

## 3 Aprendizagem

Neste capítulo descrevemos como as Galerias Inteligentes usam a interação do usuário com sua interface para aprender suas preferências, e como esse conhecimento é aplicado para ajudar o usuário.

### 3.1 Máquinas de aprendizagem

No capítulo 2, apresentamos uma visão geral das Galerias Inteligentes como uma nova abordagem que usa parâmetros selecionados pelo usuário em uma interface de galerias para aprender suas preferências. Isto caracteriza o problema de aprendizagem supervisionada, ou seja, aprendizagem por exemplo. Nessa abordagem, é necessário criar um banco de dados  $\Gamma$  (*training set*) que caracterize cada parâmetro  $\omega_i$  na forma de um vetor  $x_i \in \mathbb{R}^p$  (*input*) associado a uma classificação  $c_i$  (*output*), que no nosso caso representa um bom ou mau parâmetro, de acordo com a avaliação do usuário.

Para incluir um parâmetro  $\omega_i$  no banco de dados  $\Gamma$ , é necessário extrair descritores  $\varphi_j$  que representem suas principais características usando números. Formalmente, dado um parâmetro  $\omega_i \in \Omega$ , cada descritor será representado como uma aplicação  $\varphi_j: \Omega \mapsto \mathbb{R}^{p_j}$ , que avalia parâmetros e retorna vetores de números reais. Estes vetores são concatenados em um único vetor  $x_i \in \mathbb{R}^p$ , tal que  $x_i = \varphi(\omega_i) = (\varphi_1(\omega_i), \varphi_2(\omega_i), \dots, \varphi_m(\omega_i))$ , onde  $p = \sum_{k=1}^m p_k$ .

Além disso, representamos a classificação binária do usuário (como boa ou ruim) usando uma variável  $c_i \in \{-1, 1\}$ . Os pares  $(\varphi(\omega_i), c_i)$  vão representar o dado associado a  $\omega_i$  no banco de dados  $\Gamma$ , que deve ser utilizado por algum método que realize aprendizagem supervisionada.

Neste trabalho, usamos uma derivação das Máquinas de Suporte Vetorial, uma classe de métodos que nos possibilita realizar classificação não linear. Dentre suas qualidades, destacamos:

- Eficiência computacional: O cálculo da função classificadora e avaliação é eficiente e paralelizável;

- Boa adaptação em bancos de dados esparsos: apenas dados na fronteira das classes, ou vetores de suporte, são considerados;
- Complexidade computacional pouco dependente da dimensão do espaço de entrada  $\mathbb{R}^p$ : o cálculo da função classificadora e sua avaliação tem sua complexidade dominada pela quantidade de amostras do banco de dados, o que nos permite adicionar novos descritores na formulação do problema sem prejudicar consideravelmente a eficiência;
- Classificação robusta a dados imprecisos: em diversos cenários, o usuário pode ter dificuldade em classificar parâmetros em regiões próximas à fronteira das classes. Usando uma formulação com margem flexível, as Máquinas de Suporte Vetorial consideram apenas dados distantes da fronteira para realizar classificações;
- Matematicamente bem fundamentada: Ótima pela teoria de aprendizagem estatística Vapnik-Chervonenkis (45);

O apêndice A contém uma introdução sobre esses métodos.

Dado um banco de dados representado por um conjunto

$$\Gamma = \{(x_i, c_i) | x_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n, \quad (3-1)$$

desejamos inicialmente criar um classificador não linear  $g: \mathbb{R}^p \mapsto \{-1, 1\}$  que, dado um parâmetro  $\omega_i$  representado por seu vetor de descritores  $x_i = \varphi(\omega_i)$ , nos retorne uma classificação  $g(x_i)$  que imite as preferências do usuário, de acordo com o banco de dados. Usando a formulação não linear com margem flexível das Máquinas de Suporte Vetorial descrita no apêndice A, criamos um classificador não linear  $g: x \mapsto \text{sign}(\hat{f}(x))$ , onde  $\hat{f}$  é uma função linear em um espaço de alta dimensão, ou dimensão infinita  $\mathcal{F}$ :

$$\hat{f}(x) = \sum_{i \in SV} [\alpha_i c_i \langle \phi(x_i), \phi(x) \rangle] + b. \quad (3-2)$$

Aqui, a função  $\phi: \mathbb{R}^p \mapsto \mathcal{F}$  mapeia não linearmente os descritores para o espaço  $\mathcal{F}$ , e é representada implicitamente por *kernels* do tipo  $K: \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}$ . Os valores  $\alpha_i$  são multiplicadores de Lagrange e  $b$  está relacionado com o deslocamento do hiperplano de margem máxima em relação à origem.

Para escolher o *kernel* mais eficiente para nosso método, poderíamos comparar a eficiência de diversos *kernels*. Porém, optamos, por simplicidade, pelo *kernel* Gaussiano, dado por

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right). \quad (3-3)$$

Para expandir as funcionalidades das Galerias Inteligentes, desejamos também que a função classificadora nos possibilite ordenar parâmetros, do melhor para o pior, respeitando as preferências do usuário. Porém, a função de classificação  $g$  retorna apenas um valor binário representando a classe à qual um dado parâmetro pertence.

Para obter uma avaliação contínua consideramos a seguinte idéia: a distância dos descritores  $x_i = \varphi(\omega_i)$  ao hiperplano  $\hat{f}(x) = 0$  nos dá uma boa aproximação da qualidade do parâmetro  $\omega_i$ . Por exemplo, pontos distantes do hiperplano devem representar parâmetros muito bons ou muito ruins, de acordo com o sub-espço ao qual pertencem. Baseando-se nessa intuição, usaremos para ordenação de parâmetros a função  $\hat{f}$  (sem o operador sinal).

### 3.2

#### Usos das máquinas de aprendizagem nas Galerias Inteligentes

Apresentaremos agora os principais usos das Galerias Inteligentes relacionadas com sua máquina de aprendizagem. Destacaremos seis usos.

**Aprendizagem das preferências do usuário** A interação da máquina de aprendizagem com a interface das Galerias Inteligentes começa no momento em que o usuário seleciona imagens  $\pi(\omega_i)$  na galeria, na fase que chamaremos de treino. Neste cenário, o usuário pode classificar parâmetros como bons ou ruins, ou simplesmente ignorá-los, como na figura 3.1. Cada parâmetro  $\omega_i$  selecionado pelo usuário vai dar origem a um novo par  $(\varphi(\omega_i), c_i)$  que será incluído no banco de dados.

Uma possível variação do método aceitaria classificações contínuas por parte do usuário ( $c_i \in [-1, 1]$ ), desde que se realizem adaptações na interface

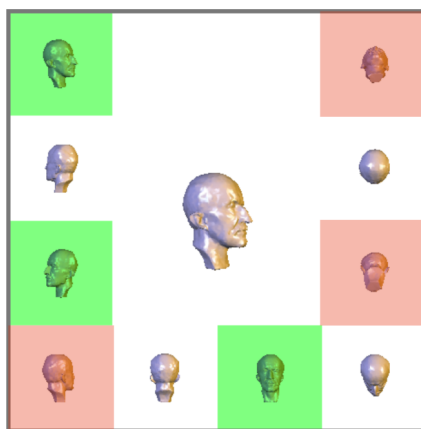


Figura 3.1: Classificação de parâmetros: o usuário pode classificar parâmetros como bons (verdes), ruins (vermelhos) ou ignorá-los.

das galerias que não prejudiquem a experiência do usuário. Além disso, a possibilidade de ignorar parâmetros da galeria simplifica o trabalho do usuário, que pode se focar em classificar apenas parâmetros que considere relevantes ou cuja classificação seja mais clara para ele, tornando desnecessária a classificação de todos os parâmetros da galeria.

A partir do momento em que o banco de dados é alimentado com dados provenientes das seleções do usuário, a máquina torna-se capaz de imitar suas preferências, realizando seleções e ordenações de parâmetros com o objetivo de ajudar o usuário. Essa aprendizagem pode ser analisada com o objetivo de destacar os descritores mais usados pelo usuário, como proposto no capítulo 4.

**Seleção automática de bons parâmetros** Este procedimento permite mostrar as escolhas da máquina ao usuário. As seleções automáticas podem ser executadas quando uma nova galeria é gerada, pelo método de amostragem inicial do domínio  $\Omega$ , ou como resultado de combinações das seleções do usuário. A máquina gera sua função classificadora  $\hat{f}$  usando o banco de dados atualizado, e seleciona parâmetros classificados como bons por  $\hat{f}$ . Uma possível variação deste procedimento é a seleção dos  $\nu$  melhores parâmetros.

**Correção de seleções automáticas** Devemos considerar que sempre existe a possibilidade de a máquina errar a preferência do usuário, principalmente em treinos iniciais, quando o banco de dados armazena apenas uma ou duas dezenas de parâmetros. Porém, a partir do momento em que a máquina seleciona parâmetros da galeria, o usuário pode corrigir essas seleções, que serão incorporadas ao banco de dados, e, conseqüentemente, a máquina será capaz de imitar as preferências do usuário com mais precisão.

Destacamos que esse aumento de precisão da função classificadora também é notável em dois cenários: quando o usuário treina galerias mais finas, ou seja, galerias que são criadas após o processo de classificação de parâmetros pelos usuários; e quando o usuário combina o treino de diferentes instâncias como, por exemplo, seleção de parâmetros de câmera de cenas diferentes.

**Ordenação dos parâmetros da galeria** Outra aplicação importante é a ordenação de parâmetros da galeria. Neste cenário, os parâmetros  $\omega_i$  da galeria são ordenados de acordo com suas classificações  $\hat{f}(\omega_i)$ , e reposicionados na interface, destacando sempre os melhores, com o objetivo de facilitar a identificação de bons parâmetros pelo usuário. Isto permite oferecer ao usuário boas opções primeiro, acelerando a sua escolha.

**Seleção automática do melhor parâmetro** O usuário pode usar este procedimento para obter automaticamente o parâmetro ótimo de acordo com a avaliação da máquina. A seleção automática do melhor parâmetro é realizada por refinamento sucessivo das galerias usando um algoritmo genético. Partindo de uma galeria inicial, a máquina seleciona sozinha os  $\nu$  melhores parâmetros e gera uma nova galeria, refinando a seleção anterior. A máquina continua selecionando os melhores parâmetros e gerando galerias mais finas, até convergir para uma galeria onde todos os parâmetros são semelhantes.

Usando a linguagem dos algoritmos genéticos, queremos otimizar uma função objetivo representada pela função classificadora  $\hat{f}$ . A população inicial do algoritmo é representada pelos parâmetros de uma galeria inicial, e as galerias seguintes representam evoluções da população inicial. Em cada geração, os melhores indivíduos (parâmetros) são selecionados para reprodução, onde são recombinados por pares e reproduzidos por combinações convexas. O apêndice B dá uma introdução aos algoritmos genéticos.

**Incorporação de regras objetivas** Finalmente, em diversos problemas, é necessário combinar regras subjetivas, como as representadas pela função classificadora, com regras objetivas. Por exemplo, no problema de posicionamento de câmera em uma cena tridimensional, a seleção das visões pelo usuário fornece um banco de dados para gerar regras subjetivas. Essas regras poderiam ser combinadas a princípios de cinematografia, por exemplo, para filtrar bons parâmetros. Esse tipo de filtragem por regras objetivas pode ser facilmente acoplada nas Galerias Inteligentes, removendo dos parâmetros selecionados os que não satisfazem a regra.

### 3.3

#### Componentes das Galerias Inteligentes

Nesta seção vamos descrever formalmente os procedimentos realizados pelas Galerias Inteligentes.

**Pré-processamento e geração da galeria inicial** Na fase de pré-processamento, a instância do problema é carregada, e as computações necessárias para tornar as próximas etapas eficientes são realizadas, como cálculo de propriedades, por exemplo.

Para gerar a galeria inicial, realizamos uma amostragem uniforme no espaço de parâmetros  $\Omega$ . O conjunto de amostras  $\Psi \subset \Omega$  é utilizado para gerar o conjunto de imagens  $\pi(\Psi)$  que irá representar os parâmetros na interface

da galeria. Além disso, nesta fase também é gerado um conjunto  $\Phi = \varphi(\Psi)$  contendo os vetores de descritores dos parâmetros.

Para permitir que os dados de treino sejam compatíveis com diferentes instâncias do problema, sempre devemos normalizar os valores dos descritores  $x_i = \varphi(\omega_i)$  de cada instância específica do problema. Usamos o conjunto de descritores  $\Phi$  que acabamos de criar para gerar uma função de normalização  $\chi: \mathbb{R}^p \mapsto \mathbb{R}^p$ , que deve ser aplicada, a partir de agora, em todos os descritores desta instância. Este procedimento de normalização é essencial para a robustez do método e será analisado mais detalhadamente no capítulo 4.

**Geração da função classificadora e classificação de parâmetros** Os parâmetros classificados pelo usuário em cada galeria treinada são incorporados ao banco de dados  $\Gamma$  em forma de pares  $(\chi(\varphi(\omega_i)), c_i)$ . O banco de dados atualizado dá origem a uma função de classificação  $\hat{f}$  sempre que seja necessário avaliar parâmetros. Em geral, essa etapa ocorre no momento de geração de uma nova galeria, quando um novo conjunto de parâmetros  $\Psi'$  é criado e precisa ser classificado.

**Reprodução e organização da nova galeria** O conjunto de parâmetros classificados pelo usuário é dado como entrada para um algoritmo genético que irá gerar um novo conjunto de parâmetros  $\Psi' \subset \Omega$  e dará origem a um novo conjunto de imagens  $\pi(\Psi')$  e um novo conjunto de descritores  $\Phi' = \chi(\varphi(\Psi'))$ .

Em seguida, este conjunto de descritores  $\Phi'$  é dado como entrada para a função classificadora com o objetivo de ordenar os parâmetros de  $\Psi'$  por qualidade, ou seja, do melhor para o pior, de acordo com os valores de  $\hat{f}(\Phi')$ . Após esta ordenação, as imagens  $\pi(\Psi')$  são organizadas na galeria, privilegiando-se os melhores parâmetros. Por simplicidade, organizamos estes parâmetros do melhor para o pior no sentido anti-horário, começando do canto superior esquerdo, como ilustra a figura 3.2.

**Seleção automática** Em qualquer momento após alimentar o banco de dados pela primeira vez, o usuário pode solicitar ajuda das Galerias Inteligentes para selecionar bons parâmetros na galeria atual. Descrevemos dois métodos para realizar esta seleção.

No primeiro método, a galeria seleciona na interface todos os parâmetros  $\omega \in \Psi$ , tais que  $\hat{f}(\chi(\varphi(\omega))) > 0$ . Na segunda estratégia, selecionamos os  $\nu$  melhores parâmetros, de acordo com as avaliações  $\hat{f}(\Phi)$ . Note que se

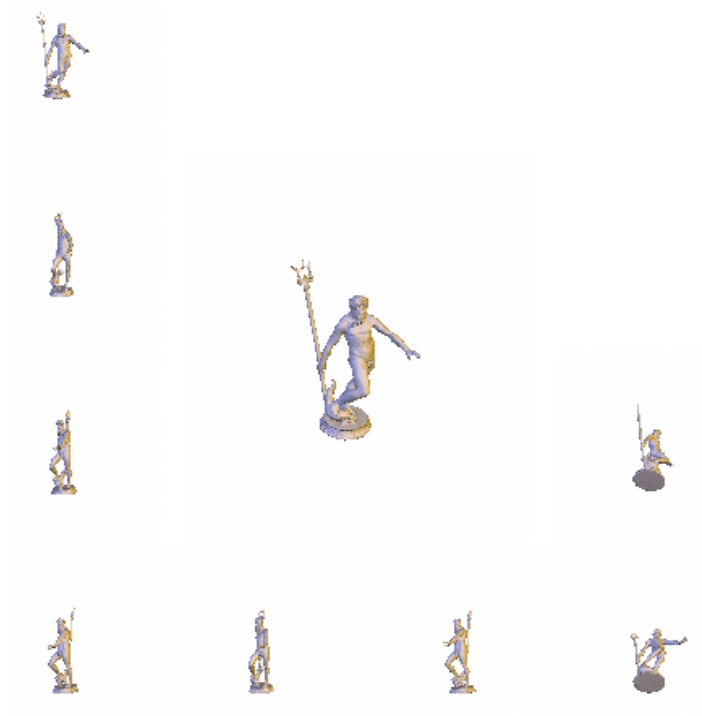


Figura 3.2: Galeria ordenada pela função classificadora do melhor para o pior parâmetro, começando do canto superior esquerdo no sentido anti-horário.

$\lambda = \left| \left\{ \omega \in \Psi \mid \hat{f}(\chi(\varphi(\omega_i))) > 0 \right\} \right| < \nu$ , então apenas  $\lambda$  parâmetros serão selecionados.