances é capturado pela dependência entre a categoria e as características do objeto.

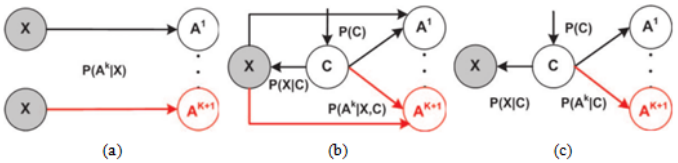


Fig. 4: (a) Modelo gráfico da abordagem PD. (b) Modelo gráfico da abordagem CA-full. (a) Modelo gráfico da abordagem CA-chain.
Fonte: SUN *et al.* (2010).

Dadas essas suposições, a questão remanescente é como fatorar a distribuição conjunta $p(x, c, a)$. Na abordagem PD, a categoria $c$ é marginalizada para fora do modelo e a distribuição resultante é:

$$p(x,a) = p(a \mid x)\,p(x) = \prod_{k=1}^{K} p(a^k \mid x)\,p(x), \tag{4}$$

onde a última igualdade decorre da independência condicional de $a^k$. O modelo PD é ilustrado como uma rede bayesiana na Fig. 4a. Na abordagem CA, considera-se que a categorização de objetos com base em suas características é suficiente para fornecer pistas úteis para a previsão de affordances. Assim, a distrubuição conjunta é fatorada como:

$$p(x,a,c) = p(c)\,p(x \mid c)\prod_{k=1}^{K} p(a^k \mid x,c). \tag{5}$$

Essa é a forma mais geral do modelo CA e é denominada *CA-full*. A Fig. 4b descreve o modelo CA-full por meio de uma rede bayesiana.

Após o treinamento desse modelo (métodos de treinamento serão discutidos na Subseção 3.B.), é possível calcular a distribuição condicional de um affordance $a^k$ dado as características $x$ de um objeto, marginalizando o rótulo da categoria desconhecida:

$$p(a^k \mid x) = \sum_{c} p(a^k \mid c, x)\,p(c \mid x), \tag{6}$$

onde $p(c|x)$ pode ser obtida, por exemplo, pela regra de Bayes.

Uma alternativa ao modelo CA-full é o modelo *CA-chain*, o qual considera que o affordance é condicionalmente independente das características do objeto dado a categoria. Nesse caso $p(a^k \mid x, c) = p(a^k \mid c)$, levando a uma fatoração alternativa:

$$p(x,a,c) = p(c)\,p(x \mid c)\prod_{k=1}^{K} p(a^k \mid c). \tag{7}$$

Nesse modelo, as características determinam a categoria, a qual determina o affordance. A Fig. 4c apresenta o modelo *CA-chain*.

### B. *Treinamento do Modelo Categoria-Affordance*

Os métodos de treinamento propostos por SUN *et al.* (2010) se basearam no modelo CA-full, porque o modelo CA-chain é um caso especial simplificado de CA-full.

Foram propostas dois métodos de treinamento: (*i*) um generativo que maximiza a função de log-verossimilhança conjunta (Eq. 8); e (*ii*) um discriminativo que maximiza a função de log-verossimilhança condicional (Eq. 9):

$$LL(\theta; D) = \sum_{n} \log\, p(a_n, x_n \mid \theta), \tag{8}$$

$$CLL(\theta; D) = \sum_{n} \log\, p(a_n \mid x_n, \theta), \tag{9}$$

onde $n$ se refere a e*n*ésima amostra de treinamento; e $\theta$ é o vetor de parâmetros do modelo escolhido (por exemplo, em um modelo de mistura seriam os pesos, centros e matrizes de covariância das gaussianas; em uma regressão logística seria o vetor de pesos).

Em geral, maximizar a função *LL* resulta em um modelo generativamente treinado porque o objetivo é descrever precisamente a distribuição geral conjunta dos dados. Em contrapartida, maximizar *CLL* é um método de treinamento discriminativo porque se concentra unicamente no componente do modelo que determina a precisão da predição do rótulo. Ambas as funções, *LL* e *CLL*, podem ser maximizadas através do algoritmo EM (DEMPSTER *et al.*, 1977), tratando o rótulo da categoria $c$ como um dado faltante. Maiores detalhes de como implementar o algoritmo EM serão dados apenas com relação ao treinamento generativo.

O passo-E do algoritmo EM encontra o valor esperado da função de verossimilhança:

$$\begin{aligned} Q(\theta, \theta^{(t)}) &= E^{\theta^{(t)}} \sum_{n} \log p(x_n, \tilde{c}_n, a_n \mid \theta) \\ &= \sum_{n}\sum_{c} q_{n,c}^{(t)} \cdot \log p(x_n, a_n, c \mid \theta), \end{aligned} \tag{10}$$

onde $\tilde{c}$ é a categoria não observada e $E^{\theta^{(t)}}$ denota que a esperança é tomada sob a distribuição posterior de

$\tilde{c}$ calculada com os parâmetros do modelo da iteração *t*. A distribuição posterior de *c* para a *enésima* amostra de treinamento, $q_{n,c}^{(t)}$, é dada por:

$$q_{n,c}^{(t)} = p\left(c \mid x_n, a_n, \theta^{(t)}\right) \propto p\left(c \mid x_n, \theta^{(t)}\right) p\left(a_n \mid c, x_n, \theta^{(t)}\right). \quad (11)$$

A Eq. 10 pode ser reescrita da seguinte maneira:

$$\begin{aligned} Q(\theta, \theta^{(t)}) = & \sum_n \sum_c q_{n,c}^{(t)} \cdot \log p\left(c|\phi\right) \\ & + \sum_n \sum_c q_{n,c}^{(t)} \cdot \log p\left(x_n|c, \phi\right) \\ & + \sum_n \sum_c q_{n,c}^{(t)} \cdot \log p\left(a_n|c, x_n, \psi\right), \end{aligned} \quad (12)$$

onde os parâmetros do modelo são divididos em dois subconjuntos independentes de tal forma que $\theta \doteq (\phi, \psi)$, com $\phi$ sendo o modelo CA e $\psi$ os classificadores de affordances. Portando, a cada iteração, $\phi$ pode ser obtido maximizando os dois primeiros termos da Eq. 12, e $\psi$ pode ser obtido maximizando o terceiro termo, de forma dissociada.

Finalmente, trocando a ordem dos somatórios, chega-se ao passo-M com três maximizações independentes, com os parâmetros $\phi$ e $\psi$ mais uma vez divididos em $\phi \doteq (\phi_\pi, \{\phi_c\})$ e $\psi \doteq \{\psi_c\}$ (sendo que $\phi_\pi$ é o antecedente da categoria; e $\phi_c$ e $\psi_c$ são o modelo de aparência e os classificadores de affordance para uma determinada categoria *c*, respectivamente):

$$\phi_\pi^{(t+1)} = \arg\max_{\phi_\pi} \sum_c \left( \sum_n q_{n,c}^{(t)} \right) \log\ p\left(c|\phi_\pi\right), \quad (13)$$

$$\forall c\ \ \phi_c^{(t+1)} = \arg\max_{\phi_c} \sum_n q_{n,c}^{(t)} \log\ p\left(x_n|c, \phi_c\right), \quad (14)$$

$$\forall c\ \ \psi_c^{(t+1)} = \arg\max_{\psi_c} \sum_n q_{n,c}^{(t)} \log\ p\left(a_n|c, x_n, \psi_c\right), \quad (15)$$

A Eq. 13 obtém os antecedentes das categorias do objeto combinando $\phi_\pi$ com a distribuição a posteriori de cada categoria. A Eq. 14 aprende o modelo de aparência de cada categoria de objeto e cada categoria é aprendida de forma independente. A Eq. 15 aprende os classificadores para affordances múltiplos dentro de cada categoria de objeto. Embora classificadores para diferentes categorias sejam independentes, dentro de cada categoria, classificadores para affordances diferentes são conectados por $p(a_n \mid c, x_n, \psi_c)$. Isso pode ser uma tarefa árdua, mas pode ser simplificada pela seguinte aproximação:

para todo *c* e todo *k*,

$$\psi_{c,k}^{(t+1)} = \arg\max_{\psi_{c,k}} \sum_{n \in S_k} q_{n,c}^{(t)} \log\ p\left(a_n^k \middle| c, x_n, \psi_{c,k}\right), \quad (16)$$

onde $S_k$ é o conjunto de treinamento para o *k*-ésimo affordance. Essa é a união de todos os dados de treinamento tal que o *k*-ésimo affordance seja observado no vetor $a_n$. A Eq. 16 pode ser facilmente maximizada por métodos convencionais. Por exemplo, em SUN *et al.* (2010), classificadores de regressão logística foram obtidos iterativamente pelo método dos mínimos quadrados reponderado (MMQR) (JORDAN & JACOBS, 1994). O Algoritmo 4 esboça o algoritmo utilizado para o treinamento.

Algoritmo 4: Treinamento Generativo via EM Generalizado

Inicialize $\phi^{(0)}$ e $\psi^{(0)}$.
**Para** cada iteração *t* =1, 2, ..., *T* **faça**
Usando os parâmetros do modelo da iteração *t* – 1, estime $q_{n,c}t-1$.
{Aprender o modelo Categoria-Aparência}
**Para** cada categoria *c* =1, 2, ..., *C* **faça**
Com os dados de treinamento para a categoria *c*, use o algoritmo EM para obter o modelo de mistura gaussiano.
**Fim Para**
{Aprender o classificador de affordance para uma categoria específica}
**Para** cada categoria *c* =1, 2, ..., *C* **faça**
**Para** cada affordance *k* =1, 2, ..., *K* **faça**
Com os dados de treinamento para o affordance *k*, use MMQR para aprender os classificadores de regressão logística.
**Fim Para**
**Fim Para**
**Fim Para**
{Pós-seleção do modelo de processamento}
**Para** cada iteração *t* =1, 2, ..., *T* **faça**
Aplique o modelo aprendido na iteração *t* para testar o resultado da classificação de affordance, selecione o modelo com maior precisão.
**Fim Para**

## IV. COMPARAÇÃO ENTRE OS MODELOS

Uma distinção importante entre os modelos PD, CA-full e CA-chain se relaciona à aprendizagem incremental de novos affordances. Na Fig. 4, cada modelo contém uma parte em vermelho que denota um affordance adicional a ser aprendido. O método PD é o de maior custo computacional porque ao adicionar um novo affordance é necessário adicionar um novo componente ao modelo que precisará ser treinado a partir do zero. O menos custoso é o modelo CA-chain porque adicionar um novo affordance requer, simplesmente, a adição de uma linha (correspondente a $p(a^{k+1}|c)$) na tabela de probabilidade $p(a|c)$. Note que isso não

requer nenhuma modificação em $p(x|c)$ e não requer acesso às características $x$. Além de $p(a^{k+1}|c)$, o modelo CA-full requer o treinamento de $C$ novos classificadores de affordances de categoria específica $p(a^{k+1}|c, x)$. Assim como no modelo CA-chain, adicionar um novo affordance ao modelo CA-full não exige alterar a categorização dos dados. Nos modelos CA, $p(x|c)$ é compartilhado entre os affordances e não precisa ser ajustado quando um novo affordance é adicionado. Isso traz grandes benefícios em termos de escalabilidade em comparação à abordagem PD (SUN *et al.*, 2010).

Exemplificando, suponha que o robô precise aprender se o terreno se torna escorregadio quando é molhado. No modelo PD, o fato de o robô ter aprendido que, por exemplo, a grama, o concreto e o cascalho são transitáveis quando secos, não vai ajudá-lo a descobrir se eles se tornam escorregadios quando molhados. Isso acontece porque ao aprender um mapeamento direto entre as características desses terrenos e a transitabilidade dos mesmos, o sistema pulou um conceito intermediário, que é o de categoria. Se o robô tivesse aprendido as características de cada uma dessas categorias de terreno, aprender um novo affordance exigiria uma experimentação mínima com o novo affordance para cada categoria. Ademais, novos dados sobre as características do terreno não seriam necessários porque o robô já as conhece (por exemplo, o robô já sabe como a grama se parece).

Ainda com relação ao modelo PD versus o modelo CA, é importante enfatizar que a categorização dos objetos é um passo fundamental para o modelo CA. Se ela não for bem realizada, as predições realizadas pelo modelo PD provavelmente serão bem melhores que do que as realizadas pelo modelo CA.

Outra questão importante é decidir entre a utilização do modelo CA-full e CA-chain. Para entender quando o modelo *CA-chain* pode ser suficiente, considere que algumas categorizações para um conjunto de dados de treinamento foram obtidas, e imagine que para cada categoria $c$ construí-se o vetor-$K$ de probabilidades definido por $[p(a^k = 1 | c)]_{k=1}^{K}$. Se esse vetor for aproximadamente binário (isto é, todas as probabilidades são aproximadamente 0 ou 1), então o conhecimento da categoria é suficiente para predizer o conjunto de affordances com alta confiança. Ao contrário, se $p(a^k = 1|c)$ é próximo de 0,5, a categorização não fornece informação sobre $a^k$. Nessa situação, o modelo *CA-full* deve ser aplicado, no qual nós interpretamos $p(a^k = 1|x, c)$ como uma regra de classificação específica de categoria, mapeando $x$ para $a^k$ no contexto de uma categoria particular $c$. Comparando com o modelo PD, isso pode ser visto como uma abordagem de "mistura de especialistas" (JORDAN & JACOBS, 1994), na qual o espaço de características é dividido em domínios distintos em que regras de decisão específicas podem ser aprendidas. A suposição de que o rótulo de categoria por si só pode determinar os affordances de um objeto pode parecer muito forte, mas é muitas vezes uma boa escolha quando os dados de treinamento são escassos. Ademais, os seres humanos frequentemente fazem generalizações desse tipo, por exemplo, quando consideram que podem sentar em um objeto que se parece com uma cadeira.

## V. CONSIDERAÇÕES FINAIS

Neste trabalho, um universo bem pequeno das possibilidades de aplicação de affordances foi investigado. Entretanto, foram apresentadas algumas propostas interessantes existentes na literatura para a aplicação prática do conceito de affordance na navegação de robôs autônomos. Basicamente, foram apresentados 2 modelos de aprendizagem de affordances, o modelo PD e o modelo CA, sendo que este último pode ser dividido em CA-full e CA-chain. Também foram apresentadas algumas comparações entre esses modelos.

Embora o foco dos trabalhos apresentados seja o aprendizado de affordances de transitabilidade, nada impede que eles sejam estendidos à aplicação de outros tipos de affordances.

## REFERÊNCIAS


[1] A. Chemero and M.T. Turvey. Gibsonian affordances for roboticists. *Adaptive Behavior*, 15(4):473, 2007.

[2] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*. Series B (Methodological), pages 1–38, 1977.

[3] J.J. Gibson. *The senses considered as perceptual systems*. Houghton Mifflin, 1966.

[4] J.J. Gibson. *The ecological approach to visual perception*. Lawrence Erlbaum, 1979.

[5] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[6] D. Kim, J. Sun, S.M. Oh, J.M. Rehg, and A.F. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *Robotics and Automation,2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 518–525. IEEE, 2006.

[7] K. Kira and L.A. Rendell. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

[8] E. Sahin, M. Cakmak, M.R. Dogar, E. Ugur, and G. Uçoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447, 2007.

[9] J. Sun, T. Mehta, D. Wooden, M. Powers, J. Rehg, T. Balch, and M. Egerstedt. Learning from examples in unstructured, outdoor environments. *Journal of Field Robotics*, 23(11-12):1019–1036, 2006.

[10] J. Sun, J.L. Moore, A. Bobick, and J.M. Rehg. Learning visual object categories for robot affordance prediction. *The International Journal of Robotics Research*, 29(2-3):174, 2010.

[11] E. Ugur and E. Sahin. Traversability: a case study for learning and perceiving affordances in robots. *Adaptive Behavior*, 18(3-4):258, 2010.

[12] V.N. Vapnik. *Statistical learning theory*. Wiley-Interscience, 1998.

[13] W.H. Warren. Perceiving affordances: Visual guidance of stair climbing. *Journal of Experimental Psychology: Human Perception and Performance*, 10(5):683, 1984.