4

Aprendizagem de Máquina para o Extended Hyperlink Induced Topic Search

O algoritmo *XHITS* provê como resultado o vetor r_σ com os pesos por categorias para todas as páginas, calculando o autovetor associado ao maior autovalor da matriz de influência M_σ.

Como visto no capítulo 3, as matrizes F e B são responsáveis por fazer um ajuste fino na estrutura da matriz M_σ. Conforme são modificados os valores dos elementos das matrizes F e B, o autovalor da matriz M_σ e o seu autovetor associado se alteram, assim modificando a classificação final das páginas.

Desta forma, o algoritmo *XHITS* possui um conjunto de parâmetros independentes da consulta σ, representados pelas matrizes F e B que devem ser definidas para o cálculo da classificação final das páginas. Para a definição desses valores, é apresentada neste capítulo uma abordagem de aprendizado de máquina que utiliza, dentre outras, as técnicas de Decomposição em Valores Singulares(SVD), denominada de Aprendizagem com Múltiplas Categorias Latentes (AMCL).

A próxima seção apresenta todos os conceitos da AMCL. Na seção 4.2, a AMCL é formalizada em um algoritmo cujo funcionamento é explicado detalhadamente e, por fim, na seção 4.3 é provada uma propriedade importante do mecanismo de aprendizagem.

4.1

Aprendizagem com Múltiplas Categorias Latentes

A propriedade fundamental da aprendizagem de máquina é a habilidade do componente computacional conseguir aprender com as informações disponíveis no seu ambiente e melhorar o seu desempenho (HAYKIN, 1998).

Em um modelo simples de aprendizagem de máquina, como mostrado na

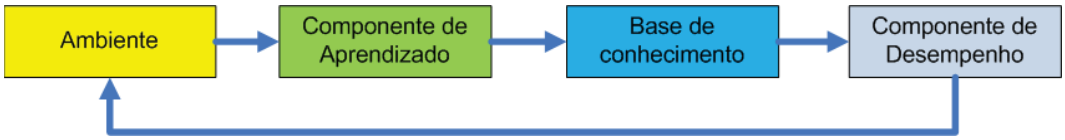

Figura 4.1: Modelo básico de aprendizagem de máquina.

figura 4.1, o *ambiente* em que está inserido o *componente de aprendizagem* lhe fornece alguma informação. A partir dessa informação, o componente efetua melhorias na sua *base de conhecimento*, e um *componente de desempenho* utiliza a mesma para avaliar o componente de aprendizagem.

O tipo de informação fornecida pelo ambiente à máquina geralmente é imperfeito e, por consequência, o componente de aprendizagem não sabe *a priori* como preencher os detalhes ausentes ou ignorar os desnecessários. Então, o componente de aprendizagem opera por suposição e recebe uma avaliação do componente de desempenho, que permite que o mesmo evolua suas hipóteses e as revise se for necessário.

A aprendizagem de máquina pode envolver dois tipos de processamento de informação: indutivo e dedutivo.

No processamento indutivo da informação, padrões e regras são determinados por conjuntos de dados e experiência em torno dos mesmos.

No processamento da informação dedutivo, regras gerais são usadas para determinar fatos específicos.

Assim, se a aprendizagem for feita por similaridade esse usa indução, enquanto a prova de um teorema é uma dedução a partir de axiomas conhecidos e outros teoremas.

Uma das formas de se fazer o aprendizado indutivo, de acordo com o modelo apresentado acima, é através da aprendizagem por correção de erro.

Neste caso, o componente de desempenho recebe, além da informação constante na base de conhecimento, a informação referencial a que se deseja igualar.

O componente de desempenho efetua a sua avaliação, calculando o erro existente entre a informação produzida e a referencial. O erro então é retroalimentado e atua como um mecanismo de controle, que proporciona sucessivos ajustes para que a informação gerada e a referencial se igualem.

Dentro desse contexto, a estrutura da AMCL possui quatro componentes

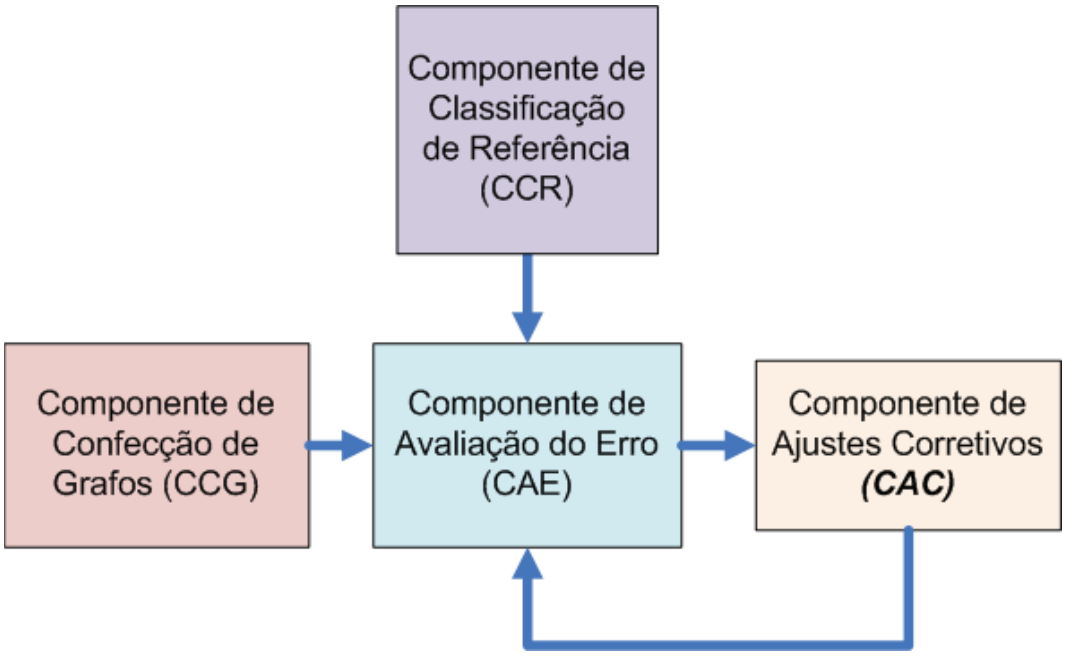

Figura 4.2: Modelo resumido do aprendizado de máquina baseado em correção de erro.

participantes (figura 4.2), quais sejam:

- Componente de Confecção de Grafos (CCG);
- Componente de Classificação de Referência (CCR);
- Componente de Ajustes Corretivos (CAC);
- e, Componente de Avaliação do Erro (CAE).

O Componente de Confecção dos Grafos (CCG) gera os grafos induzidos G_σ, representado pela matriz de adjacência A_{sigma}(cap. 2 e 3), para cada consulta σ que vai ser utilizada no processo de aprendizagem. Após gerar o conjunto de grafos, $L = \{A_\sigma | \sigma = 1, \ldots, q\}$, esse é repassado para o Componente de Avaliação do Erro (CAE).

O Componente de Classificação de Referência (CCR) gera um conjunto de classificações de referência das páginas *Web* para cada uma das consultas σ de treinamento, $O = \{o_{\sigma j} | \sigma = 1, \ldots, q \text{ and } j = 1, \cdots, p\}$, em que j é o identificador da página e $o_{\sigma j}$ é a classificação correta da página j para a consulta σ. Este conjunto de classificações é utilizado com o objetivo que se deseja alcançar no treinamento e é repassado para o CAE.

Um dos núcleos da Aprendizagem com Múltiplas Categorias Latentes é o CAE. Nessa componente o erro existente entre o conjunto L, contendo os grafos de treinamento, e o O, contendo as classificações de referência, é

calculado através da função erro (E), e repassado para o Componente de Ajustes Corretivos (CAC).

Para compreender o calculo do erro pela E, algumas considerações têm que ser feitas.

Primeiro, em sua concepção, o algoritmo *XHITS* provê como resultado o vetor r_σ com os pesos por categorias para todas as páginas, calculando o autovetor associado ao maior autovalor da matriz de influência M_σ.

Assim, analisando a equação 3-10, a organização interna do vetor r_σ é um pouco peculiar. Para cada uma das k categorias, o vetor guarda os pesos das p páginas em posições consecutivas. Ou seja, os pesos das p páginas da primeira categoria estão compreendidos entre a primeira e a p-ésima posição do vetor, os da segunda entre a $p+1$-ésima e a $2p$-ésima posições e assim sucessivamente.

Generalizando, para encontrar o peso da página j na categoria k basta calcular $l = j + (k-1)p$ para encontrar a posição, onde l é o índice de uma posição do vetor, $r_{\sigma l}$.

A primeira parcela da função E calcula o erro entre $o_{\sigma j}$ e $r_{\sigma l}$, através do erro quadrático médio. Como $o_{\sigma j}$ e $r_{\sigma l}$ podem possuir escalas de valores diferentes para os pesos das páginas, reescalamos os dois vetores utilizando a função sigmoide.

Logo, podemos definir a primeira parcela da função E como

$$\left[\frac{1}{qp}\sum_{\sigma=1}^{q}\sum_{j=1}^{p}\left(\frac{1}{1+e^{-o_{\sigma j}}}-\frac{1}{1+e^{-r_{\sigma j}}}\right)^2\right] \tag{4-1}$$

Ainda, os pesos contidos em r_σ que são comparados na equação 4-1 são os relativos à primeira categoria, como pode ser visto nos limites de j, $1 \leq j \leq p$. Ademais, a primeira categoria do *XHITS* corresponde à autoridade no *HITS*.

A segunda consideração consiste na essência do vetor r_σ. Esse representa o autovetor associado ao autovalor dominante da matriz de influência M_σ. Então, a segunda parcela da função E foi concebida de modo que, conforme vai sendo minimizada, coloca o vetor r_σ na direção do autovetor associado ao maior autovalor.

Para tal, algumas técnicas de *SVD* foram utilizadas, o que resultou na seguinte formulação para a segunda parcela

$$\left[\frac{1}{(kqp)^2}\sum_{\sigma=1}^{q}\sum_{i=1}^{kp}\sum_{j=1}^{kp}(m_{\sigma ij}-b_{\sigma i}.r_{\sigma j})^2\right] \tag{4-2}$$

A prova e o processo de formulação da equação 4-2 se encontram na seção 4.3.

Por fim, unindo as duas parcelas da função do erro, obtemos

$$E=\frac{1}{q}\sum_{\sigma=1}^{q}\left\{\begin{array}{c}\left[\frac{1}{p}\sum_{j=1}^{p}(\frac{1}{1+e^{o_{\sigma j}}}-\frac{1}{1+e^{r_{\sigma j}}})^2\right]+\\ \\ +\left[\frac{1}{(kp)^2}\sum_{i=1}^{kp}\sum_{j=1}^{kp}(m_{\sigma ij}-b_{\sigma i}.r_{\sigma j})^2\right]\end{array}\right\} \tag{4-3}$$

Analisando a equação 4-3, é notório que a mesma possui variáveis que a função E não conhece o valor correto *a priori*, quais sejam, F,r_σ e b_σ. A dependência em relação a r_σ e b_σ está explícita na equação e, em relação a F, a dependência surge como decorrência da definição da matriz M_σ, conforme pode ser visto na equação 3-11.

Como todo processo de aprendizado, essas variáveis recebem um valor inicial que ao longo do aprendizado vão sendo ajustados melhorando a performance do sistema. Esse papel é desempenhado pelo Componente de Ajustes Corretivos (CAC).

O CAC é o segundo núcleo mais importante da AMCL. A sua função principal consiste em atualizar as variáveis de entrada do CAE, (F,r_σ e b_σ) de forma a minimizar o erro calculado pela função E. Assim, precisamos resolver um problema de otimização que consiste em minimizar a função E em relação a F,r_σ e b_σ.

A condição necessária para atingir o ponto ótimo é $\nabla E = 0$, onde ∇ é o operador gradiente e ∇E é o vetor gradiente da função erro

$$\nabla E=\left[\frac{\partial E}{\partial F},\frac{\partial E}{\partial r_\sigma},\frac{\partial E}{\partial b_\sigma},\right]^T \tag{4-4}$$

Dentre as diversas abordagens de otimização (HSIEH; DHILLON, 2011; LUKSAN; MATONOHA; VLCEK, 2009; GOLUB; LOAN, 1996), a mais presente foi a de utilizar o algoritmo do gradiente descendente para minimizar a função E. Esse pertence a uma classe de algoritmos de otimização que é baseada na ideia de uma busca iterativa.

Tal ideia, consiste em inicializar os parâmetros de entrada com valores aleatórios, por exemplo, e gerar uma sequência de novos valores de tal forma que a função a ser minimizada seja reduzida a cada iteração do algoritmo.

Assim, os sucessivos ajustes aplicados aos parâmetros de entrada da função a ser minimizada são na direção descendente, ou seja, em direção oposta ao vetor gradiente.

Desta forma, para uma função $f(\vec{x})$, onde $\vec{x}$ representa um vetor de parâmetros o algoritmo é formalmente descrito como

$$x^{s+1} = x^s - \mu \nabla f(\vec{(x)}) \tag{4-5}$$

onde μ é uma constante positiva denominada de taxa de aprendizado.

Adaptando a definição acima para o E, temos o seguinte conjunto de equações

$$b_\sigma^{s+1} \leftarrow b_\sigma^s - \mu \frac{\partial E}{\partial b_\sigma} \tag{4-6}$$

$$r_\sigma^{s+1} \leftarrow r_\sigma^s - \mu \frac{\partial E}{\partial r_\sigma} \tag{4-7}$$

$$F^{s+1} \leftarrow F^s - \mu \frac{\partial E}{\partial F} \tag{4-8}$$

onde μ é a taxa de aprendizado e s representa a iteração do algoritmo.

O próximo passo consiste em calcular as derivadas parciais do erro E em relação a r_σ, b_σ e F, ou seja

$$\frac{\partial E}{\partial r_\sigma} = \frac{1}{q} \sum_{\sigma=1}^{q} \left\{ \begin{array}{l} \left[\frac{2}{p} \sum_{j=1}^{p} \left(\frac{1}{1+e^{o_{\sigma j}}} - \frac{1}{1+e^{r_{\sigma j}}} \right) \left(\frac{e^{r_{\sigma j}}}{(1+e^{r_{\sigma j}})^2} \right) \right] - \\ \\ - \left[\frac{2}{(kp)^2} \sum_{i=1}^{kp} \sum_{j=1}^{kp} (m_{\sigma,i,j} - b_{\sigma i}.r_{\sigma j}).b_{\sigma i} \right] \end{array} \right\} \tag{4-9}$$

$$\frac{\partial E}{\partial b_\sigma} = -\frac{1}{q} \sum_{\sigma=1}^{q} \left[\frac{2}{(kp)^2} \sum_{i=1}^{kp} \sum_{j=1}^{kp} (m_{\sigma ij} - b_{\sigma i}.r_{\sigma j}).r_{\sigma i} \right] \tag{4-10}$$

$$\frac{\partial E}{\partial F} = \frac{1}{q}\sum_{\sigma=1}^{q}\left[\frac{2}{(kp)^2}\sum_{i=1}^{kp}\sum_{j=1}^{kp}(m_{\sigma ij} - b_{\sigma i}.r_{\sigma j}).\frac{\partial m_{\sigma ij}}{\partial F}\right] \quad (4\text{-}11)$$

Depois que os novos valores de F,r_σ e b_σ são gerados, são enviados para o CAE, iniciando um novo ciclo na AMCL.

Finalmente, definidas as estruturas do aprendizado, na próxima seção são abordados o pseudocódigo do algoritmo de aprendizagem, bem como a sua descrição.

4.2
Algoritmo de Aprendizagem com Múltiplas Categorias Latentes

O algoritmo de Aprendizagem com Múltiplas Categorias Latentes (AMCL) recebe como parâmetros de entrada o conjunto L contendo os grafos de treinamento, o conjunto O contendo as classificações de referência, as consultas σ, a precisão do erro de treinamento (PET) – fator de corte da função do erro do treinamento E – e o número máximo de iterações do algoritmo de treinamento ($MaxIt$). Como resultado, o algoritmo retorna a matriz F definida, de forma a minimizar E sem exceder os parâmetros de corte PET e $MaxIt$.

Primeiro, na linha 2, inicializamos o contador de iterações do algoritmo com 0.

Em seguida, entre as linhas 3 e 6, temos o primeiro laço do algoritmo que inicializa os vetores r_σ e b_σ para cada uma das consultas σ. Como é mencionado na seção 4.1, é necessário que essas variáveis recebam um valor inicial para que ao longo do aprendizado venham a ser ajustadas melhorando a performance do sistema.

Em seguida, a matriz F é inicializada com valores aleatórios.

Para cada iteração do laço compreendido entre as linhas 8 e 17, o algoritmo calcula a derivada parcial da função E em relação a r_σ e b_σ, com as respectivas equações 4-9 e 4-10, e calcula os próximos valores de r_σ e b_σ, de acordo com as equações 4-7 e 4-6 respectivamente, para cada uma das consultas σ, conforme o laço compreendido entre as linhas 9 e 12.

Após, na linha 13, calcula o valor da derivada parcial de função E em

Algoritmo 4.1: Algoritmo de Aprendizagem com Múltiplas Categorias Latentes

Entrada:
L:Conjunto contendo os grafos de treinamento
O:Conjunto contendo as classificações de referência para cada consulta σ
σ:As consultas
PET:Precisão do Erro do Treino
$MaxIt$:Número máximo de iterações
Saída: A matriz F definida

```
1  início
2      iter=0;
3      para cada σ faça
4          r_σ=Aleatório();
5          b_σ=Aleatório();
6      fim
7      F=Aleatório();
8      repita
9          para cada consulta σ faça
10             Calcular (∂E/∂r_σ) e (∂E/∂b_σ);
11             Calcular (b_σ^{s+1}) and (r_σ^{s+1});
12         fim
13         Calcular (∂E/∂F);
14         Calcular (F^{s+1});
15         Calcular E;
16         iter++;
17     até (E < PET) ou (iter > MaxIt);
18 fim
```

relação a F conforme a equação 4-11.

A partir da derivada parcial calculada, na linha 14, calcula o próximo valor da matriz F conforme a equação 4-8.

Finalmente, o algoritmo calcula a equação E conforme a equação 4-3 e verifica se já se encontra abaixo do PET e se o número máximo de iterações não foi excedido, voltando para a linha 8, repetindo o laço.

4.3 Decomposição aplicada a Aprendizagem com Múltiplas Categorias Latentes

O objetivo principal da presente seção é provar que a segunda parcela da função E (equação 4-2), ao ser minimizada, iguala o vetor r_σ do autovetor

associado ao maior autovalor absoluto da matriz de influência M_σ.

Iniciando a prova a partir da própria matriz $M_\sigma = [m_{ij}]$, que é simétrica e possui todos os seus valores reais e maiores ou iguais a zero, podemos decompô-la em seus autovetores ortogonais e seus autovalores associados através do teorema 4.3.1 de decomposição de *Schur* (GOLUB; LOAN, 1996), ou seja

$$M_\sigma = \sum_{i=1}^{n} \lambda_i q_i q_i^T \tag{4-12}$$

Teorema 4.3.1 ***Decomposição Simétrica de*** **Schur**

Se $X \in \Re^{n \times n}$ ***é simétrica, então existe uma matriz ortogonal real*** Q ***tal que***

$Q^T X Q = \Lambda = diag(\lambda_1, \ldots, \lambda_n)$ ***onde*** λ_i ***são os autovalores de*** X ***e*** q_i ***são os autovetores associados.***

Subtraindo da matriz M_σ a primeira parcela do somatório, podemos adaptar a equação 4-12 para

$$M_\sigma - \lambda_1 q_1 q_1^T = \sum_{i=2}^{n} \lambda_i q_i q_i^T \tag{4-13}$$

e calculando $(M_\sigma - \lambda_1 q_1 q_1^T)^2$ tem-se

$$(M_\sigma - \lambda_1 q_1 q_1^T)^2 = \left(\sum_{i=2}^{n} \lambda_i q_i q_i^T \right)^2 \tag{4-14}$$

e

$$(M_\sigma - \lambda_1 q_1 q_1^T)^2 = (\lambda_2 q_2 q_2^T + \ldots + \lambda_n q_n q_n^T)(\lambda_2 q_2 q_2^T + \ldots + \lambda_n q_n q_n^T). \tag{4-15}$$

Observando a forma expandida do somatório (equação 4-15) da equação 4-13 podemos verificar que na multiplicação termo a termo, duas possibilidades ocorrem, quais sejam:

– para $i \neq j$,

$$(\lambda_i q_i q_i^T . \lambda_j q_j q_j^T) = (\lambda_i \lambda_j q_i q_i^T q_j q_j^T) = (\lambda_i \lambda_j q_i 0 q_j^T) = 0;$$

– e, para $i = j$,

$$(\lambda_i q_i q_i^T)^2 = (\lambda_i^2 q_i q_i^T q_i q_i^T) = (\lambda_i^2 q_i 1 q_i^T) = (\lambda_i^2 q_i q_i^T).$$

Aplicando as observações acima na equação 4-15, então

$$(M_\sigma - \lambda_1 q_1 q_1^T)^2 = \sum_{i=2}^{n} \lambda_i^2 q_i q_i^T \tag{4-16}$$

Como M_σ e $\lambda_1 q_1 q_1^T$ são matrizes simétricas, então $(M_\sigma - \lambda_1 q_1 q_1^T)$ também é simétrica.

Assim, podemos concluir que

$$(M_\sigma - \lambda_1 q_1 q_1^T)^2 = (M_\sigma - \lambda_1 q_1 q_1^T)^T (M_\sigma - \lambda_1 q_1 q_1^T) \tag{4-17}$$

Calculando o traço da equação 4-17, então

$$tr((M_\sigma - \lambda_1 q_1 q_1^T)^2) = tr((M_\sigma - \lambda_1 q_1 q_1^T)^T (M_\sigma - \lambda_1 q_1 q_1^T)) = \tag{4-18}$$

$$= tr(\sum_{i=2}^{n} \lambda_i^2 q_i q_i^T) = tr(\lambda_2^2 q_2 q_2^T) + \ldots + tr(\lambda_n^2 q_n q_n^T) = \tag{4-19}$$

$$= \lambda_2^2 + \ldots + \lambda_n^2 = \lambda_1^2 + \ldots + \lambda_n^2 - \lambda_1^2 \tag{4-20}$$

Adicionalmente, a norma de *Frobenius*, que é definida como

$$\|X\|_F^2 = \sum_{i=1}^{r} \sum_{j=1}^{n} |x_{ij}|^2 = tr(MM^T) = \sum_{i=1}^{min\{r,n\}} \rho_i^2 \tag{4-21}$$

aplicada nas equações 4-18 e 4-20, implica em

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (m_{\sigma ij} - \lambda_1 q_{1i} q_{1i}^T)^2 = tr((M - \lambda_1 q_1 q_1^T)(M - \lambda_1 q_1 q_1^T)^T) \tag{4-22}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (m_{\sigma ij} - \lambda_1 q_{1i} q_{1i}^T)^2 = \lambda_2^2 + \ldots + \lambda_n^2 = \lambda_1^2 + \ldots + \lambda_n^2 - \lambda_1^2. \tag{4-23}$$

Ao minimizar a equação 4-23, ou seja

$$min\{\sum_{i=1}^{n}\sum_{j=1}^{n}(m_{\sigma ij} - \lambda_1 q_{1i} q_{1i}^T)^2\} = min\{\lambda_1^2 + \ldots + \lambda_n^2 - \lambda_1^2\} \qquad (4\text{-}24)$$

podemos afirmar que o autovalor λ_1 da matriz M_σ é o que possui o maior valor absoluto, pois qualquer outro menor não minimizaria a diferença do lado direito da igualdade, e por consequência, não minimizaria a equação.

Por fim, substituímos q_{1j} por $r_{\sigma j}$ e $\lambda_1 q_{1i}$ por $b_{\sigma i}$ na equação 4-24, assim completando a prova.