

# UMA ESTRATÉGIA ONLINE PARA PREDIÇÃO ESTRUTURADA UTILIZANDO A FORMULAÇÃO DE MÁXIMA MARGEM

Maurício Archanjo Nunes Coelho, Raul Fonseca Neto e Carlos Cristiano H. Borges

Departamento de Ciência da Computação, Universidade Federal de Juiz de Fora, 36.001-970, Minas Gerais, Juiz de Fora, Brasil.

*mauarchanjo@hotmail.com*

*raulfonsecaneto@ig.com.br*

*cchb@lnc.cbr*

**Resumo** – Os algoritmos de busca necessitam que os custos estipulados para os ambientes ou mapas sejam confiáveis. A dificuldade está justamente na definição dos custos referentes a cada tipo de área ou terreno nos mapas a serem examinados. Como se pode observar, o problema mencionado inclui a dificuldade em se determinar qual o custo de cada característica relevante presente no mapa, bem como os custos de suas possíveis combinações. A proposta do artigo é mostrar como é feita a predição desses custos em novos ambientes tendo como base a predição de dados estruturados definindo um aprendizado funcional entre domínios de entrada e saída, estruturados e arbitrários. O problema de aprendizado em questão é normalmente formulado como um problema de otimização convexa de máxima margem bastante similar a formulação de máquinas de vetores suporte multi-classe. Como técnica de solução realizou-se a implementação do algoritmo MMP (Maximum Margin Planning) [3]. Como contribuição, desenvolveu-se e implementou-se dois algoritmos alternativos, o primeiro denominado Perceptron Estruturado e o segundo Perceptron Estruturado com Margem, ambos os métodos de relaxação baseados na formulação do Perceptron. Os mesmos foram analisados e comparados. Posteriormente temos a exploração dos ambientes por um agente inteligente utilizando técnicas de aprendizado por reforço. Tornando todo o processo, desde a análise do ambiente e descoberta de custos, até sua exploração e planejamento do caminho, um completo processo de aprendizado.

**Palavras-chave** – Aprendizado de Máquina, Planejamento com Máxima Margem, Perceptron Multi-classe, Planejamento de Caminhos e Predição de Dados Estruturados.

## 1 Introdução

O objetivo do presente trabalho é possibilitar a predição de custos em novos ambientes ou mapas tendo como base a predição de dados estruturados definindo um aprendizado funcional entre domínios de entrada e saída, estruturados e arbitrários. Aplicado ao problema de planejamento de caminhos este aprendizado torna possível a obtenção de planos ou políticas a partir da percepção das características dos mapas, sendo de grande importância a sua utilização em sistemas de navegação de robôs móveis. Frequentemente, nestes sistemas de navegação, ocorre uma clara distinção entre os níveis de percepção e de planejamento, sendo o planejamento de caminhos obtido somente a partir do prévio conhecimento da matriz de custos relacionada ao espaço de estados-ações do problema.

Neste problema, relacionado à predição de custos, tem-se como dados de entrada um conjunto de caminhos ou trajetórias escolhidos por um especialista em um ou mais mapas. Estes caminhos são escolhidos de forma a beneficiarem algum tipo de estratégia relacionada à presença ou não de determinadas características, servindo de base ou de exemplo na definição do mapeamento de custos de novos mapas. Desta forma, o mapeamento obtido possibilitará o planejamento de novos caminhos em novos ambientes de forma similar ao tipo de estratégia escolhida pelo especialista.

O problema de aprendizado em questão é formulado como um problema de otimização convexa de máxima margem estruturado bastante similar a formulação de máquinas de vetores suporte multi-classe [2]. Como técnica de solução realizou-se a implementação do algoritmo MMP proposto por Ratliff et al. (2006) que utiliza uma técnica de sub-gradiente, aplicada à minimização de uma função objetiva irrestrita, relacionada à formulação de máxima margem. Também, de forma alternativa, para a solução do problema de predição, propôs-se a implementação de um método de relaxação, denominado Perceptron Estruturado, o qual tem semelhanças com o esquema de correção proposto para o algoritmo perceptron multi-classe (Bakır et al., 2006).

Com o intuito de comprovar a eficiência desta abordagem bem como a correteza dos algoritmos implementados, foram realizados diversos testes com diferentes dados de entrada refletidos na escolha de diferentes mapas, caminhos e características. Os resultados alcançados corresponderam ao esperado demonstrando sempre uma forte associação entre a saída obtida representada pelos custos aprendidos e o planejamento de novos caminhos seguindo a estratégia definida pelo especialista para o conjunto de treinamento.

## 2 Modelo de Predição de Dados Estruturados

### 2.1 Formulação de Máxima Margem

Dado um conjunto de treinamento  $S = \{(x(i), y(i), i=1, \dots, m)\}$  formado por uma coleção de pares, sendo cada par formado por uma amostra representada por um objeto estruturado  $x(i)$  (tal como um mapa) e uma solução desejada  $y(i)$  (tal como um caminho) deseja-se obter um vetor de parâmetros  $w$  tal que [1]:

$$\arg \text{Max}_{y \in Y(i)} \{w^T \cdot f(x(i), y)\} \approx y(i), i=1, \dots, m, \quad (1)$$

sendo  $Y(i)$  o espaço de todas as soluções ou exemplos possíveis dependente do objeto estruturado  $x(i)$ . Muito embora a cardinalidade de  $Y(i)$  possa ser muito elevada, é possível a utilização de uma técnica de geração que resolve de forma eficiente a determinação do falso-exemplo  $y$ .

Ou seja, o aprendizado do vetor de parâmetros  $w$  permite que a melhor solução encontrada para cada par da coleção seja exatamente a solução proposta no conjunto de treinamento. A abordagem mais utilizada na solução deste problema, relacionado à predição de dados estruturados, se baseia em uma generalização do princípio da formulação de máxima margem [4] empregado nas máquinas de vetores suportes [6] e implica na solução do seguinte problema de programação quadrática:

$$\begin{aligned} & \text{Min } \frac{1}{2} \|w\|^2 \\ & \text{Sujeito a:} \end{aligned} \quad (2)$$

$$w^T \cdot f_i(y(i)) \geq \text{Max}_{y \in Y(i)} \{w^T \cdot f_i(y) + l_i(y)\}, i=1, \dots, m,$$

sendo  $f_i(y) = f(x(i), y)$  e a função  $l_i(y) = l(y(i), y)$  definida como uma função de perda que escalona o valor geométrico de margem requerido para o falso-exemplo  $y$  em relação à amostra  $y(i)$ .

A margem de uma amostra  $(x(i), y(i))$  sobre qualquer outro exemplo  $y \in Y(i)$  é interpretada como:

$$w^T \cdot (f_i(y(i)) - \text{Max}_{y \in Y(i)} \{f_i(y)\}) / \|w\|_2 \quad (3)$$

Em uma versão modificada pode-se considerar a utilização de um parâmetro de margem identificado por  $\gamma$  o qual deve ser maximizado com a restrição adicional de controle da norma do vetor de parâmetros  $w$ . Assim o problema de otimização convexa, sugerido em [5], pode ser reescrito na forma:

$$\begin{aligned} & \text{Max } \gamma \\ & \|w\| = 1 \\ & \text{Sujeito a:} \end{aligned} \quad (4)$$

$$w^T \cdot f_i(y(i)) - \text{Max}_{y \in Y(i)} \{w^T \cdot f_i(y) + l_i(y)\} \geq \gamma, i=1, \dots, m$$

Desta forma, a minimização da norma do vetor  $w$  ou a equivalente maximização da margem  $\gamma$  resulta na obtenção de uma solução de máxima margem que satisfaz o conjunto de restrições. Para os casos de não separabilidade pode-se admitir a introdução de variáveis de folga como utilizado nas máquinas de vetores suporte. É interessante observar que o problema:  $\text{Max}_{y \in Y(i)} \{w^T \cdot f_i(y) + l_i(y)\}$  tem precisamente a mesma forma do problema de predição para o qual se necessita aprender os parâmetros. Em geral, este problema é de complexidade polinomial e de fácil solução (tal como um problema de caminho mínimo sobre um grafo de estados).

## 2.2 Problema de Predição de Custos

O problema de predição de custos foi formulado no contexto de uma coleção de mapas e caminhos tendo como objetivo a maximização de recompensas. Neste problema cada mapa é representado por um grafo de estados formado um conjunto de pares estados-ações e cada estado de um mapa é caracterizado por um conjunto de características. Cada caminho é representado por um vetor de frequências  $\mu$  que indica a ocorrência ou não das diversas ações ou arcos para cada possível par estado-ação ou transição definido pelo mapa. Cada mapa é representado por uma matriz  $F$  de características, distribuída para todo conjunto de pares estado-ações. Assim o produto  $F_i \cdot \mu$  representa a quantidade de cada característica presente no  $i$ -ésimo mapa em função da escolha do caminho representado por  $\mu$ . Finalmente o produto  $w^T \cdot F_i \mu$  representa a recompensa total do caminho  $\mu$  no  $i$ -ésimo mapa computado em função dos valores apresentados pelo vetor de parâmetros  $w$ . O equacionamento deste problema foi apresentado por Ratliff et al. (2006), tendo a forma:

$$\begin{aligned} & \text{Min } \frac{1}{2} \|w\|^2 \\ & \text{Sujeito a:} \end{aligned} \quad (5)$$

$$w^T \cdot F_i \mu_i \geq \text{Max}_{\mu \in G(i)} \{w^T \cdot F_i \mu + l_i^T \cdot \mu\}, i=1, \dots, m,$$

sendo  $G(i)$  a representação de todas as escolhas possíveis de caminhos referentes ao  $i$ -ésimo mapa.

Sem perda de generalidade optou-se neste trabalho pelo problema de minimização de custos. Assim, o equacionamento toma a forma:

$$\text{Min } \frac{1}{2} \|w\|^2$$

Sujeito a:

$$w^T \cdot F_i \mu_i \leq \text{Min}_{\mu \in G(i)} \{ w^T \cdot F_i \mu + l_i^T \cdot \mu \}, i=1, \dots, m,$$

(6)

### 3 Técnicas de Solução

#### 3.1 Formulação de Programação Quadrática

Para a solução do problema de maximização de recompensas apresentado em (5) Ratliff et al. (2006) propõem a minimização de uma função objetiva não diferenciável mas convexa, obtida da relaxação do problema, com o uso de uma técnica de sub-gradiente. Assim deve-se minimizar a função:

$$L(w) = \frac{1}{2} \|w\|_2^2 + C \cdot \sum_i \text{Max}_{\mu \in G(i)} \{ w^T \cdot F_i \mu + l_i^T \cdot \mu \} - w^T \cdot F_i \mu_i \quad (7)$$

Esta função possui como sub-gradiente:

$$g = w + C \cdot (\sum_i F_i \mu_i^* - F_i \mu_i), \quad (8)$$

computado em função da determinação da política ótima  $\mu^*$  relativa ao caminho ótimo da matriz de custos de cada mapa de entrada obtida da equação:  $(w^T \cdot F_i + l_i^T)$ . Para tanto se utiliza o algoritmo A\* que apresenta complexidade quadrática em relação à quantidade de estados ou células.

Assim o vetor  $w$  fica atualizado na forma:

$$w \leftarrow w - \eta \cdot g, \text{ ou } w \leftarrow w \cdot (1 - \eta) - \eta \cdot (C \cdot (\sum_i F_i \mu_i^* - F_i \mu_i)) \quad (9)$$

para uma taxa  $\eta$  de aprendizado reduzida de forma gradativa de acordo com o número de iterações e um parâmetro de regularização positivo  $C$ .

#### 3.2 Método de Relaxação

Se não for imposta à condição de maximização da margem pode-se reformular o problema de predição com minimização de custos como um problema de viabilidade de um sistema de inequações na forma:

Encontrar um vetor  $w$  viável, tal que:

$$w^T \cdot F_i \mu_i \leq \text{Min}_{\mu \in G(i)} \{ w^T \cdot F_i \mu + l_i^T \cdot \mu \}, i=1, \dots, m, \quad (10)$$

Ou, de modo alternativo:

$$w^T \cdot F_i \mu_i - \text{Min}_{\mu \in G(i)} \{ w^T \cdot F_i \mu + l_i^T \cdot \mu \} \leq 0, i=1, \dots, m \quad (11)$$

Para a solução deste problema pode-se empregar uma variante do algoritmo primal do perceptron o qual foi denominado Perceptron Estruturado com a seguinte regra de correção:

Para cada mapa representado pelo índice  $i=1, \dots, m$ :

$$\text{Se } (w^T \cdot F_i \mu_i - \text{Min}_{\mu \in G(i)} \{ w^T \cdot F_i \mu + l_i^T \cdot \mu \}) > 0, \quad (12)$$

ou seja, se uma inequação não está sendo satisfeita, então atualizar o vetor  $w$  na forma:

$$w \leftarrow w - \eta \cdot (F_i \mu_i - F_i \mu^*), \text{ sendo } 0 < \eta \leq 1, \quad (13)$$

sendo  $\eta$  uma taxa de aprendizado constante e o vetor  $\mu^*$  determinado através da comparação direta de todo conjunto de caminhos existentes propostos no conjunto de treinamento  $S$ , ou seja:

$$\mu^* = \arg \text{Min}_{\mu \in S \text{ e } \mu \neq y(i)} \{ w^T \cdot F_i \mu + l_i^T \cdot \mu \}. \quad (14)$$

Assim, o vetor  $\mu^*$  representa sempre a melhor alternativa de caminho desconsiderando o caminho correto ou escolhido associado a cada mapa fornecido. Caso a diferença não seja positiva certifica-se que o caminho escolhido tem um custo inferior a melhor alternativa não sendo neste caso necessária a correção do vetor de parâmetros  $w$ . Observa-se que neste caso a função de perda  $l_i()$  não escalona o valor geométrico da margem mas atua como um fator de penalidade com o objetivo de evitar a ocorrência de custos negativos nos mapas. Isto é necessário devido à utilização do algoritmo A\* para a determinação dos caminhos ótimos relacionados ao conjunto de teste. A vantagem de se utilizar a escolha da melhor alternativa de caminho entre um conjunto de caminhos existentes e não sobre todos os caminhos possíveis está na redução do esforço

computacional. Neste novo processo a complexidade é linear e relacionada à quantidade de mapas existentes no conjunto de treinamento, lembrando que os custos dos caminhos em um mapa são resultados do produto interno do vetor  $w$  pelos vetores pré-computado de características dos caminhos representados por  $F_i\mu$ .

A regra de correção obtida para o algoritmo Perceptron Estruturado pode ser facilmente derivada seguindo o raciocínio descrito a seguir. Seja o vetor diferença definido para cada mapa da coleção na forma:

$$\delta_i = F_i\mu_i - F_i\mu^*. \quad (15)$$

A condição de viabilidade do algoritmo Perceptron é dada por:

$$-w^T \cdot \delta_i \geq 0, \quad i=1, \dots, m. \quad (16)$$

Desta forma, a função de perda relacionada à estratégia de minimização de custos será dada por:

$$J(w) = \sum_i \text{Max}\{0, w^T \cdot \delta_i\}. \quad (17)$$

Esta função deve ser minimizada. Portanto, pode-se definir o gradiente local relativo a  $i$ -ésima amostra como:

$$\nabla_w J(w) = \delta_i = F_i\mu_i - F_i\mu^*. \quad (18)$$

Assim, caso ocorra um erro relacionada a  $i$ -ésima amostra, ou seja:

$$-w^T \cdot \delta_i < 0 \text{ ou } w^T \cdot (F_i\mu_i - F_i\mu^*) > 0, \quad (19)$$

utiliza-se a seguinte regra de correção para o vetor de custos:

$$w \leftarrow w - \eta \cdot (F_i\mu_i - F_i\mu^*), \quad 0 < \eta \leq 1, \text{ equivalente, portanto a equação de correção proposta em (13).}$$

De outra forma, para um problema de maximização de recompensas tem-se:

$$\delta_i = \text{Max}_{\mu \in G(i)} (F_i\mu + l_i\mu) - F_i\mu_i = F_i\mu^* - F_i\mu_i \quad (20)$$

A condição de viabilidade do algoritmo Perceptron permanece a mesma sendo dada por:

$$-w^T \cdot \delta_i \geq 0, \quad i=1, \dots, m. \quad (21)$$

Assim tem-se a mesma função de perda apresentada em (17) definindo o gradiente local:

$$\nabla_w J(w) = \delta_i = F_i\mu^* - F_i\mu_i. \quad (22)$$

Portanto, caso ocorra um erro relacionada a  $i$ -ésima amostra, ou seja:

$$-w^T \cdot \delta_i < 0 \text{ ou } w^T \cdot (F_i\mu^* - F_i\mu_i) > 0, \quad (23)$$

utiliza-se a seguinte regra de correção para o vetor de custos:

$$w \leftarrow w - \eta \cdot (F_i\mu^* - F_i\mu_i), \quad 0 < \eta \leq 1. \quad (24)$$

É interessante observar a similaridade entre a equação (24) e a equação de correção do algoritmo MMP representada em (9). A diferença principal está relacionada à redução dos valores das componentes do vetor  $w$  e consequentemente no valor da sua norma no sentido de atender ao objetivo de minimização da função apresentada em (7).

## 4 Resultados Experimentais

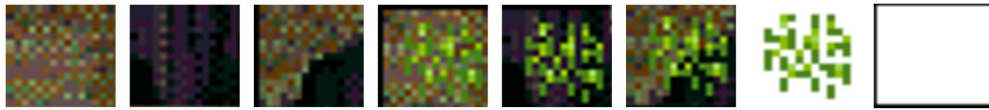
### 4.1 Conceituação do Problema

Para validar os conceitos e algoritmos desenvolvidos neste trabalho optou-se pela solução de um problema prático de planejamento relacionado à determinação de caminhos em um mapa quadriculado. O problema foi definido como um problema de minimização de custos ou de obtenção do caminho mínimo. Os mapas foram simulados com o uso de matrizes 5x5. Cada componente de uma matriz define uma célula ou estado contendo um tipo de terreno. Para a realização de movimentos foram permitidas quatro ações possíveis relacionadas, respectivamente, as direções: norte, leste, sul e oeste. Também, determinou-se uma célula de origem e uma célula de destino para todos os caminhos a fim de conseguir uma melhor visualização e comparação dos resultados.

No exemplo foram utilizados seis mapas 5x5 para o conjunto de treinamento. Cada mapa contém as características do terreno e o melhor caminho dado pelo especialista do domínio. Para o conjunto de teste, foram utilizados doze mapas.

Nos mapas pode-se ter a presença ou não de cada característica bem como a sua intensidade. Desta forma pode-se combinar a utilização de características na definição de um tipo de terreno de cada célula. Como exemplo pode-se mencionar a existência de um terreno contendo um pouco de vegetação, um pouco de rocha e rastros de uma antiga trilha. Para uma melhor

visualização e simplificação dos resultados optou-se somente pela presença ou não de cada característica e suas combinações possíveis, definindo assim oito tipos diferentes de terreno de acordo com a cardinalidade do conjunto potência. As características e suas combinações utilizadas estão representadas na Figura 1:



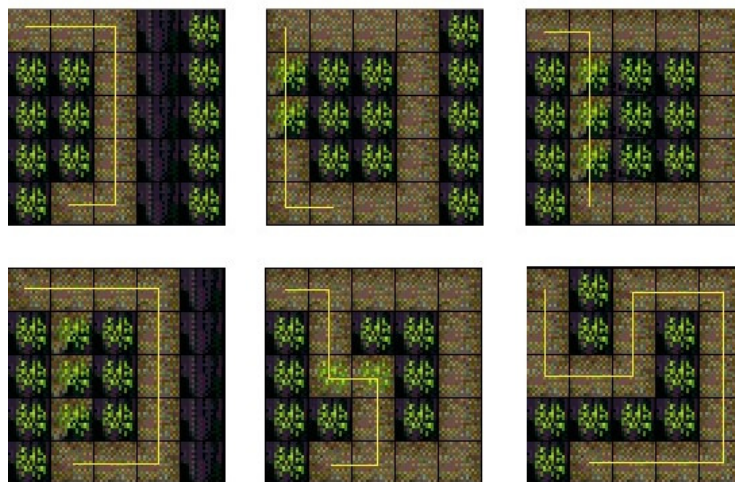
**Figura 1:** No primeiro quadro tem-se somente trilha; no segundo, somente rocha; no terceiro: trilha e rocha; no quarto: trilha e vegetação; no quinto: rocha e vegetação; no sexto: trilha, rocha e vegetação (trilha abandonada); no sétimo, somente vegetação e, finalmente, no oitavo tem-se a ausência de características.

## 4.2 Resultados do Conjunto de Treinamento

A Figura 2 apresenta os mapas utilizados no conjunto de treinamento incluindo os caminhos traçados pelo especialista. Pode-se observar nesta figura que todos os caminhos possuem a célula 1 correspondente a entrada (1,1) da matriz como origem e a célula 22 correspondente a entrada (5,2) da matriz como destino. Pelos resultados alcançados constatou-se que ambos os algoritmos convergiram para os mesmos caminhos definidos pela estratégia do especialista, indicando desta forma que o aprendizado foi bem sucedido e as matrizes de custos obtidas refletem a estratégia de planejamento adotada pelo mesmo.

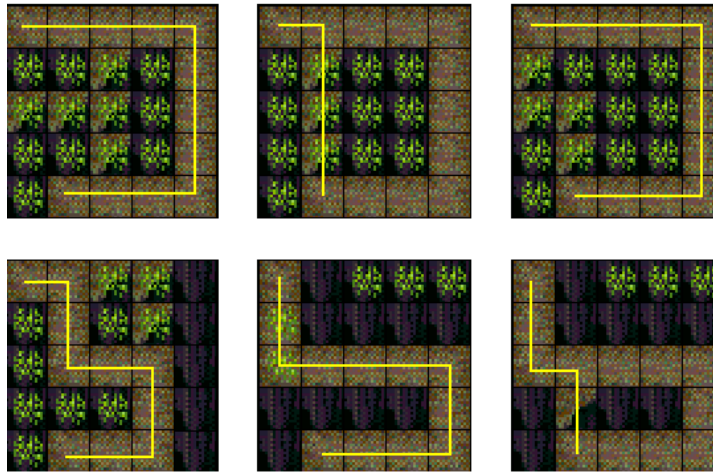
Como parâmetros do algoritmo MMP foram utilizados: taxa de aprendizado 0.5, valor de perda 2 e parâmetro de regularização 1. O vetor  $w$  foi inicializado com [0.1,0.1,0.1]. O algoritmo MMP necessitou de 17 iterações para convergir com tempo de execução aproximado de 0.035 segundos. Também foi estipulado um máximo de 1000 iterações a fim de analisar o efeito sobre a maximização da margem, ou seja, como o algoritmo e sua margem se comportam tendo um alto número de iterações fixos. Seu tempo de execução é 0.402 segundos. Como parâmetros do algoritmo Perceptron Estruturado foram utilizados: taxa de aprendizado 0.2, valor de penalidade 2. O vetor  $w$  também foi inicializado com [0.1,0.1,0.1]. O algoritmo necessitou de somente 6 iterações para convergir com tempo de execução aproximado de 0.015 segundos.

Os parâmetros do algoritmo Perceptron Estruturado com Margem foram os mesmos do Perceptron Estruturado e a separação das classes se dá de maneira idêntica, com mesmo número de iterações e tempo. Foi também estipulado um máximo de 1000 iterações para analisar a maximização da margem e o tempo de execução foi de 0.125 segundos.



**Figura 2:** Mapas de treinamento com seus respectivos caminhos traçados pelo especialista do domínio.

Na análise da Fig. 2 pode-se concluir que, segundo o especialista, geralmente é melhor escolher um caminho que segue uma trilha sem vegetação e sem rocha. Entretanto, caso o caminho requeira uma trilha de maior comprimento o especialista poderá sugerir que o caminho passe por uma trilha com vegetação, com rocha ou com vegetação e rocha. Pode-se perceber também que o especialista evita fortemente as células contendo vegetação e rocha, mas sem a existência de trilha.



**Figura 3: Mapas de prova com seus respectivos caminhos traçados pelo especialista do domínio.**

A Fig. 3 é um conjunto de controle que não participa do processo de aprendizado, porém é utilizado no conjunto de teste, sendo respectivamente seus seis primeiros mapas. Isso é feito para uma melhor comparação dos resultados obtidos com aquilo que era esperado pelo especialista.

É importante ressaltar que os resultados obtidos atendem a um processo de escolha relacionado à técnica de planejamento por máxima margem. Isto indica que a escolha do especialista reflete também o melhor caminho ou aquele de menor custo quando comparado a todas alternativas presentes no conjunto de treinamento considerando as matrizes de custos aprendidas pelos algoritmos. Ou seja, pode-se citar o fato de que, para cada mapa, nenhuma escolha alternativa dos caminhos apresentados possuirá menor custo que o caminho sugerido pelo especialista para o mesmo. Para demonstrar esta propriedade do planejamento foram elaboradas 4 tabelas (Tabelas 1, 2, 3 e 4) que mostram os valores de custos geométricos para todos os mapas e caminhos do conjunto de treinamento para cada algoritmo. Estes valores são obtidos dividindo-se os valores dos custos dos caminhos pelas respectivas normas do vetor  $w$ .

Sendo que, para o algoritmo MMP até a separação das classes obteve-se o vetor de custos  $w$ ,  $w[0]$ : 0.819397,  $w[1]$ : 0.209730,  $w[2]$ : -5.461417, com norma = 5.526524, já com 1000 iterações obteve-se:  $w[0]$ : 0.530317,  $w[1]$ : 0.450290,  $w[2]$ : -5.367245, com norma = 5.409307. Para o algoritmo Perceptron Estruturado obteve-se o vetor de custos:  $w[0]$ : 1.100000,  $w[1]$ : 0.900000,  $w[2]$ : -4.70000, com norma = 4.910193. Para o Perceptron com margem em suas 1000 iterações obteve-se:  $w[0]$ : 0.942207,  $w[1]$ : 0.813118,  $w[2]$ : -4.900445 com norma = 5.056014 e margem máxima final: 0.171629.

Uma medida de qualidade da solução proposta por cada treinamento pode ser considerada como a diferença entre o custo geométrico do caminho do especialista e o do caminho alternativo de menor custo. Os valores em negrito correspondem aos custos geométricos do caminho do especialista e os custos geométricos dos caminhos alternativos de menor custo.

**Tabela 1 – Custos geométricos do algoritmo MMP.**

Mapa	Caminho 1	Caminho 2	Caminho 3	Caminho 4	Caminho 5	Caminho 6	Margem
1	<b>0.68218</b>	5.18501	4.01057	6.55951	<b>3.03105</b>	13.13026	<b>2.34887</b>
2	4.20548	<b>0.85970</b>	4.01057	<b>0.87708</b>	5.37992	11.22972	<b>0.01738</b>
3	4.20548	5.18501	<b>1.04591</b>	4.40039	<b>3.40348</b>	6.34577	<b>2.35756</b>
4	4.20548	5.18501	<b>1.04591</b>	<b>0.87708</b>	3.40348	12.02819	<b>0.16883</b>
5	1.89456	5.18501	<b>1.69965</b>	4.40039	<b>0.75807</b>	5.06101	<b>0.94157</b>
6	3.03105	<b>2.83614</b>	4.01057	5.57482	4.20548	<b>1.46181</b>	<b>1.37432</b>

**Tabela 2 – Custos geométricos do algoritmo MMP com 1000 iterações.**

Mapa	Caminho 1	Caminho 2	Caminho 3	Caminho 4	Caminho 5	Caminho 6	Margem
1	<b>0.82247</b>	5.27904	4.10615	6.50595	<b>3.16825</b>	13.32495	<b>2.34578</b>
2	4.34114	<b>0.94985</b>	4.10615	<b>1.05746</b>	5.51403	11.50793	<b>0.10760</b>
3	4.34114	5.27904	<b>1.13104</b>	4.57613	<b>3.53062</b>	6.63518	<b>2.39958</b>
4	4.34114	5.27904	<b>1.13104</b>	<b>1.05746</b>	3.53062	12.08367	<b>0.07357</b>
5	2.07855	5.27904	<b>1.84356</b>	4.57613	<b>0.98885</b>	5.44749	<b>0.85470</b>
6	3.16825	<b>2.93326</b>	4.10615	5.74902	4.34114	<b>1.76243</b>	<b>1.17082</b>

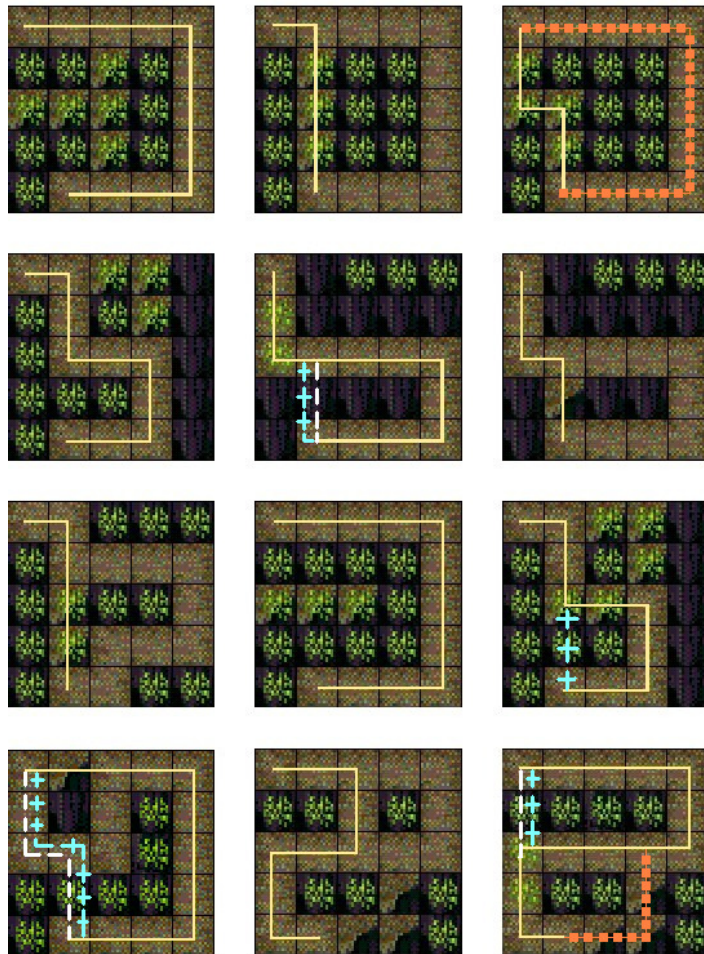
**Tabela 3 – Custos geométricos do algoritmo Perceptron Estruturado.**

Mapa	Caminho 1	Caminho 2	Caminho 3	Caminho 4	Caminho 5	Caminho 6	Margem
1	<b>1.85328</b>	6.78181	5.41730	8.28887	<b>4.58230</b>	17.24983	<b>2.72901</b>
2	5.94681	<b>2.13840</b>	5.41730	<b>2.38279</b>	7.31132	15.70202	<b>0.24439</b>
3	5.94681	6.78181	<b>2.54572</b>	6.47632	<b>5.39693</b>	9.83668	<b>2.85121</b>
4	5.94681	6.78181	<b>2.54572</b>	<b>2.38279</b>	5.39693	15.74276	<b>0.16292</b>
5	3.40108	6.78181	<b>2.87157</b>	6.47632	<b>2.21987</b>	8.43144	<b>0.65170</b>
6	4.58230	<b>4.05279</b>	5.41730	7.84083	5.94681	<b>3.97133</b>	<b>0.08146</b>

**Tabela 4 – Custos geométricos do algoritmo Perceptron Estruturado com Margem.**

Mapa	Caminho 1	Caminho 2	Caminho 3	Caminho 4	Caminho 5	Caminho 6	Margem
1	<b>1.52232</b>	6.35300	5.03659	7.73519	<b>4.15513</b>	16.10454	<b>2.63281</b>
2	5.47154	<b>1.78172</b>	5.03659	<b>1.95727</b>	6.78794	14.48772	<b>0.17554</b>
3	5.47154	6.35300	<b>2.12890</b>	5.90649	<b>4.84948</b>	8.87492	<b>2.72058</b>
4	5.47154	6.35300	<b>2.12890</b>	<b>1.95727</b>	4.84948	14.65284	<b>0.17162</b>
5	2.99955	6.35300	<b>2.12890</b>	5.90649	<b>1.84396</b>	7.53298	<b>0.72063</b>
6	4.15513	<b>3.72018</b>	5.03659	7.22289	5.47154	<b>3.26211</b>	<b>0.45806</b>

### 4.3 Resultados do Conjunto de Teste



**Figura 4: Mapas de testes com os custos já associados a cada característica e os caminhos traçados pelo algoritmo A\*.**

Na Figura 4 foram apresentados os resultados dos dois algoritmos aplicados ao conjunto de teste. Primeiramente, utilizou-se o algoritmo MMP e, em seguida, utilizou-se o algoritmo Perceptron Estruturado. Para a obtenção da matriz de custos associada a cada mapa multiplicou-se o vetor  $w$  de custos pela sua matriz  $F$  de características. A matriz de características possui dimensões  $3 \times 25 \times 4$  representando a ocorrência das 3 características para cada um dos possíveis pares estado-ação do mapa, estando os

valores das características relacionados ao tipo de terreno para o qual a ação associada ao estado incide. Assim, após a realização do produto  $w.F$  tem-se uma matriz  $25 \times 4$  representando o custo de transição relativo a cada par estado-ação a qual finalmente é reduzida para uma matriz  $5 \times 5$  representando o custo final de cada célula ou estado. Esta redução é possível considerando o fato de que todas as transições para uma mesma célula possuem o mesmo custo associado obtido diretamente do custo da célula ou do estado sucessor.

A linha contínua amarela são os caminhos achados com base no vetor de custo do algoritmo MMP até a separação das classes, ou seja, 17 iterações. Na maioria dos mapas, os caminhos coincidiram para todos os vetores de custos dos vários algoritmos, porém quando não ocorreu, foram traçados desvios com formas diferentes para os quatro algoritmos como segue. O caminho com quadrados laranja, mapas três e doze, pertence aos desvios feitos pelo algoritmo MMP com suas 1000 iterações. O caminho com a linha tracejada branca representa o algoritmo Perceptron Estruturado, mapas cinco, dez e doze. O algoritmo Perceptron Estruturado com Margem, mapas cinco, nove, dez e doze, representado pelo caminho com cruces azuis claras.

Após a obtenção das matrizes de custos obtêm-se o caminho ótimo ou caminho mínimo com a utilização do algoritmo  $A^*$ . Para tanto, a matriz de custos não deve conter custos negativos no sentido de observar a condição de monótona crescente da função de avaliação. Observando a fig. 4 pode-se constatar que os resultados obtidos foram bastante semelhantes e satisfatórios.

Na análise dos resultados pode-se observar que os caminhos traçados corresponderam em sua maioria às expectativas do especialista. Além disso, pode-se notar que os caminhos escolhidos quase sempre evitaram as células sem trilha e com vegetação e rocha, mesmo quando o caminho não correspondeu exatamente ao do especialista definido no conjunto de prova. Somente os mapas 3 e 5 diferiram do caminho do conjunto de prova, com o MMP e o Perceptron respectivamente.

## 5 Conclusão

Neste trabalho foi apresentado o algoritmo Perceptron Estruturado, uma alternativa a técnica de solução do algoritmo MMP. Como foi observado tanto nas Fig. 2, 3 e 4, quanto nas Tabelas 1, 2, 3 e 4, os algoritmos apresentaram resultados bem semelhantes, todavia o algoritmo Perceptron Estruturado exige um menor gasto de memória e processamento, o que o deixou mais rápido. Isto se deve principalmente ao fato do mesmo não precisar executar o algoritmo  $A^*$  em seu processo de aprendizado.

Constatou-se, também, que o algoritmo Perceptron Estruturado prediz eficientemente os mapas de custos de novos ambientes a partir de um pequeno número de mapas de treinamento, possibilitando que a sua utilização seja estendida para outras aplicações envolvendo o problema de predição de dados estruturados.

## 6 Referências

- [1] Bakır, G., Hofmann, T., Schölkopf, B., Smola, A., Taskar, B., Vishwanathan, S.V.N., 2006. Predicting Structured Data. MIT Press.
- [2] Weston, J. e Watkins, C., 2003. Multi-class support vector machines. URL [citesser.ist.psu.edu/article/weston98multiclass.html](http://citesser.ist.psu.edu/article/weston98multiclass.html).
- [3] Ratliff, N., Bagnell, J. A., e Zinkevich, M., 2006. Maximum margin planning. Em Twenty Second International Conference on Machine Learning. (ICML06), pp. 729–736.
- [4] Taskar, B., Chatalbashev, V., Koller, D., e Guestrin, C., 2005. Learning structured prediction models: a large margin approach. Em Twenty Second international Conference on Machine Learning. (ICML05), pp. 896–903.
- [5] Tsochantaridis, I., Joachims, T., Hofmann, T. e Altun, Y., 2005. Large Margin Methods for Structured and Interdependent Output Variables. Em Journal of Machine Learning, pp 1453–1484.
- [6] Vapnik, V., 1998. Statistical Learning Theory. Wiley and Sons Inc.