

---

## Capítulo 2

# Colaboração baseada em bairros Filtragem

---

“Quando um vizinho ajuda o outro, fortalecemos nossas comunidades.” –  
Jennifer Pahlka

### 2.1 Introdução

---

Algoritmos de filtragem colaborativa baseados em vizinhança, também conhecidos como algoritmos baseados em memória, estão entre os primeiros algoritmos desenvolvidos para filtragem colaborativa. Esses algoritmos se baseiam no fato de que usuários semelhantes exibem padrões semelhantes de comportamento de avaliação e itens semelhantes recebem avaliações semelhantes. Existem dois tipos principais de algoritmos baseados em vizinhança:

1. Filtragem colaborativa baseada no usuário: neste caso, as classificações fornecidas por usuários semelhantes a um usuário-alvo A são usadas para fazer recomendações para A. As classificações previstas de A são calculadas como os valores médios ponderados dessas classificações de “grupo de pares” para cada item.
2. Filtragem colaborativa baseada em itens: para fazer recomendações para o item alvo B, o primeiro passo é determinar um conjunto S de itens que são mais semelhantes ao item B.  
Em seguida, para prever a classificação de qualquer usuário A para o item B, são determinadas as classificações no conjunto S, especificadas por A. A média ponderada dessas classificações é usada para calcular a classificação prevista do usuário A para o item B.

Uma distinção importante entre a filtragem colaborativa baseada no usuário e os algoritmos de filtragem colaborativa baseados em itens é que as classificações no primeiro caso são previstas usando as classificações de usuários vizinhos, enquanto as classificações no último caso são previstas usando

as avaliações do próprio usuário sobre itens vizinhos (ou seja, intimamente relacionados). No primeiro caso, as vizinhanças são definidas por similaridades entre os usuários (linhas da matriz de avaliações), enquanto no segundo caso, as vizinhanças são definidas por similaridades entre os itens (colunas da matriz de avaliações). Assim, os dois métodos compartilham uma relação complementar. No entanto, existem diferenças consideráveis nos tipos de recomendações obtidas com esses dois métodos.

Para fins de discussão subsequente, assumimos que a matriz de classificações de itens por usuário é uma matriz incompleta  $m \times n R = [ruj]$  contendo  $m$  usuários e  $n$  itens. Assume-se que apenas um pequeno subconjunto da matriz de classificações é especificado ou observado. Como todos os outros algoritmos de filtragem colaborativa, os algoritmos de filtragem colaborativa baseados em vizinhança podem ser formulados de duas maneiras:

1. Predição do valor da classificação de uma combinação usuário-item: Esta é a formulação mais simples e primitiva de um sistema de recomendação. Neste caso, prevê-se a classificação ausente  $ruj$  do usuário  $u$  para o item  $j$ .
2. Determinando os  $k$  principais itens ou os  $k$  principais usuários: Na maioria dos cenários práticos, o comerciante não está necessariamente procurando por valores de classificação específicos de combinações usuário-item. Em vez disso, é mais interessante aprender os  $k$  principais itens mais relevantes para um usuário específico, ou os  $k$  principais usuários mais relevantes para um item específico. O problema de determinar os  $k$  principais itens é mais comum do que o de encontrar os  $k$  principais usuários. Isso ocorre porque a primeira formulação é usada para apresentar listas de itens recomendados aos usuários em cenários centrados na Web. Em algoritmos de recomendação tradicionais, o "problema dos  $k$  principais" quase sempre se refere ao processo de encontrar os  $k$  principais itens, em vez dos  $k$  principais usuários.

No entanto, a última formulação também é útil para o comerciante porque pode ser usada para determinar os melhores usuários a serem segmentados com esforços de marketing.

Os dois problemas mencionados estão intimamente relacionados. Por exemplo, para determinar os  $k$  principais itens para um determinado usuário, é possível prever as avaliações de cada item para esse usuário.

Os  $k$  itens mais importantes podem ser selecionados com base na classificação prevista. Para aumentar a eficiência, métodos baseados em vizinhança pré-calculam alguns dos dados necessários para a previsão em uma fase offline. Esses dados pré-calculados podem ser usados para realizar a classificação de forma mais eficiente.

Este capítulo discutirá vários métodos baseados em vizinhança. Estudaremos o impacto de algumas propriedades das matrizes de classificação em algoritmos de filtragem colaborativa. Além disso, estudaremos o impacto da matriz de classificação na eficácia e eficiência das recomendações.

Discutiremos o uso de agrupamentos e representações baseadas em grafos para implementar métodos baseados em vizinhança. Também discutiremos as conexões entre métodos de vizinhança e técnicas de modelagem de regressão. Métodos de regressão fornecem uma estrutura de otimização para métodos baseados em vizinhança. Em particular, o método baseado em vizinhança pode ser demonstrado como uma aproximação heurística de um modelo de regressão de mínimos quadrados [72]. Essa equivalência aproximada será mostrada na seção 2.6. Tal estrutura de otimização também abre caminho para a integração de métodos de vizinhança com outros modelos de otimização, como modelos de fatores latentes. A abordagem integrada é discutida em detalhes na seção 3.7 do Capítulo 3.

Este capítulo está organizado da seguinte forma. A Seção 2.2 discute uma série de propriedades-chave das matrizes de classificação. A Seção 2.3 discute os principais algoritmos para algoritmos de filtragem colaborativa baseados em vizinhança. A Seção 2.4 discute como algoritmos baseados em vizinhança podem ser tornados mais rápidos com o uso de métodos de agrupamento. A Seção 2.5 discute o uso de métodos de redução de dimensionalidade para aprimorar algoritmos de filtragem colaborativa baseados em vizinhança.

Uma visão de modelagem de otimização de métodos baseados em vizinhança é discutida na seção 2.6.

Uma abordagem de regressão linear é utilizada para simular o modelo de vizinhança dentro de uma estrutura de aprendizado e otimização. A Seção 2.7 discute como representações baseadas em grafos podem ser usadas para aliviar o problema de esparsidade em métodos de vizinhança. O resumo é fornecido na Seção 2.8.

## 2.2 Principais propriedades das matrizes de classificação

---

Conforme discutido anteriormente, assumimos que a matriz de classificações é denotada por  $R$  e é uma matriz  $m \times n$  contendo  $m$  usuários e  $n$  itens. Portanto, a classificação do usuário  $u$  para o item  $j$  é denotada por  $r_{uj}$ . Apenas um pequeno subconjunto das entradas na matriz de classificações é normalmente especificado. As entradas especificadas da matriz são chamadas de dados de treinamento, enquanto as entradas não especificadas da matriz são chamadas de dados de teste. Essa definição tem um análogo direto em algoritmos de classificação, regressão e aprendizado semissupervisionado [22]. Nesse caso, todas as entradas não especificadas pertencem a uma coluna especial, que é conhecida como variável de classe ou variável dependente. Portanto, o problema de recomendação pode ser visto como uma generalização do problema de classificação e regressão.

As classificações podem ser definidas de várias maneiras, dependendo da aplicação em questão:

1. Classificações contínuas: As classificações são especificadas em uma escala contínua, correspondendo ao nível de gosto ou desgosto do item em questão. Um exemplo desse sistema é o mecanismo de recomendação de piadas Jester [228, 689], no qual as classificações podem assumir qualquer valor entre -10 e 10. A desvantagem dessa abordagem é que ela cria um fardo para o usuário, que precisa pensar em um valor real entre um número infinito de possibilidades.  
Portanto, tal abordagem é relativamente rara.
2. Classificações baseadas em intervalos: em classificações baseadas em intervalos, as classificações geralmente são extraídas de uma escala de 5 ou 7 pontos, embora escalas de 10 e 20 pontos também sejam possíveis. Exemplos dessas classificações podem ser valores inteiros numéricos de 1 a 5, de -2 a 2 ou de 1 a 7. Uma suposição importante é que os valores numéricos definem explicitamente as distâncias entre as classificações, e os valores de classificação são normalmente equidistantes.
3. Classificações ordinais: As classificações ordinais são muito semelhantes às classificações baseadas em intervalos, exceto que podem ser utilizados valores categóricos ordenados. Exemplos desses valores categóricos ordenados podem ser respostas como "Discordo Totalmente", "Discordo", "Neutro", "Concordo" e "Concordo Totalmente". Uma diferença importante em relação às classificações baseadas em intervalos é que não se presume que a diferença entre qualquer par de valores de classificação adjacentes seja a mesma. No entanto, na prática, essa diferença é apenas teórica, porque esses diferentes valores categóricos ordenados são frequentemente atribuídos a valores de utilidade igualmente espaçados. Por exemplo, pode-se atribuir à resposta "Discordo Totalmente" um valor de classificação de 1, e à resposta "Concordo Totalmente" um valor de classificação de 5. Nesses casos, as classificações ordinais são quase equivalentes às classificações baseadas em intervalos. Geralmente, os números de respostas positivas e negativas são igualmente equilibrados para evitar viés. Nos casos em que um número par de respostas é usado, a opção "Neutro" não está presente. Essa abordagem é chamada de método de escolha forçada porque a opção neutra não está presente.
4. Classificações binárias: No caso de classificações binárias, apenas duas opções estão presentes, correspondendo a respostas positivas ou negativas. As classificações binárias podem ser consideradas um caso especial de classificações baseadas em intervalos e ordinais. Por exemplo, a estação de rádio online Pandora oferece aos usuários a possibilidade de curtir ou não uma determinada faixa musical.

As classificações binárias são um exemplo de caso em que uma escolha forçada é imposta ao usuário. Nos casos em que o usuário é neutro, ele geralmente não especifica nenhuma classificação.

5. Classificações unárias: Esses sistemas permitem que o usuário especifique uma preferência positiva por um item, mas não há um mecanismo para especificar uma preferência negativa. Isso costuma acontecer em muitas situações do mundo real, como o uso do botão "curtir" no Facebook. Mais frequentemente, as classificações unárias são derivadas das ações do cliente. Por exemplo, o ato de um cliente comprar um item pode ser considerado um voto positivo para o item. Por outro lado, se o cliente não comprou o item, isso não indica necessariamente uma antipatia pelo item. As classificações unárias são especiais porque simplificam o desenvolvimento de modelos especializados nesses cenários.

Vale ressaltar que a derivação indireta de classificações unárias a partir de ações do cliente também é chamada de feedback implícito, porque o cliente não fornece feedback explicitamente.

Em vez disso, o feedback é inferido de forma implícita por meio das ações do cliente. Esses tipos de "classificações" costumam ser mais fáceis de obter, pois os usuários são muito mais propensos a interagir com itens em um site online do que a classificá-los explicitamente. O contexto do feedback implícito (ou seja, classificações unárias) é inherentemente diferente, pois pode ser considerado o análogo de complementação de matriz do problema de aprendizagem positiva não rotulada (PU) na modelagem de classificação e regressão.

A distribuição de classificações entre itens frequentemente satisfaz uma propriedade em cenários do mundo real, conhecida como propriedade da cauda longa. De acordo com essa propriedade, apenas uma pequena fração dos itens é classificada com frequência. Esses itens são chamados de itens populares. A grande maioria dos itens é classificada raramente. Isso resulta em uma distribuição altamente assimétrica das classificações subjacentes.

Um exemplo de distribuição assimétrica de classificações é ilustrado na Figura 2.1.

O eixo X mostra o índice do item em ordem decrescente de frequência, e o eixo Y mostra a frequência com que o item foi avaliado. É evidente que a maioria dos itens é avaliada apenas um pequeno número de vezes. Essa distribuição de classificação tem implicações importantes para o processo de recomendação:

1. Em muitos casos, os itens de alta frequência tendem a ser itens relativamente competitivos, com pouco lucro para o comerciante. Por outro lado, os itens de menor frequência apresentam margens de lucro maiores. Nesses casos, pode ser vantajoso para o comerciante recomendar itens de menor frequência. De fato, análises sugerem [49] que muitas empresas, como a Amazon.com, obtêm a maior parte de seus lucros vendendo itens na cauda longa.
2. Devido à raridade das classificações observadas na cauda longa, geralmente é mais difícil fornecer previsões de classificação robustas na cauda longa. De fato, muitos algoritmos de recomendação tendem a sugerir itens populares em vez de itens pouco frequentes [173]. Esse fenômeno também tem um impacto negativo na diversidade, e os usuários podem ficar entediados ao receber o mesmo conjunto de recomendações de itens populares.
3. A distribuição de cauda longa implica que os itens frequentemente avaliados pelos usuários são em menor número. Esse fato tem implicações importantes para algoritmos de filtragem colaborativa baseados em vizinhança, pois as vizinhanças são frequentemente definidas com base nesses itens frequentemente avaliados. Em muitos casos, as avaliações desses itens de alta frequência não são representativas dos itens de baixa frequência devido às diferenças inerentes nos padrões de avaliação das duas classes de itens. Como resultado, o processo de predição pode gerar resultados enganosos. Como discutiremos na seção 7.6 do Capítulo 7, esse fenômeno também pode causar avaliações enganosas de algoritmos de recomendação.

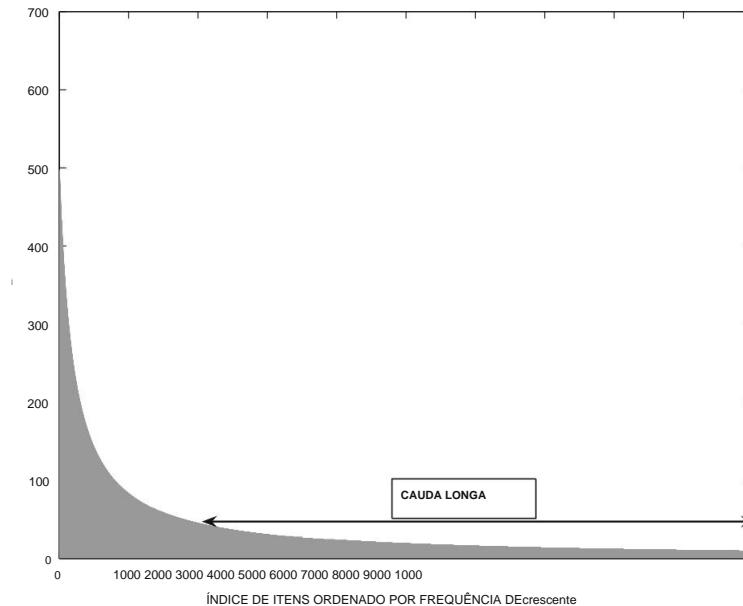


Figura 2.1: A longa cauda das frequências de classificação

Características importantes das classificações, como a dispersão e a cauda longa, precisam ser levadas em consideração durante o processo de recomendação. Ao ajustar os algoritmos de recomendação para levar em conta essas propriedades do mundo real, é possível obter previsões mais significativas [173, 463, 648].

## 2.3 Previsão de classificações com base em vizinhança Métodos

---

A ideia básica dos métodos baseados em vizinhança é usar a similaridade usuário-usuário ou a similaridade item-item para fazer recomendações a partir de uma matriz de classificação. O conceito de vizinhança implica que precisamos determinar usuários ou itens semelhantes para fazer previsões. A seguir, discutiremos como os métodos baseados em vizinhança podem ser usados para prever as classificações de combinações específicas de usuário-item. Existem dois princípios básicos utilizados em modelos baseados em vizinhança:

1. Modelos baseados em usuários: Usuários semelhantes têm avaliações semelhantes para o mesmo item. Portanto, se Alice e Bob avaliaram filmes de forma semelhante no passado, é possível usar as avaliações observadas de Alice no filme Exterminador do Futuro para prever as avaliações não observadas de Bob para esse filme.
2. Modelos baseados em itens: Itens semelhantes são avaliados de forma semelhante pelo mesmo usuário. Portanto, as avaliações de Bob em filmes de ficção científica semelhantes, como Alien e Predador, podem ser usadas para prever sua avaliação em O Exterminador do Futuro.

Como o problema de filtragem colaborativa pode ser visto como uma generalização do problema de modelagem de classificação/regressão, os métodos baseados em vizinhança podem ser vistos como generalizações dos classificadores de vizinhos mais próximos na literatura de aprendizado de máquina. Ao contrário

Tabela 2.1: Cálculo de similaridade usuário-usuário entre o usuário 3 e os demais usuários

Item-Id	1	2	3	4	5	6	Média	Cosseno(i, 3)	Pearson(i, 3)				
ID do usuário													Classificação (usuário-usuário) (usuário-usuário)
1	5	5	5	5	5	6	5,5	0,9567063944	5,467	7,7	4,3	4,3	
2	3	1	1	2	2	3	3	4,1	1,2	4,8	0,981	0,939	
	3	3									1,0	1,0	
3	4									2,2,5	0,789	-1,0	
5										2	0,645	-0,817	

Tabela 2.2: Matriz de classificações da Tabela 2.1 com centralização média para similaridade de cosseno ajustada cálculo entre itens. As similaridades de cosseno ajustadas dos itens 1 e 6 com outros os itens são mostrados nas duas últimas linhas.

Id do item	1	Id do usuário	2	3	4	5	6
usuário	ÿ						
1	2	1,5	0,5		-1,5	-0,5	-1,5
2	1	2,2	1,5 ?	-0,8	-1,8	-0,8	
	1,2 ?	1	1	-1	-1	?	
3	4	-1,5	-0,5 ?	-0,5	0,5	0,5	1,5
5	5	-1		-1	0	1	1
Cosseno(1, j) (item-item)		1	0,735 0,912 -0,848	0,813 -0,990			
Cosseno(6, j) (item-item)		-0,990 -0,622 -0,912 0,829 0,730					1

classificação, onde os vizinhos mais próximos são sempre determinados apenas com base na linha similaridade, é possível encontrar os vizinhos mais próximos na filtragem colaborativa com base de linhas ou colunas. Isso ocorre porque todas as entradas ausentes estão concentradas em um único coluna na classificação, enquanto as entradas ausentes são distribuídas pelas diferentes linhas e colunas na filtragem colaborativa (cf. seção 1.3.1.3 do Capítulo 1). Na discussão a seguir, discutiremos os detalhes dos modelos de vizinhança baseados em usuários e em itens, juntamente com suas variações naturais.

### 2.3.1 Modelos de vizinhança baseados no usuário

Nesta abordagem, os bairros baseados em usuários são definidos para identificar usuários semelhantes a o usuário-alvo para o qual as previsões de classificação estão sendo calculadas. A fim de determinar a vizinhança do usuário alvo i, sua similaridade com todos os outros usuários é computada.

Portanto, uma função de similaridade precisa ser definida entre as classificações especificadas pelos usuários. Tal cálculo de similaridade é complicado porque diferentes usuários podem ter diferentes escalas de classificações. Um usuário pode ser tendencioso em gostar da maioria dos itens, enquanto outro usuário pode ser tendencioso em não gostar da maioria dos itens. Além disso, diferentes usuários podem ter avaliado diferentes itens. Portanto, é necessário identificar mecanismos para abordar essas questões.

Para a matriz de classificações  $m \times n$   $R = [r_{uj}]$  com m usuários e n itens, deixe  $I_u$  denotar o conjunto de índices de itens para os quais as classificações foram especificadas pelo usuário (linha) u. Por exemplo, se as classificações do primeiro, terceiro e quinto itens (colunas) do usuário (linha) u são especificadas (observadas)

e os restantes estão faltando, então temos  $I_u = \{1, 3, 5\}$ . Portanto, o conjunto de itens avaliados por ambos os usuários  $u$  e  $v$  é dado por  $I_u \cap I_v$ . Por exemplo, se o usuário  $v$  avaliou os quatro primeiros itens, então  $I_v = \{1, 2, 3, 4\}$ , e  $I_u \cap I_v = \{1, 3, 5\} \cap \{1, 2, 3, 4\} = \{1, 3\}$ . É possível (e bastante comum) que  $I_u \cap I_v$  seja um conjunto vazio porque as matrizes de classificação são geralmente esparsas. O conjunto  $I_u \cap I_v$  define as classificações mutuamente observadas, que são usadas para calcular a similaridade entre os usuários  $u$  e  $v$  para computação de vizinhança.

Uma medida que captura a similaridade  $\text{Sim}(u, v)$  entre os vetores de classificação de dois usuários  $u$  e  $v$  é o coeficiente de correlação de Pearson. Como  $I_u \cap I_v$  representa o conjunto de índices de itens para os quais tanto o usuário  $u$  quanto o usuário  $v$  possuem classificações especificadas, o coeficiente é calculado apenas para esse conjunto de itens. O primeiro passo é calcular a classificação média  $\bar{y}_u$  para cada usuário  $u$  usando suas classificações especificadas:

$$\bar{y}_u = \frac{\sum_{i \in I_u \cap I_v} r_{ui}}{|I_u \cap I_v|} \quad \bar{y}_u \in \{1, \dots, m\} \quad (2.1)$$

Então, o coeficiente de correlação de Pearson entre as linhas (usuários)  $u$  e  $v$  é definido da seguinte forma:

$$\text{Sim}(u, v) = \text{Pearson}(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{y}_u)(r_{vi} - \bar{y}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{y}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{y}_v)^2}} \quad (2.2)$$

A rigor, a definição tradicional de  $\text{Pearson}(u, v)$  determina que os valores de  $\bar{y}_u$  e  $\bar{y}_v$  devem ser calculados apenas sobre os itens que são avaliados pelos usuários como  $u$  e  $v$ .

Ao contrário da Equação 2.1, tal abordagem levará a um valor diferente de  $\bar{y}_u$ , dependendo da escolha do outro usuário  $v$  para o qual a similaridade de Pearson está sendo computada. No entanto, é bastante comum (e computacionalmente mais simples) calcular cada  $\bar{y}_u$  apenas uma vez para cada usuário  $u$ , de acordo com a Equação 2.1. É difícil argumentar que uma dessas duas maneiras de calcular  $\bar{y}_u$  sempre fornece recomendações estritamente melhores do que a outra. Em casos extremos, onde os dois usuários têm apenas uma classificação mutuamente especificada, pode-se argumentar que o uso da Equação 2.1 para calcular  $\bar{y}_u$  fornecerá resultados mais informativos, porque o coeficiente de Pearson será indeterminado sobre um único item comum na definição tradicional. Portanto, trabalharemos com a suposição mais simples de usar a Equação 2.1 neste capítulo. No entanto, é importante que o leitor tenha em mente que muitas implementações de métodos baseados no usuário calculam  $\bar{y}_u$  e  $\bar{y}_v$  em pares durante o cálculo de Pearson.

O coeficiente de Pearson é calculado entre o usuário-alvo e todos os outros usuários. Uma maneira de definir o grupo de pares do usuário-alvo seria usar o conjunto de  $k$  usuários com o maior coeficiente de Pearson com o alvo. No entanto, como o número de avaliações observadas no grupo de pares top- $k$  de um usuário-alvo pode variar significativamente com o item em questão, os  $k$  usuários mais próximos são encontrados para o usuário-alvo separadamente para cada item previsto, de modo que cada um desses  $k$  usuários tenha classificações especificadas para aquele item. A média ponderada dessas classificações pode ser retornada como a classificação prevista para aquele item. Aqui, cada classificação é ponderada com o coeficiente de correlação de Pearson de seu proprietário para o usuário-alvo.

O principal problema dessa abordagem é que diferentes usuários podem fornecer classificações em escalas diferentes. Um usuário pode classificar todos os itens como positivos, enquanto outro pode classificar todos os itens negativamente. As classificações brutas, portanto, precisam ser centradas na média, por linha, antes de determinar a classificação média (ponderada) do grupo de pares. A classificação centrada na média  $s_{uj}$  de um usuário  $u$  para o item  $j$  é definida subtraindo-se sua classificação média da classificação bruta  $r_{uj}$ .

$$s_{uj} = r_{uj} - \bar{y}_u \quad \bar{y}_u \in \{1, \dots, m\} \quad (2.3)$$

Como antes, a média ponderada da classificação centrada na média de um item no grupo de pares top-k do usuário-alvo u é usada para fornecer uma previsão centrada na média. A classificação média do usuário-alvo é então adicionada de volta a essa previsão para fornecer uma previsão de classificação bruta,  $\hat{r}_{uj}$ , do usuário-alvo u para o item j. A notação de chapéu sobre  $\hat{r}_{uj}$  indica uma classificação prevista, em oposição a uma que já foi observada na matriz de classificações original. Seja  $P_u(j)$  o conjunto de k usuários mais próximos do usuário-alvo u, que especificaram classificações para o item j. Usuários com correlações muito baixas ou negativas com o usuário-alvo u às vezes são filtrados de  $P_u(j)$  como um aprimoramento heurístico. Então, a função de previsão geral baseada em vizinhança é a seguinte:

$$r^u j = \ddot{y} u + \frac{\ddot{v} y P_u(j) - \text{Sim}(u, v) \cdot svj}{\ddot{v} y P_u(j)} = \ddot{y} u + \frac{\text{Sim}(u, v) \cdot (rvj - \ddot{y} v)}{|\text{Sim}(u, v)|} = \frac{\text{Sim}(u, v) \cdot (rvj - \ddot{y} v) y P_u(j)}{|\text{Sim}(u, v)|} \quad (2.4)$$

Essa abordagem mais ampla permite uma série de variações diferentes em termos de como a função de similaridade ou previsão é calculada ou em termos de quais itens são filtrados durante o processo de previsão.

## Exemplo de algoritmo baseado no usuário

Considere o exemplo da Tabela 2.1. Neste caso, as avaliações de cinco usuários 1 a 5 são indicadas para seis itens denotados por 1 a 6. Cada avaliação é extraída do intervalo {1 a 7}. Considere o caso em que o índice do usuário-alvo é 3 e queremos fazer previsões de itens com base nas avaliações da Tabela 2.1. Precisamos calcular as previsões  $\hat{r}_{31}$  e  $\hat{r}_{36}$  do usuário 3 para os itens 1 e 6 a fim de determinar o item mais recomendado.

O primeiro passo é calcular a similaridade entre o usuário 3 e todos os outros usuários. Mostramos duas maneiras possíveis de calcular a similaridade nas duas últimas colunas da mesma tabela. A penúltima coluna mostra a similaridade com base no cosseno bruto entre as avaliações, e a última coluna mostra a similaridade com base no coeficiente de correlação de Pearson.

Por exemplo, os valores de Cosine(1, 3) e Pearson(1, 3) são calculados da seguinte forma:

$$\begin{aligned}
 & 3) = \frac{\bar{y}62 + 72 + 42}{\sqrt{\frac{6 \cdot 3+7 \cdot 3+4 \cdot 1+5 \cdot 1}{6} \cdot \text{Cosseno}(1, 3)}} = 0,956 \\
 & + 52 \cdot \bar{y}32 + 32 + 12 + 12 \cdot \text{Pearson}(1, 3) = (6 \cdot \bar{y}5,5) \cdot (3 \\
 & - \bar{y}2) + (7 \cdot \bar{y}5,5) \cdot (3 \\
 & = \frac{\bar{y}2 + (4 \cdot \bar{y}5,5) \cdot (1 \cdot \bar{y}2) + (5 \cdot \bar{y}5,5) \cdot (1 \cdot \bar{y}2)}{\sqrt{1,52 + 1,52 + (\bar{y}1,5)^2 + (\bar{y}0,5)^2 \cdot 12 + 12 + (\bar{y}1)^2 + (\bar{y}1)^2}} \\
 & = 0,894
 \end{aligned}$$

As similaridades de Pearson e do cosseno bruto do usuário 3 com todos os outros usuários são ilustradas nas duas colunas finais da Tabela 2.1. Observe que o coeficiente de correlação de Pearson é muito mais discriminativo e o sinal do coeficiente fornece informações sobre similaridade e dissimilaridade. Os dois usuários mais próximos do usuário 3 são os usuários 1 e 2, de acordo com ambas as medidas. Utilizando a média ponderada de Pearson das avaliações brutas dos usuários 1 e 2, as seguintes previsões são obtidas para o usuário 3 em relação aos seus itens não avaliados 1 e 6:

$$r^{\wedge}31 = \frac{7 \circ 0,894 + 6 \circ 0,939 \circ 6,49}{0,894 + 0,939 \circ 4} \\ 0,894 + 4 \circ 0,939 r^{\wedge}36 = = \\ 4 \quad \frac{0,894 + 0,939}{0,894 + 0,939}$$

1Em muitos casos, k pares válidos do usuário-alvo u com classificações observadas para o item j podem não existir. Esse cenário é particularmente comum em matrizes de classificações esparsas, como o caso em que o usuário u tem menos de k classificações observadas. Nesses casos, o conjunto  $Pu(j)$  terá cardinalidade menor que k.

Assim, o item 1 deve ser priorizado em relação ao item 6 como recomendação ao usuário 3. Além disso, a previsão sugere que o usuário 3 provavelmente estará mais interessado nos filmes 1 e 6 grau do que qualquer um dos filmes que ela já avaliou. Isso é, no entanto, resultado do viés causado pelo fato de que o grupo de pares {1, 2} de índices de usuários é um grupo muito mais otimista com avaliações positivas, em comparação com o usuário alvo 3. Vamos agora examinar o impacto de classificações centradas na média na previsão. As classificações centradas na média são ilustradas em Tabela 2.2. As previsões correspondentes com a equação centrada na média 2.4 são as seguintes:

$$\begin{array}{rcl} \text{r}^{\wedge}31 = 2+ & \frac{1,5 \ddot{y} 0,894 + 1,2 \ddot{y} 0,939}{0,894 + 0,939} & \ddot{y} 3,35 \\ & \ddot{y} 1,5 \ddot{y} 0,894 \ddot{y} 0,8 \ddot{y} 0,939 & \\ \text{r}^{\wedge}36 = 2+ \ddot{y} 0,86 & \hline & 0,894 + 0,939 \end{array}$$

Assim, o cálculo centrado na média também fornece a previsão de que o item 1 deve ser priorizado em relação ao item 6 como recomendação ao usuário 3. Há, no entanto, uma diferença crucial em relação à recomendação anterior. Neste caso, a classificação prevista para o item 6 é apenas 0,86, que é menor do que todos os outros itens que o usuário 3 classificou. Este é um resultado drasticamente diferente do caso anterior, onde a classificação prevista para o item 6 foi maior do que todos os outros itens que o usuário 3 avaliou. Ao inspecionar visualmente a Tabela 2.1 (ou Tabela 2.2), é de fato evidente que o item 6 deve ser avaliado como muito baixo pelo usuário 3 (em comparação com seus outros itens), porque seus pares mais próximos (usuários 1 e 2) também o classificaram como inferior aos outros itens. Assim, o processo de centralização da média permite uma previsão relativa muito melhor em relação às classificações já observadas. Em muitos casos, também pode afetar a ordem relativa dos itens previstos. A única fraqueza neste resultado é que a previsão a classificação do item 6 é 0,85, o que está fora da faixa de classificações permitidas. Essas classificações podem sempre ser usado para classificação, e o valor previsto pode ser corrigido para o valor mais próximo em o intervalo permitido.

### 2.3.1.1 Variantes da Função de Similaridade

Várias outras variantes da função de similaridade são usadas na prática. Uma variante é usar a função cosseno nas classificações brutas em vez das classificações centradas na média:

$$\text{RawCosine}(u, v) = \frac{\text{kyluylv ruk} \cdot \text{rvk}}{\sqrt{\text{kyluylv}^2} \cdot \sqrt{\text{rvk}^2}} \quad (2.5)$$

Em algumas implementações do cosseno bruto, os fatores de normalização no denominador são baseados em todos os itens especificados e não nos itens classificados mutuamente.

$$\text{RawCosine}(u, v) = \frac{\text{kyluylv ruk} \cdot \text{rvk}}{\sqrt{\text{kylu}^2} \cdot \sqrt{\text{rvk}^2}} \quad (2.6)$$

Em geral, o coeficiente de correlação de Pearson é preferível ao cosseno bruto devido a o efeito de ajuste de viés da centralização média. Este ajuste explica o fato de que diferentes usuários exibem diferentes níveis de generosidade em seus padrões de classificação global.

A confiabilidade da função de similaridade  $\text{Sim}(u, v)$  é frequentemente afetada pelo número de classificações comuns  $|lu \cap lv|$  entre os usuários  $u$  e  $v$ . Quando os dois usuários têm apenas uma pequena número de classificações em comum, a função de similaridade deve ser reduzida com um desconto fator para diminuir a importância desse par de usuários. Este método é conhecido como ponderação de significância. O fator de desconto entra em ação quando o número de classificações comuns

entre os dois usuários for menor que um determinado limite  $\hat{y}$ . O valor do fator de desconto é dado por  $\frac{\min\{|u-y|, \hat{y}\}}{\hat{y}}$ , e sempre se encontra no intervalo [0, 1]. Portanto, o valor descontado similaridade DiscountedSim( $u, v$ ) é dado pelo seguinte:

$$\text{SimDescontado}(u, v) = \text{Sim}(u, v) \cdot \frac{\min\{|u - y|, \hat{y}\}}{\hat{y}} \quad (2.7)$$

A similaridade descontada é usada tanto para o processo de determinação do grupo de pares quanto para calcular a previsão de acordo com a Equação 2.4.

### 2.3.1.2 Variantes da Função de Previsão

Existem muitas variantes da função de previsão usada na Equação 2.4. Por exemplo, em vez de centralizar a classificação bruta  $r_{uj}$  para o valor centralizado  $s_{uj}$ , pode-se usar o Z-score  $z_{uj}$ , que divide ainda mais  $s_{uj}$  com o desvio padrão  $\hat{y}_u$  das classificações observadas do usuário  $u$ . O desvio padrão é definido da seguinte forma:

$$\hat{y}_u = \sqrt{\frac{\sum_{j \in I_u} (r_{uj} - \hat{y}_u)^2}{|I_u| - 1}} \quad \hat{y}_u \in \{1, \dots, m\} \quad (2.8)$$

Então, a classificação padronizada é calculada da seguinte forma:

$$z_{uj} = \frac{r_{uj} - \hat{y}_u}{\hat{y}_u} = \frac{s_{uj}}{\hat{y}_u} \quad (2.9)$$

Seja  $P_u(j)$  o conjunto dos principais  $k$  usuários semelhantes ao usuário alvo  $u$ , para os quais as classificações de item  $j$  foram observados. Neste caso, a classificação prevista  $\hat{r}_{uj}$  do usuário alvo  $u$  para o item  $j$  é o seguinte:

$$\hat{r}_{uj} = \hat{y}_u + \hat{y}_u \cdot \frac{\frac{v_j P_u(j)}{v_j P_u(j)} \cdot \text{Sim}(u, v) \cdot z_{vj}}{|\text{Sim}(u, v)|} \quad (2.10)$$

Observe que a média ponderada precisa ser multiplicada por  $\hat{y}_u$  neste caso. Em geral, se um função  $g(\cdot)$  é aplicada durante a normalização das classificações, então seu inverso precisa ser aplicado durante o processo de previsão final. Embora seja geralmente aceito que a normalização melhora a previsão, parece haver conclusões conflitantes em vários estudos sobre se a centralização média ou o escore Z fornecem resultados de maior qualidade [245, 258]. Um problema com O Z-score indica que as classificações previstas podem frequentemente estar fora da faixa de classificações permitidas. No entanto, mesmo quando os valores previstos estão fora da faixa de classificações permitidas, eles podem ser usados para classificar os itens em ordem de deseabilidade para um determinado grupo. usuário.

Uma segunda questão na previsão é a da ponderação das várias classificações na Equação 2.4. Cada classificação centrada na média  $s_{vj}$  do usuário  $v$  para o item  $j$  é ponderada com a similaridade  $\text{Sim}(u, v)$  do usuário  $v$  para o usuário alvo  $u$ . Enquanto o valor de  $\text{Sim}(u, v)$  foi escolhido para ser o Coeficiente de correlação de Pearson, uma prática comumente usada é amplificá-lo exponenciando elevado à potência de  $\hat{y}$ . Em outras palavras, temos:

$$\text{Sim}(u, v) = \text{Pearson}(u, v)^\hat{y} \quad (2.11)$$

Ao escolher  $\hat{y} > 1$ , é possível ampliar a importância da similaridade na ponderação da Equação 2.4.

Conforme discutido anteriormente, os métodos de filtragem colaborativa baseados em vizinhança são generalizações dos métodos de classificação/regressão do vizinho mais próximo. A discussão acima mencionada está mais próxima da modelagem de regressão do vizinho mais próximo do que da classificação do vizinho mais próximo, porque o valor previsto é tratado como uma variável contínua ao longo do processo de previsão. Também é possível criar uma função de previsão mais próxima de um método de classificação, tratando as classificações como valores categóricos e ignorando a ordenação entre as classificações. Uma vez identificado o grupo de pares do usuário-alvo  $u$ , o número de votos para cada valor de classificação possível (por exemplo, Concordo, Neutro, Discordo) dentro do grupo de pares é determinado. A classificação com o maior número de votos é prevista como a relevante. Essa abordagem tem a vantagem de fornecer a classificação mais provável em vez da classificação média. Essa abordagem geralmente é mais eficaz em casos em que o número de classificações distintas é pequeno. Também é útil no caso de classificações ordinais, em que as distâncias exatas entre pares de valores de classificação não são definidas. Em casos em que a granularidade das classificações é alta, essa abordagem é menos robusta e perde muitas informações de ordenação entre as classificações.

### 2.3.1.3 Variações na filtragem de grupos de pares

O grupo de pares de um usuário-alvo pode ser definido e filtrado de diversas maneiras. A abordagem mais simples é usar os  $k$  usuários mais semelhantes ao usuário-alvo como seu grupo de pares. No entanto, essa abordagem pode incluir usuários com correlação fraca ou negativa com o alvo. Usuários com correlação fraca podem aumentar o erro na previsão. Além disso, classificações com correlação negativa geralmente não têm tanto valor preditivo em termos de potencial inversão das classificações. Embora a função de previsão permita tecnicamente o uso de classificações fracas ou negativas, seu uso não é consistente com o princípio mais amplo dos métodos de vizinhança. Portanto, classificações com correlações fracas ou negativas são frequentemente descartadas.

### 2.3.1.4 Impacto da Cauda Longa

Conforme discutido na seção 2.2, a distribuição de avaliações geralmente apresenta uma distribuição de cauda longa em muitos cenários reais. Alguns filmes podem ser muito populares e podem aparecer repetidamente como itens com avaliações comuns por diferentes usuários. Essas avaliações podem, às vezes, priorizar a qualidade das recomendações, pois tendem a ser menos discriminatórias entre diferentes usuários. O impacto negativo dessas recomendações pode ser experimentado tanto durante o cálculo do grupo de pares quanto durante o cálculo da predição (cf. Equação 2.4). Essa noção é semelhante, em princípio, à deterioração na qualidade da recuperação causada por palavras populares e não informativas (por exemplo, "um", "uma", "o") em aplicativos de recuperação de documentos. Portanto, as soluções propostas usadas na filtragem colaborativa também são semelhantes às usadas na literatura de recuperação de informações. Assim como a noção de Frequência Inversa de Documentos (idf) existe na literatura de recuperação de informações [400], pode-se usar a noção de Frequência Inversa de Usuários neste caso. Se  $m_j$  for o número de avaliações do item  $j$  e  $m$  for o número total de usuários, então o peso  $w_j$  do item  $j$  é definido como o seguinte:

$$w_j = \log \frac{m}{m_j} \quad j \in \{1 \dots n\} \quad (2.12)$$

Cada item  $j$  é ponderado por  $w_j$  tanto durante o cálculo de similaridade quanto durante o processo de recomendação. Por exemplo, o coeficiente de correlação de Pearson pode ser modificado para incluir os pesos da seguinte forma:

$$\text{Pearson}(u, v) = \frac{k_{\text{lyluylv}} \text{semana} \cdot (r_{\text{uk}} \ddot{\text{y}} u) \cdot (r_{\text{vk}} \ddot{\text{y}} v)}{k_{\text{lyluylv}} \text{semana} \cdot (r_{\text{uk}} \ddot{\text{y}} u)^2 \cdot k_{\text{lyluylv}} \text{semana} \cdot (r_{\text{vk}} \ddot{\text{y}} v)^2} \quad (2.13)$$

A ponderação de itens também pode ser incorporada em outros métodos de filtragem colaborativa. Por exemplo, a etapa final de predição de algoritmos de filtragem colaborativa baseados em itens pode ser modificada. usar pesos, mesmo que a similaridade do cosseno ajustada entre dois itens permaneça inalterada pelos pesos.

### 2.3.2 Modelos de vizinhança baseados em itens

Em modelos baseados em itens, os grupos de pares são construídos em termos de itens e não de usuários. Portanto, as similaridades precisam ser calculadas entre os itens (ou colunas na matriz de classificação). Antes de calcular as similaridades entre as colunas, cada linha da matriz de classificação é centrado em uma média de zero. Como no caso das classificações baseadas no usuário, a classificação média de cada item na matriz de classificação é subtraída de cada classificação para criar uma média centrada matriz. Este processo é idêntico ao discutido anteriormente (ver Equação 2.3), que resulta no cálculo de classificações centradas na média  $s_{ui}$ . Sejam  $U_i$  os índices do conjunto de usuários que especificaram classificações para o item  $i$ . Portanto, se o primeiro, terceiro e quarto usuários tiverem classificações especificadas para o item  $i$ , então temos  $U_i = \{1, 3, 4\}$ .

Então, a similaridade do cosseno ajustada entre os itens (colunas)  $i$  e  $j$  é definida como segue:

$$\text{CossenoAjustado}(i, j) = \frac{\sum_{u \in U_i} s_{ui} \cdot \sum_{u \in U_j} s_{uj}}{\sqrt{\sum_{u \in U_i} s_{ui}^2} \cdot \sqrt{\sum_{u \in U_j} s_{uj}^2}} \quad (2.14)$$

Essa similaridade é chamada de similaridade de cosseno ajustada porque as classificações são centradas na média antes do cálculo do valor de similaridade. Embora a correlação de Pearson também possa ser usado nas colunas no caso do método baseado em itens, o cosseno ajustado geralmente fornece resultados superiores.

Considere o caso em que a classificação do item alvo  $t$  para o usuário  $u$  precisa ser determinada. O primeiro passo é determinar os  $k$  itens mais semelhantes ao item  $t$  com base na similaridade de cosseno ajustada mencionada anteriormente. Deixe os  $k$  itens mais semelhantes ao item  $t$ , para os quais o usuário  $u$  especificou classificações, denotadas por  $Q_t(u)$ . O valor médio ponderado dessas classificações (brutas) as classificações são relatadas como o valor previsto. O peso do item  $j$  nesta média é igual a a similaridade do cosseno ajustada entre o item  $j$  e o item alvo  $t$ . Portanto, a previsão a classificação  $\hat{r}_{ut}$  do usuário  $u$  para o item alvo  $t$  é a seguinte:

$$\hat{r}_{ut} = \frac{\sum_j \text{Cosseno ajustado}(j, t) \cdot r_{uj}}{\sum_j |\text{CossenoAjustado}(j, t)|} \quad (2.15)$$

A ideia básica é alavancar as avaliações do próprio usuário sobre itens semelhantes na etapa final de fazendo a previsão. Por exemplo, em um sistema de recomendação de filmes, o item peer grupo normalmente será de filmes de um gênero semelhante. O histórico de avaliações do mesmo usuário em tais filmes são um indicador muito confiável dos interesses daquele usuário.

A seção anterior discutiu uma série de variantes da abordagem básica para filtragem colaborativa baseada no usuário. Como os algoritmos baseados em itens são muito semelhantes aos baseados no usuário, algoritmos, variantes semelhantes da função de similaridade e da função de previsão podem ser projetado para métodos baseados em itens.

#### Exemplo de algoritmo baseado em itens

Para ilustrar o algoritmo baseado em itens, usaremos o mesmo exemplo da Tabela 2.1, que foi aproveitado para demonstrar o algoritmo baseado no usuário. As avaliações ausentes do usuário

3 são previstos com o algoritmo baseado em itens. Como as classificações dos itens 1 e 6 estão ausentes para o usuário 3, a similaridade das colunas dos itens 1 e 6 precisa ser calculada em relação às outras colunas (itens).

Primeiro, a similaridade entre os itens é calculada após o ajuste para centralização média. A matriz de classificações centrada na média é ilustrada na Tabela 2.2. As similaridades dos cossenos ajustados correspondentes de cada item com 1 e 6, respectivamente, são indicadas nas duas últimas linhas da tabela. Por exemplo, o valor do cosseno ajustado entre os itens 1 e 3, denotado por AdjustedCosine(1, 3), é o seguinte:

$$\text{CossenoAjustado}(1, 3) = \frac{1,5 \ddot{\circ} 1,5 + (\ddot{\circ}1,5) \ddot{\circ} (\ddot{\circ}0,5) + (\ddot{\circ}1) \ddot{\circ} (\ddot{\circ}1) 1,52 +}{(\ddot{\circ}1,5)^2 + (\ddot{\circ}1)^2 \cdot 1,52 + (\ddot{\circ}0,5)^2 + (\ddot{\circ}1)^2} = 0,912$$

Outras similaridades item-item são calculadas de forma exatamente análoga e ilustradas nas duas últimas linhas da Tabela 2.2. É evidente que os itens 2 e 3 são mais semelhantes ao item 1, enquanto os itens 4 e 5 são mais semelhantes ao item 6. Portanto, a média ponderada das avaliações brutas do usuário 3 para os itens 2 e 3 é usada para prever a avaliação  $\hat{r}_{31}$  do item 1, enquanto a média ponderada das avaliações brutas do usuário 3 para os itens 4 e 5 é usada para prever a avaliação  $\hat{r}_{36}$  do item 6:

$$\hat{r}_{31} = \frac{3 \ddot{\circ} 0,735 + 3 \ddot{\circ} 0,912}{0,735 + 0,912} = 3 \\ \frac{0,829 + 1 \ddot{\circ} 0,730 \hat{r}_{36} = 1}{0,829 + 0,730}$$

Assim, o método baseado em itens também sugere que o item 1 tem maior probabilidade de ser preferido pelo usuário 3 do que o item 6. No entanto, neste caso, como as avaliações são previstas usando as avaliações do próprio usuário 3, as avaliações previstas tendem a ser muito mais consistentes com as demais avaliações deste usuário. Como exemplo específico, vale ressaltar que a avaliação prevista do item 6 não está mais fora da faixa de avaliações permitidas, como no caso do método baseado em usuários.

A maior precisão de previsão do método baseado em itens é sua principal vantagem. Em alguns casos, o método baseado em itens pode fornecer um conjunto diferente de recomendações top-k, embora as listas de recomendações geralmente sejam aproximadamente semelhantes.

### 2.3.3 Implementação eficiente e complexidade computacional

Métodos baseados em vizinhança são sempre utilizados para determinar as melhores recomendações de itens para um usuário-alvo ou as melhores recomendações de usuários para um item-alvo. A discussão acima demonstra apenas como prever as classificações para uma combinação específica de usuário-item, mas não discute o processo de classificação em si. Uma abordagem simples consiste em calcular todas as previsões de classificação possíveis para os pares de usuário-item relevantes (por exemplo, todos os itens para um usuário específico) e, em seguida, classificá-los. Embora esta seja a abordagem básica utilizada nos sistemas de recomendação atuais, é importante observar que o processo de previsão para muitas combinações de usuário-item reutiliza muitas quantidades intermediárias. Portanto, é aconselhável ter uma fase offline para armazenar esses cálculos intermediários e, em seguida, aproveitá-los no processo de classificação.

Os métodos baseados em vizinhança são sempre divididos em uma fase offline e uma fase online. Na fase offline, os valores de similaridade usuário-usuário (ou item-item) e os grupos de pares dos usuários (ou itens) são computados. Para cada usuário (ou item), o grupo de pares relevante é pré-armazenado com base nesse cálculo. Na fase online, esses valores de similaridade e grupos de pares são aproveitados para fazer previsões com o uso de relacionamentos como a Equação 2.4. Seja  $n$  o número máximo de avaliações especificadas de um usuário (linha), e

mm é o número máximo de classificações especificadas de um item (coluna). Observe que n é o tempo máximo de execução para calcular a similaridade entre um par de usuários (linhas) e m é o tempo máximo de execução para calcular a similaridade entre um par de itens (colunas). No caso de métodos baseados em usuários, o processo de determinação do grupo de pares de um usuário-alvo pode exigir tempo  $O(m \cdot n)$ . Portanto, o tempo de execução offline para calcular os grupos de pares de todos os usuários é dado por  $O(m^2 \cdot n)$ . Para métodos baseados em itens, o tempo de execução offline correspondente é dado por  $O(n^2 \cdot m)$ .

Para poder usar a abordagem para valores variáveis de k, pode ser necessário armazenar todos os pares de similaridades diferentes de zero entre pares de usuários (ou itens). Portanto, os requisitos de espaço dos métodos baseados em usuários são  $O(m^2)$ , enquanto os requisitos de espaço dos métodos baseados em itens são  $O(n^2)$ . Como o número de usuários é tipicamente maior que o número de itens, os requisitos de espaço dos métodos baseados em usuários são geralmente maiores que os dos métodos baseados em itens.

O cálculo online do valor previsto, de acordo com a Equação 2.4, requer tempo  $O(k)$  para os métodos baseados em usuários e em itens, onde k é o tamanho da vizinhança usuário/item usada para a previsão. Além disso, se essa previsão precisar ser executada em todos os itens para classificá-los para um usuário-alvo, o tempo de execução será  $O(k \cdot n)$  para os métodos baseados em usuários e em itens. Por outro lado, um comerciante pode ocasionalmente desejar determinar os principais usuários a serem segmentados para um item específico. Nesse caso, a previsão precisa ser executada em todos os usuários para classificá-los para um item-alvo, e o tempo de execução é  $O(k \cdot m)$  para os métodos baseados em usuários e em itens. Vale ressaltar que a principal complexidade computacional dos métodos baseados em vizinhança reside na fase offline, que precisa ser executada ocasionalmente. Como resultado, os métodos baseados em vizinhança tendem a ser eficientes quando usados para previsão online. Afinal, pode-se dar ao luxo de ser generoso ao alojar significativamente mais tempo computacional para a fase offline.

### 2.3.4 Comparando métodos baseados em usuários e em itens

Métodos baseados em itens geralmente fornecem recomendações mais relevantes devido ao fato de que as próprias avaliações do usuário são usadas para realizar a recomendação. Em métodos baseados em itens, itens semelhantes são identificados a um item alvo, e as próprias avaliações do usuário sobre esses itens são usadas para extrapolar as avaliações do alvo. Por exemplo, itens semelhantes a um filme histórico alvo podem ser um conjunto de outros filmes históricos. Nesses casos, as próprias recomendações do usuário para o conjunto semelhante podem ser altamente indicativas de sua preferência pelo alvo. Este não é o caso dos métodos baseados no usuário, nos quais as avaliações são extrapoladas de outros usuários, que podem ter interesses sobrepostos, mas diferentes. Como resultado, os métodos baseados em itens geralmente apresentam maior precisão.

Embora as recomendações baseadas em itens geralmente tenham maior probabilidade de serem precisas, a precisão relativa entre os métodos baseados em itens e os baseados no usuário também depende do conjunto de dados em questão. Como você aprenderá no Capítulo 12, métodos baseados em itens também são mais robustos a ataques de propaganda enganosa em sistemas de recomendação. Por outro lado, são precisamente essas diferenças que podem levar a uma maior diversidade no processo de recomendação para métodos baseados no usuário em comparação com métodos baseados em itens. Diversidade se refere ao fato de que os itens na lista classificada tendem a ser um pouco diferentes. Se os itens não forem diversos, se o usuário não gostar do primeiro item, ele pode não gostar de nenhum dos outros itens da lista. Uma maior diversidade também incentiva a serendipidade, por meio da qual itens um tanto surpreendentes e interessantes são descobertos. Métodos baseados em itens podem, às vezes, recomendar itens óbvios ou itens que não são novos em experiências anteriores do usuário. As noções de novidade, diversidade e serendipidade são discutidas em detalhes no Capítulo 7. Sem novidade, diversidade e serendipidade suficientes, os usuários podem ficar entediados com recomendações muito semelhantes ao que já assistiram.

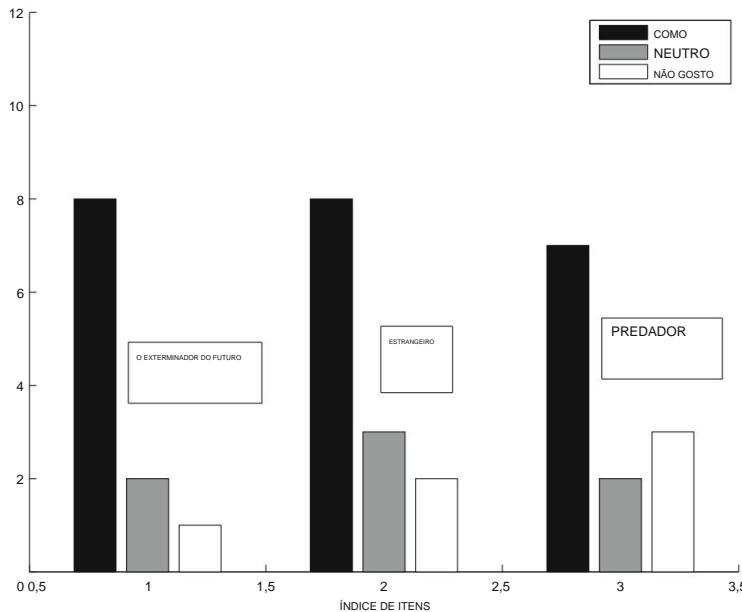


Figura 2.2: Explicando as principais recomendações de Alice com seu histograma de classificação de vizinhos

Os métodos baseados em itens também podem fornecer uma razão concreta para a recomendação. Por exemplo, a Netflix geralmente fornece recomendações com declarações como as seguintes:

Como você assistiu “Secrets of the Wings”, [as recomendações são] Lista .

Tais explicações podem ser abordadas concretamente com métodos baseados em itens<sup>2</sup>, utilizando as vizinhanças de itens. Por outro lado, essas explicações são mais difíceis de abordar com métodos baseados em usuários, pois o grupo de pares é simplesmente um conjunto de usuários anônimos e não pode ser usado diretamente no processo de recomendação.

Métodos baseados no usuário fornecem diferentes tipos de explicações. Por exemplo, considere um cenário em que os filmes O Exterminador do Futuro, Alien e Predador são recomendados para Alice.

Em seguida, um histograma das avaliações de seus vizinhos para esses filmes pode ser mostrado a ela. Um exemplo desse histograma é mostrado na Figura 2.2. Esse histograma pode ser usado por Alice para obter uma ideia de quanto ela pode gostar desse filme. No entanto, o poder desse tipo de explicação é um tanto limitado, pois não dá a Alice uma ideia de como esses filmes se relacionam com seus próprios gostos ou com os de amigos que ela realmente conhece e em quem confia. Observe que a identidade de seus vizinhos geralmente não é disponibilizada para Alice devido a questões de privacidade.

Por fim, os métodos baseados em itens são mais estáveis com alterações nas classificações. Isso ocorre por dois motivos. Primeiro, o número de usuários geralmente é muito maior do que o número de itens.

Nesses casos, dois usuários podem ter um número muito pequeno de itens avaliados mutuamente, mas é mais provável que dois itens tenham um número maior de usuários que os avaliaram conjuntamente. No caso de métodos baseados em usuários, a adição de algumas avaliações pode alterar drasticamente os valores de similaridade. Este não é o caso dos métodos baseados em itens, que são mais estáveis a mudanças nos valores das avaliações. Em segundo lugar, novos usuários provavelmente serão adicionados com mais frequência em

<sup>2</sup>O método preciso usado pela Netflix é proprietário e, portanto, desconhecido. No entanto, com base em itens métodos fornecem uma metodologia viável para atingir objetivos semelhantes.

sistemas comerciais do que novos itens. Nesses casos, o cálculo da vizinhança dos itens pode ser feito apenas ocasionalmente, pois é improvável que a vizinhança dos itens mude drasticamente com a adição de novos usuários. Por outro lado, o cálculo da vizinhança dos usuários precisa ser realizado com mais frequência com a adição de novos usuários. Nesse contexto, a manutenção incremental do modelo de recomendação é mais desafiadora no caso de métodos baseados em usuários.

### 2.3.5 Pontos fortes e fracos dos métodos baseados em vizinhança

Os métodos de vizinhança apresentam diversas vantagens relacionadas à sua simplicidade e abordagem intuitiva. Devido à sua abordagem simples e intuitiva, são fáceis de implementar e depurar. Muitas vezes, é fácil justificar a recomendação de um item específico, e a interpretabilidade dos métodos baseados em itens é particularmente notável. Tais justificativas geralmente não estão facilmente disponíveis em muitos dos métodos baseados em modelos discutidos em capítulos posteriores.

Além disso, as recomendações são relativamente estáveis com a adição de novos itens e usuários. Também é possível criar aproximações incrementais desses métodos.

A principal desvantagem desses métodos é que a fase offline pode, às vezes, ser impraticável em ambientes de larga escala. A fase offline do método baseado no usuário requer pelo menos  $O(m^2)$  de tempo e espaço. Isso pode, às vezes, ser muito lento ou consumir muito espaço em hardware de desktop, quando  $m$  é da ordem de dezenas de milhões. No entanto, a fase online dos métodos de vizinhança é sempre eficiente. A outra desvantagem principal desses métodos é sua cobertura limitada devido à escassez. Por exemplo, se nenhum dos vizinhos mais próximos de John tiver avaliado o Terminator, não será possível fornecer uma previsão de classificação do Terminator para John.

Por outro lado, nos importamos apenas com os  $k$  itens mais importantes de John na maioria das configurações de recomendação. Se nenhum dos vizinhos mais próximos de John avaliou o Exterminador do Futuro, isso pode ser uma evidência de que este filme não é uma boa recomendação para John. A escassez também cria desafios para cálculos robustos de similaridade quando o número de itens avaliados mutuamente entre dois usuários é pequeno.

### 2.3.6 Uma visão unificada de métodos baseados em usuários e itens

As respectivas fraquezas dos métodos baseados no usuário e nos itens surgem do fato de que o primeiro ignora a similaridade entre as colunas da matriz de classificações, enquanto o último ignora a similaridade entre as linhas ao determinar as entradas mais semelhantes.

Surge uma questão natural: podemos determinar as entradas mais semelhantes a uma entrada de destino unificando os dois métodos? Dessa forma, não é necessário ignorar a similaridade ao longo de linhas ou colunas. Em vez disso, é possível combinar as informações de similaridade entre linhas e colunas.

Para atingir esse objetivo, é crucial entender que os métodos baseados no usuário e nos itens são quase idênticos (com algumas pequenas diferenças), uma vez que as linhas foram centradas na média. Podemos assumir, sem perda de generalidade, que as linhas da matriz de classificações são centradas na média, pois a média de cada linha pode ser adicionada a cada entrada após a previsão. Também é importante notar que, se as linhas forem centradas na média, o coeficiente de correlação de Pearson entre as linhas é idêntico<sup>3</sup> ao coeficiente de cosseno. Com base nisso

---

<sup>3</sup>Pode haver algumas pequenas diferenças dependendo de como a média é calculada para cada linha dentro do coeficiente de Pearson. Se a média de cada linha for calculada usando todas as entradas observadas dessa linha (em vez de apenas as entradas mutuamente especificadas), o coeficiente de correlação de Pearson será idêntico ao coeficiente de cosseno para matrizes centradas na média por linha.

suposição, os métodos baseados no usuário e no item podem ser descritos de forma unificada para prever a entrada ruj na matriz de classificação R:

1. Para uma entrada de destino ( $u, j$ ), determine as linhas/colunas mais semelhantes da matriz de classificações usando o coeficiente cosseno entre linhas/colunas. Para métodos baseados em usuários, são usadas linhas, enquanto para métodos baseados em itens, são usadas colunas.
2. Preveja a entrada alvo ( $u, j$ ) usando uma combinação ponderada das classificações mais linhas/colunas semelhantes determinadas na primeira etapa.

Observe que a descrição acima ignora as linhas ou colunas em cada etapa. É possível, é claro, propor uma descrição generalizada das etapas acima mencionadas, na qual as informações de similaridade e predição ao longo das linhas e colunas são combinadas:

1. Para uma entrada de destino ( $u, j$ ), determine as entradas mais semelhantes da matriz de classificações com o uso de uma função de combinação da similaridade entre linhas e colunas. Por exemplo, pode-se usar a soma da similaridade do cosseno entre linhas e entre colunas para determinar as entradas mais semelhantes na matriz de classificações para ( $u, j$ ).
2. Preveja a entrada alvo ( $u, j$ ) usando uma combinação ponderada das classificações nas entradas mais semelhantes determinadas na primeira etapa. Os pesos são baseados nas similaridades calculadas na primeira etapa.

Destacamos as etapas que são diferentes no método generalizado. Essa abordagem funde as similaridades ao longo de linhas e colunas com o uso de uma função de combinação. Pode-se experimentar o uso de diversas funções de combinação para obter os resultados mais eficazes. Descrições detalhadas desses métodos unificados podem ser encontradas em [613, 622]. Esse princípio básico também é utilizado no modelo multidimensional de sistemas de recomendação sensíveis ao contexto, no qual as similaridades ao longo de usuários, itens e outras dimensões contextuais são unificadas em uma única estrutura (cf. seção 8.5.1 do Capítulo 8).

## 2.4 Métodos de agrupamento e baseados em vizinhança

O principal problema com métodos baseados em vizinhança é a complexidade da fase offline, que pode ser bastante significativa quando o número de usuários ou o número de itens é muito grande. Por exemplo, quando o número de usuários  $m$  é da ordem de algumas centenas de milhões, o tempo de execução  $O(m^2 \cdot n)$  de um método baseado em usuário se tornará impraticável mesmo para cálculos offline ocasionais. Considere o caso em que  $m = 108$  e  $n = 100$ . Nesse caso,  $O(m^2 \cdot n) = O(1018)$  operações serão necessárias. Se fizermos a suposição conservadora de que cada operação requer um ciclo de máquina elementar, um computador de 10 GHz exigirá 108 segundos, o que é aproximadamente 115,74 dias. Claramente, tal abordagem não será muito prática do ponto de vista da escalabilidade.

A ideia principal dos métodos baseados em agrupamento é substituir a fase de cálculo offline do vizinho mais próximo por uma fase de agrupamento offline. Assim como a fase offline do vizinho mais próximo cria um grande número de grupos de pares, que são centralizados em cada alvo possível, o processo de agrupamento cria um número menor de grupos de pares que não são necessariamente centralizados em cada alvo possível. O processo de agrupamento é muito mais eficiente do que o tempo  $O(m^2 \cdot n)$  necessário para a construção dos grupos de pares de cada alvo possível. Uma vez construídos os agrupamentos, o processo de previsão de classificações é semelhante à abordagem usada na Equação 2.4. A principal diferença é que os  $k$  pares mais próximos dentro do mesmo agrupamento são usados para realizar a previsão. Vale ressaltar que o cálculo de similaridade em pares

precisa ser realizado apenas dentro do mesmo cluster e, portanto, a abordagem pode ser significativamente mais eficiente. Essa eficiência resulta em alguma perda de precisão porque o conjunto de vizinhos mais próximos de cada alvo dentro de um cluster é de qualidade inferior ao conjunto de dados. Além disso, a granularidade do cluster regula o equilíbrio entre precisão e eficiência. Quando os clusters são granulares, a eficiência melhora, mas a precisão é reduzida. Em muitos casos, ganhos muito grandes em eficiência podem ser obtidos com pequenas reduções na precisão. Quando as matrizes de classificação são muito grandes, essa abordagem fornece uma alternativa muito prática a um custo baixo.

Um desafio com o uso dessa abordagem é o fato de que a matriz de classificações é incompleta. Portanto, os métodos de agrupamento precisam ser adaptados para trabalhar com conjuntos de dados massivamente incompletos. Nesse contexto, os métodos k-means podem ser facilmente adaptados a dados incompletos. A ideia básica de uma abordagem k-means é trabalhar com  $k$  pontos centrais (ou "médias"), que servem como representantes de  $k$  clusters diferentes. Nos métodos k-means, a solução para um agrupamento pode ser totalmente representada pela especificação desses  $k$  representantes. Dado um conjunto de  $k$  representantes  $Y_1 \dots Y_k$ , cada ponto de dados é atribuído ao seu representante mais próximo com o uso de uma função de similaridade ou distância. Portanto, o particionamento de dados pode ser definido exclusivamente pelo conjunto de representantes. Para um conjunto de dados  $m \times n$ , cada representante  $Y_i$  é um ponto de dados  $n$ -dimensional, que é um ponto central do  $i$ -ésimo cluster. Idealmente, gostaríamos que o representante central fosse a média do cluster.

Portanto, os clusters são dependentes dos representantes e vice-versa. Tal interdependência é alcançada com uma abordagem iterativa. Começamos com um conjunto de representantes  $Y_1 \dots Y_k$ , que podem ser pontos escolhidos aleatoriamente, gerados no intervalo do espaço de dados. Calculamos iterativamente as partições do cluster usando os representantes e, em seguida, recalculamos os representantes como os centroides dos clusters resultantes. Ao calcular os centroides, deve-se tomar cuidado para usar apenas os valores observados em cada dimensão. Essa abordagem iterativa em duas etapas é executada até a convergência. A abordagem em duas etapas é resumida da seguinte forma:

1. Determine os clusters  $C_1 \dots C_k$  atribuindo cada linha na matriz  $mxn$  ao seu representante mais próximo de  $Y_1 \dots Y_k$ . Normalmente, a distância euclidiana ou a distância de Manhattan é usada para o cálculo de similaridade.
2. Para cada  $i \in \{1 \dots k\}$ , redefina  $Y_i$  para o centroide do conjunto atual de pontos em  $C_i$ .

O principal problema com o uso dessa abordagem é que a matriz de classificações  $m \times n$  é incompleta. Portanto, o cálculo da média e dos valores de distância torna-se indefinido.

No entanto, é relativamente fácil calcular as médias usando apenas os valores observados dentro de um cluster. Em alguns casos, o próprio centroide pode não ser totalmente especificado, quando nenhuma classificação é especificada para um ou mais itens no cluster. Os valores de distância são calculados usando apenas o subconjunto de dimensões, que são especificadas tanto para o ponto de dados quanto para o representante do cluster. A distância também é dividida pelo número de dimensões usadas no cálculo.

Isso é feito para ajustar o fato de que diferentes números de dimensões são usados para calcular a distância de um ponto de dados a vários centroides, quando todos os centroides não são totalmente especificados. Nesse contexto, a distância de Manhattan produz melhores ajustes do que a distância euclidiana, e o valor normalizado pode ser interpretado mais facilmente como uma distância média ao longo de cada valor observado.

A abordagem mencionada agrupa as linhas para filtragem colaborativa baseada no usuário. Em métodos baseados em itens, seria necessário agrupar as colunas. A abordagem é exatamente semelhante, exceto pelo fato de ser aplicada às colunas e não às linhas. Diversos métodos de agrupamento para filtragem colaborativa eficiente são discutidos em [146, 167, 528, 643],

[\[644, 647\]](#). Alguns desses métodos são baseados no usuário, enquanto outros são baseados em itens. Vários métodos de coagrupamento [\[643\]](#) podem ser usados para agrupar linhas e colunas simultaneamente.

## 2.5 Métodos de redução de dimensionalidade e vizinhança

---

Métodos de redução de dimensionalidade podem ser usados para aprimorar métodos baseados em vizinhança, tanto em termos de qualidade quanto de eficiência. Em particular, embora similaridades pareadas sejam difíceis de calcular de forma robusta em matrizes de classificação esparsas, a redução de dimensionalidade fornece uma representação densa de baixa dimensão em termos de fatores latentes. Portanto, esses modelos também são chamados de modelos de fatores latentes. Mesmo quando dois usuários têm poucos itens avaliados em comum, é possível calcular uma distância entre seus vetores latentes de baixa dimensão.

Além disso, é mais eficiente determinar os grupos de pares com vetores latentes de baixa dimensionalidade. Antes de discutir os detalhes dos métodos de redução de dimensionalidade, faremos alguns comentários sobre duas maneiras distintas pelas quais modelos de fatores latentes são usados em sistemas de recomendação:

1. Uma representação reduzida dos dados pode ser criada em termos de fatores latentes por linha ou em termos de fatores latentes por coluna. Em outras palavras, a representação reduzida comprimirá a dimensionalidade do item ou a dimensionalidade do usuário em fatores latentes. Essa representação reduzida pode ser usada para aliviar o problema de esparsidade em modelos baseados em vizinhança. Dependendo de qual dimensão foi comprimida em fatores latentes, a representação reduzida pode ser usada para algoritmos de vizinhança baseados no usuário ou em algoritmos de vizinhança baseados em itens.
  
2. As representações latentes do espaço de linhas e do espaço de colunas são determinadas simultaneamente. Essas representações latentes são usadas para reconstruir toda a matriz de classificações de uma só vez, sem o uso de métodos baseados em vizinhança.

Como a segunda classe de métodos não está diretamente relacionada aos métodos baseados em vizinhança, ela não será discutida neste capítulo. Uma discussão detalhada da segunda classe de métodos será fornecida no Capítulo 3. Neste capítulo, nos concentraremos apenas na primeira classe de métodos.

Para facilitar a discussão, descreveremos primeiro apenas o método de filtragem colaborativa baseado no usuário. Nos métodos de filtragem colaborativa baseados no usuário, a ideia básica é transformar a matriz de classificações  $m \times n$  R em um espaço de menor dimensão usando a análise de componentes principais. A matriz R resultante tem tamanho  $m \times d$ , onde  $d \ll n$ . Assim, cada um dos vetores  $n$ -dimensionais (esparsos) de classificações correspondentes a um usuário é transformado em um espaço  $d$ -dimensional reduzido. Além disso, diferentemente do vetor de classificações original, cada uma das  $d$  dimensões é totalmente especificada. Após a determinação dessa representação  $d$ -dimensional de cada usuário, a similaridade é calculada a partir do usuário-alvo para cada usuário usando a representação reduzida.

Os cálculos de similaridade na representação reduzida são mais robustos porque o novo vetor de baixa dimensão é totalmente especificado. Além disso, os cálculos de similaridade são mais eficientes devido à baixa dimensionalidade da representação latente. Um simples cosseno ou produto escalar nos vetores reduzidos é suficiente para calcular a similaridade nesta representação reduzida.

Resta descrever como a representação de baixa dimensão de cada ponto de dados é calculada. A representação de baixa dimensão pode ser calculada usando métodos do tipo SVD ou métodos do tipo PCA. A seguir, descrevemos um método do tipo SVD.

Tabela 2.3: Exemplo de viés na estimativa de covariâncias

Índice do usuário	O Poderoso	Chefão	Gladiador	Nero
1	1	1	1	1
2	7	7	7	7
	3	1	1	1
3 4	5	7	7	7
5	3	1	?	?
6	5	7	?	?
7	3	1	?	?
8	5	7	?	?
9	3	1	?	?
10	5	7	?	?
11	3			?
12	5	1 7		?

O primeiro passo é aumentar a matriz de classificações incompletas  $m \times n$  R para preencher a falta entradas. A entrada ausente é estimada como sendo igual à média da linha correspondente na matriz (ou seja, a classificação média do usuário correspondente). Uma abordagem alternativa é para estimar a entrada ausente como a média da coluna correspondente na matriz (ou seja, a classificação média do item correspondente). Seja a matriz resultante denotada por  $R_f$ . Em seguida, calculamos a matriz de similaridade  $n \times n$  entre pares de itens, que é dada por  $S = R^T f R_f$ . Esta matriz é semidefinida positiva. Para determinar a base dominante vetores de  $R_f$  para SVD, realizamos a diagonalização da matriz de similaridade S da seguinte forma:

$$S = P \tilde{y} P^T \quad (2.16)$$

Aqui, P é uma matriz  $n \times n$ , cujas colunas contêm os autovetores orthonormais de S.  $\tilde{y}$  é uma matriz diagonal contendo os autovalores não negativos de S ao longo de sua diagonal. Seja  $P_d$  seja a matriz  $n \times d$  contendo apenas as colunas de P correspondentes ao maior d autovetores. Então, a representação de baixa dimensão de  $R_f$  é dada pelo produto matricial  $R_f P_d$ . Observe que as dimensões da representação reduzida  $R_f P_d$  são  $m \times d$ , porque  $R_f$  é uma matriz  $m \times n$  e  $P_d$  é uma matriz  $n \times d$ . Portanto, cada um dos m usuários é agora representado em um espaço d-dimensional. Esta representação é então usada para determinar o grupo de pares de cada usuário. Uma vez determinados os pares, a previsão de classificação pode ser facilmente executado com a Equação 2.4. Essa abordagem também pode ser usada para filtragem colaborativa baseada em itens, aplicando todo o método de redução de dimensionalidade ao transposição de  $R_f$  em vez de  $R_f$ .

A metodologia acima mencionada pode ser vista como uma decomposição de valor singular (DVS) da matriz de classificação  $R_f$ . Vários outros métodos [24, 472] usam componentes principais análise (ACP) em vez de SVD, mas o resultado geral é muito semelhante. No método ACP, a matriz de covariância de  $R_f$  é usada em vez da matriz de similaridade  $R^T f R_f$ . Para dados que é centrado na média ao longo das colunas, os dois métodos são idênticos. Portanto, pode-se subtrair a média de cada coluna a partir de suas entradas e, em seguida, aplicar a abordagem mencionada acima para obter uma representação transformada dos dados. Esta representação transformada é usada para determinar os pares de cada usuário. A centralização na média tem benefícios em termos de redução de viés (veja a próxima seção). Uma abordagem alternativa é primeiro centralizar ao longo de cada linha e então centro médio ao longo de cada coluna. O SVD pode ser aplicado à representação transformada. Esse tipo de abordagem geralmente fornece os resultados mais robustos.

## 2.5.1 Lidando com problemas de viés

Vale ressaltar que a matriz  $R_f$  é derivada da matriz incompleta  $R$ , preenchendo as entradas não especificadas com valores médios ao longo das linhas ou colunas. Essa abordagem provavelmente causará um viés considerável. Para entender a natureza desse viés, considere o exemplo da Tabela 2.3 das avaliações dadas por 12 usuários aos três filmes: O Poderoso Chefão, Gladiador e Nero. Suponhamos que a ACP seja usada para redução de dimensionalidade e, portanto, a matriz de covariância precise ser estimada. Suponhamos que os valores ausentes sejam substituídos pelas médias ao longo das colunas.

Neste caso, as classificações são elaboradas em uma escala de 1 a 7 por um conjunto de 4 usuários para 3 filmes. É visualmente evidente que as correlações entre as classificações dos filmes Gladiador e Nero são extremamente altas, pois as classificações são muito semelhantes nos quatro casos em que são especificadas. A correlação entre O Poderoso Chefão e Gladiador parece ser menos significativa.

No entanto, muitos usuários não especificaram suas avaliações para o Nero. Como a avaliação média do Nero é  $(1 + 7 + 1 + 7)/4 = 4$ , essas avaliações não especificadas são substituídas pelo valor médio de 4. A adição dessas novas entradas reduz significativamente a covariância estimada entre Gladiador e Nero. No entanto, a adição das novas entradas não tem impacto na covariância entre O Poderoso Chefão e Gladiador. Após preencher as avaliações ausentes, as covariâncias pareadas entre os três filmes podem ser estimadas da seguinte forma:

	O Poderoso Chefão	Gladiador	Nero
Padrinho	2,55	4,36	2,18
Gladiador	4,36	9,82	3,27
Nero	2.18	3.27	3.27

De acordo com a estimativa acima mencionada, a covariância entre Godfather e Gladiador é maior do que entre Gladiador e Nero. Isso não parece estar correto porque as classificações na Tabela 2.3 para Gladiador e Nero são idênticas para o caso em que ambos são especificados. Portanto, a correlação entre Gladiador e Nero deveria ser maior. Esse erro é resultado do viés causado pelo preenchimento das entradas não especificadas com a média daquela coluna. Esse tipo de viés pode ser muito significativo em matrizes esparsas porque a maioria das entradas não é especificada. Portanto, métodos precisam ser projetados para reduzir o viés causado pelo uso das classificações médias no lugar das entradas não especificadas. A seguir, exploramos duas possíveis soluções para esse problema.

### 2.5.1.1 Estimativa de Máxima Verossimilhança

O método de reconstrução conceitual [24, 472] propõe o uso de técnicas probabilísticas, como o algoritmo EM, para estimar a matriz de covariância. Um modelo gerativo é assumido para os dados e as entradas especificadas são vistas como os resultados do modelo gerativo. A matriz de covariância pode ser estimada como parte do processo de estimativa dos parâmetros deste modelo gerativo. A seguir, fornecemos uma simplificação desta abordagem. Nesta abordagem simplificada, a estimativa de máxima verossimilhança da matriz de covariância é calculada. A estimativa de máxima verossimilhança da covariância entre cada par de itens é estimada como a covariância entre apenas as entradas especificadas. Em outras palavras, apenas os usuários que têm classificações especificadas para um par particular de itens são usados para estimar a covariância. No caso de não haver usuários em comum entre um par de itens, a covariância é estimada em 0. Usando esta abordagem, a seguinte matriz de covariância é estimada para os dados na Tabela 2.3.

	O Poderoso	Chefão	Gladiador	Nero
Padrinho	2,55	4,36	8	
Gladiador	4,36	9,82	12	
Nero	8	12	12	

Neste caso, torna-se imediatamente evidente que a covariância entre Gladiador e Nero é quase três vezes maior que o Poderoso Chefão e Gladiador. Além disso, o filme O Nero tem mais de três vezes mais variância do que foi estimado originalmente e tem a maior variação nas classificações entre todos os filmes. Enquanto a covariância pareada entre O Poderoso Chefão e Gladiador foram os maiores em comparação com todas as outras covariâncias em pares usando a técnica de preenchimento de média, esse mesmo par agora mostra a menor de todas as covariâncias em pares. Este exemplo sugere que as correções de viés podem ser muito significativas em algumas situações.

Quanto maior a proporção de entradas não especificadas na matriz, maior o viés da técnica de preenchimento de médias. Portanto, a técnica modificada de alavancar apenas o especificado entradas é usado para calcular a matriz de covariância. Embora tal técnica nem sempre seja eficaz, é superior à técnica de preenchimento de média. A matriz de base  $n \times d$  reduzida  $P_d$  é calculado selecionando os autovetores superiores da matriz de covariância resultante.

Para reduzir ainda mais o viés na representação, a matriz incompleta  $R$  pode ser projetado diretamente na matriz reduzida  $P_d$ , em vez de projetar a matriz preenchida  $R_f$  em  $P_d$ . A ideia é calcular a contribuição de cada classificação observada para a projeção em cada vetor latente de  $P_d$  e, em seguida, calcular a média da contribuição sobre o número dessas classificações. Esta contribuição média é calculada da seguinte forma: Seja  $e_i$  a  $i$ -ésima coluna (autovetor) de  $P_d$ , para o qual a  $j$ -ésima entrada é  $e_{ij}$ . Seja  $r_{uj}$  a classificação observada do usuário  $u$  para o item  $j$  em matriz  $R$ . Então, a contribuição do usuário  $u$  para a projeção no vetor latente  $e_i$  é dada por  $r_{uj} e_{ij}$ . Então, se o conjunto  $I_u$  representa os índices das classificações de itens especificadas do usuário  $u$ , a contribuição média  $a_{ui}$  do usuário  $u$  no  $i$ -ésimo vetor latente é a seguinte:

$$a_{ui} = \frac{\sum_{j \in I_u} r_{uj} e_{ij}}{\|e_i\|} \quad (2.17)$$

Este tipo de normalização média é particularmente útil nos casos em que os diferentes usuários especificaram diferentes números de classificações. A matriz  $m \times d$  resultante  $A = [a_{ui}]_{m \times d}$  é usada como representação reduzida da matriz de classificação subjacente. Esta matriz reduzida é usado para calcular a vizinhança do usuário-alvo de forma eficiente para filtragem colaborativa baseada no usuário. Também é possível aplicar a abordagem à transposição da matriz  $R$  e reduzir a dimensionalidade ao longo da dimensão do usuário, em vez da dimensão do item. Essa abordagem é útil para calcular a vizinhança de um item alvo em um sistema baseado em itens filtragem colaborativa. Esta abordagem de usar a representação reduzida para valores ausentes a imputação é discutida em [24, 472].

### 2.5.1.2 Fatoração Matricial Direta de Dados Incompletos

Embora a metodologia acima mencionada possa corrigir o viés na estimativa da covariância até certo ponto, não é completamente eficaz quando o nível de escassez das classificações é alto. Isso ocorre porque a estimativa da matriz de covariância requer um número suficiente de observações classificações para cada par de itens para estimativa robusta. Quando a matriz é esparsa, a covariância as estimativas não serão estatisticamente confiáveis.

Uma abordagem mais direta é usar métodos de fatoração de matrizes. Métodos como fatoração singular decomposição de valores são essencialmente métodos de fatoração de matrizes. Por um momento, suponha

que a matriz de classificações  $m \times n$  R é totalmente especificada. É um fato bem conhecido da álgebra linear [568] que qualquer matriz R (totalmente especificada) pode ser fatorada da seguinte forma:

$$R = Q \tilde{y} P^T \quad (2.18)$$

Aqui, Q é uma matriz  $m \times m$  com colunas contendo os autovetores ortonormais m de  $R^T R$ .

A matriz P é uma matriz  $n \times n$  com colunas contendo os n autovetores ortonormais de  $R^T R$ .  $\tilde{y}$  é uma matriz diagonal  $m \times n$  na qual apenas as entradas diagonais<sup>4</sup> são diferentes de zero e contêm a raiz quadrada dos autovalores diferentes de zero de  $R^T R$  (ou equivalente,  $R^T R$ ). Vale ressaltar que os autovetores de  $R^T R$  e  $R^T R$  não são os mesmos e terão dimensionalidade diferente quando  $m = n$ . No entanto, eles sempre terão o mesmo número de autovalores (diferentes de zero), que são idênticos em valor. Os valores na diagonal de  $\tilde{y}$  também são chamados de valores singulares.

Além disso, é possível fatorar aproximadamente a matriz usando SVD truncado, onde apenas os autovetores correspondentes aos maiores valores singulares  $d \leq \min\{m, n\}$  são usados.

O SVD truncado é calculado da seguinte forma: R

$$\hat{y} Q d \tilde{y} P^T \quad (2.19)_d^T$$

Aqui, Qd,  $\tilde{y} d$  e Pd são matrizes  $m \times d$ ,  $d \times d$  e  $n \times d$ , respectivamente. As matrizes Qd e Pd, respectivamente, contêm os d maiores autovetores de  $R^T R$  e RT R, enquanto a matriz  $\tilde{y} d$  contém as raízes quadradas dos d maiores autovalores de qualquer matriz ao longo de sua diagonal. É digno de nota que a matriz Pd contém os autovetores superiores de RT R, que é a representação de base reduzida necessária para a redução da dimensionalidade. Além disso, a matriz Qd $\tilde{y} d$  contém a representação transformada e reduzida  $m \times d$  da matriz de classificações original na base correspondente a Pd. Pode-se demonstrar que tal fatoração aproximada tem o menor erro quadrático médio das entradas aproximadas em comparação com qualquer outra fatoração de classificação d. Portanto, se pudermos fatorar aproximadamente a matriz de classificações R na forma correspondente à Equação 2.19, ela nos fornecerá a base reduzida, bem como a representação das classificações na base reduzida. O principal problema de usar tal abordagem é que a matriz de classificações não é totalmente especificada. Como resultado, essa fatoração é indefinida. No entanto, é possível reformular a formulação como um problema de otimização, no qual o erro quadrático da fatoração é otimizado apenas sobre as entradas observadas da matriz de classificações. Também é possível resolver explicitamente essa formulação modificada usando técnicas de otimização não linear. Isso resulta em uma representação de dimensão inferior robusta e imparcial. Além disso, tal abordagem pode ser usada para estimar diretamente a matriz de classificações usando a Equação 2.19, uma vez que as matrizes fatoriais reduzidas tenham sido determinadas. Em outras palavras, tais métodos têm uma utilidade direta além dos métodos baseados em vizinhança. Mais detalhes desses modelos de fatores latentes e técnicas de otimização não linear serão discutidos na seção 3.6 do Capítulo 3. O leitor deve consultar esta seção para saber como a representação reduzida pode ser calculada usando formulações de otimização modificadas.

## 2.6 Uma visão de modelagem de regressão dos métodos de vizinhança

---

Uma observação importante sobre os métodos baseados em usuários e em itens é que eles preveem avaliações como funções lineares das avaliações do mesmo item por usuários vizinhos ou do mesmo usuário em itens vizinhos. Para entender este ponto,

<sup>4</sup>Matrizes diagonais são geralmente quadradas. Embora esta matriz não seja quadrada, apenas entradas com índices iguais são diferentes de zero. Esta é uma definição generalizada de uma matriz diagonal.

replicar a função de previsão dos métodos de vizinhança baseados no usuário (cf. Equação 2.4) abaixo:

$$\hat{r}_{uj} = \hat{y}_{ju} + \frac{\sum_{v \neq u} \hat{y}_{vu} P_{uv}(j) \cdot \text{Sim}(u, v) \cdot (r_{vj} - \bar{y}_{vj})}{\sum_{v \neq u} \hat{y}_{vu} P_{uv}(j) \cdot |\text{Sim}(u, v)|} \quad (2.20)$$

Observe que a classificação prevista é uma combinação linear ponderada de outras classificações do mesmo item. A combinação linear foi restrita apenas às classificações do item  $j$  pertencente a usuários com gostos suficientemente semelhantes para atingir o usuário  $u$ . Essa restrição é habilitada com o uso do conjunto de classificação por pares  $P_{uv}(j)$ . Lembre-se da discussão anterior neste capítulo que  $P_{uv}(j)$  é o conjunto de  $k$  usuários mais próximos do usuário alvo  $u$ , que também classificaram o item  $j$ . Observe que se permitíssemos o conjunto  $P_{uv}(j)$  para conter todas as classificações do item  $j$  (e não apenas usuários pares específicos), então o A função de predição torna-se semelhante<sup>5</sup> à da regressão linear [22]. Na regressão linear, as classificações também são previstas como combinações ponderadas de outras classificações, e os pesos (coeficientes) são determinados com o uso de um modelo de otimização. Na abordagem baseada em vizinhança, os coeficientes da função linear são escolhidos de forma heurística com as semelhanças entre usuários, em vez do uso de um modelo de otimização.

Uma observação semelhante se aplica ao caso dos métodos de vizinhança baseados em itens, onde a função de previsão (cf. Equação 2.15) é a seguinte:

$$\text{rota} = \frac{\sum_{t \in Qt(u)} \text{Cosseno ajustado}(j, t) \cdot r_{jt}}{\sum_{t \in Qt(u)} |\text{CossenoAjustado}(j, t)|} \quad (2.21)$$

O conjunto  $Qt(u)$  representa o conjunto dos  $k$  itens mais próximos do item alvo  $t$  que também foram avaliado pelo usuário  $u$ . Neste caso, a classificação de um usuário  $u$  para um item alvo  $t$  é expressa como combinação linear de suas próprias classificações. Como no caso dos métodos baseados no usuário, os coeficientes da combinação linear são definidos heuristicamente com valores de similaridade. Portanto, um modelo baseado no usuário expressa uma classificação prevista como uma combinação linear de classificações no mesmo coluna, enquanto um modelo baseado em itens expressa uma classificação prevista como uma combinação linear de classificações na mesma linha. Deste ponto de vista, os modelos baseados em vizinhança são heurísticos variantes de modelos de regressão linear, nos quais os coeficientes de regressão são definidos heuristicamente para valores de similaridade para itens/usuários relacionados (vizinhos) e para 0 para itens/usuários não relacionados.

Vale ressaltar que o uso de valores de similaridade como pesos de combinação é bastante heurístico e arbitrário. Além disso, os coeficientes não levam em conta as interdependências entre itens. Por exemplo, se um usuário classificou certos conjuntos de itens correlacionados de forma muito semelhante, então os coeficientes associados a esses itens também serão interdependentes. O uso de semelhanças como pesos heurísticos não levam em conta tais interdependências.

Surge a questão de saber se é possível fazer melhor aprendendo os pesos com o uso de uma formulação de otimização. Acontece que se pode derivar regressão análoga baseada em modelos baseados em usuários e em itens. Diversas formulações de otimização diferentes foram propostos na literatura, os quais podem alavancar modelos baseados em usuários, modelos baseados em itens ou uma combinação dos dois. Esses modelos podem ser vistos como generalizações teóricas do modelo heurístico do vizinho mais próximo. A vantagem de tais modelos é que eles são matematicamente melhor fundamentado no contexto de uma formulação de otimização precisa, e os pesos para combinar as classificações podem ser melhor justificados devido à sua otimalidade uma perspectiva de modelagem. A seguir, discutimos uma vizinhança baseada em otimização modelo, que é uma simplificação do trabalho em [309]. Isso também prepara o cenário para a combinação o poder deste modelo com outros modelos de otimização, como a fatoração de matrizes, em seção 3.7 do Capítulo 3.

---

<sup>5</sup>Uma discussão sobre regressão linear é fornecida na seção 4.4.5 do Capítulo 4, mas no contexto de sistemas baseados em conteúdo.

### 2.6.1 Regressão do vizinho mais próximo baseada no usuário

Considere a previsão baseada no usuário da Equação 2.20. Pode-se substituir a (normalizada) coeficiente de similaridade com o parâmetro desconhecido  $w_{user}^{você}$  para modelar a classificação prevista  $\hat{r}_{uj}$  do usuário alvo  $u$  para o item  $j$  da seguinte forma:

$$\hat{r}_{uj} = \hat{y}_u + \frac{\text{similaridade}_{você} \cdot (r_{vj} \hat{y}_{vj})}{v_{vj} P_u(j)} \quad (2.22)$$

Como no caso dos modelos de vizinhança, pode-se usar o coeficiente de correlação de Pearson para definir  $P_u(j)$ . Há, no entanto, uma diferença sutil, mas importante, em termos de como  $P_u(j)$  é definido neste caso. Em modelos baseados em vizinhança,  $P_u(j)$  é o conjunto de  $k$  usuários mais próximos de usuário alvo  $u$ , que especificou classificações para o item  $j$ . Portanto, o tamanho de  $P_u(j)$  é frequentemente exatamente  $k$ , quando pelo menos  $k$  usuários avaliaram o item  $j$ . No caso de métodos de regressão, o conjunto  $P_u(j)$  é definido primeiro determinando os  $k$  pares mais próximos para cada usuário  $e$ , em seguida, restando apenas aqueles para os quais as classificações são observadas. Portanto, o tamanho do conjunto  $P_u(j)$  é frequentemente significativamente menor que  $k$ . Observe que o parâmetro  $k$  precisa ser definido para valores muito maiores na estrutura de regressão em comparação com os modelos de vizinhança devido à sua interpretação diferente.

Intuitivamente, o coeficiente desconhecido  $w_{user}^{você}$  controla a parte da previsão de classificações dado pelo usuário  $u$ , que vem de sua semelhança com o usuário  $v$ , porque esta parte é dada por  $\text{similaridade}_{você} \cdot (r_{vj} \hat{y}_{vj})$ . É possível para  $w_{user}^{você}$  ser diferente de  $w_{user}^{uv}$ . Também é digno de nota que  $w_{user}^{você}$  é definido apenas para os  $k$  valores diferentes de  $v$  (índices do usuário) que estão mais próximos de usuário  $u$  com base no coeficiente de Pearson. Os outros valores de  $w_{user}^{você}$  não são necessários para a função de predição da Equação 2.22 e, portanto, não precisam ser aprendidas. Isso tem o efeito benéfico de reduzir o número de coeficientes de regressão.

Pode-se usar a diferença quadrática agregada entre as classificações previstas  $\hat{r}_{uj}$  (de acordo com a Equação 2.22) e as classificações observadas  $r_{uj}$  para criar uma função objetivo que estima a qualidade de um determinado conjunto de coeficientes. Portanto, pode-se usar o observado classificações na matriz para configurar um problema de otimização de mínimos quadrados sobre os valores desconhecidos de  $w_{user}^{você}$  para minimizar o erro geral. A ideia é prever cada (observado) classificação do usuário  $u$  com seus usuários  $k$  mais próximos em um modelo de regressão formal e, em seguida, medir o erro da previsão. Os erros quadrados podem ser adicionados a todos os itens avaliados pelo usuário  $u$  para criar uma formulação de mínimos quadrados. Portanto, o problema de otimização é configurado para cada usuário alvo  $u$ . Seja  $I_u$  o conjunto de itens que foram avaliados pelo usuário alvo  $u$ . O a função objetivo dos mínimos quadrados para o usuário  $u$  pode ser declarada como a soma dos quadrados de os erros na previsão de cada item em  $I_u$  com os  $k$  vizinhos mais próximos do usuário de forma formal modelo de regressão:

$$\begin{aligned} \text{Minimizar } J_u &= \sum_{j \in I_u} (r_{uj} - \hat{r}_{uj})^2 \\ &= \sum_{j \in I_u} (\hat{y}_{uj} - \hat{y}_{uj})^2 + \sum_{j \in I_u} \text{similaridade}_{você} \cdot (r_{vj} \hat{y}_{vj})^2 \end{aligned}$$

A segunda relação é obtida substituindo a expressão na Equação 2.22 por  $\hat{r}_{uj}$ . Observe que este problema de otimização é formulado separadamente para cada usuário alvo  $u$ . No entanto, é possível somar os valores da função objetivo  $J_u$  sobre diferentes usuários alvo  $u \in \{1 \dots m\}$  sem diferença para a solução ótima. Isso ocorre porque os vários valores de  $J_u$  são expressos em termos de conjuntos mutuamente disjuntos de variáveis de otimização  $w_{user}^{você}$ . Portanto, o

O problema de otimização consolidado é expresso da seguinte forma:

$$\text{Minimizar}_{\substack{u=1 \\ u=1}} \quad J_u = \sum_{j=1}^m \sum_{j=1}^m (\hat{y}_j - \hat{y}_{rui})^2 + w_{user} \cdot \sum_{v=1}^n (\hat{y}_v - \hat{y}_v)^2 \quad (2.23)$$

É possível resolver cada um dos problemas menores de otimização (ou seja, a função objetivo  $J_u$ ) em sua forma decomposta de forma mais eficiente, sem afetar a solução geral. No entanto, a formulação consolidada tem a vantagem de poder ser combinada com outros modelos de otimização, como métodos de fatoração de matrizes (cf. seção 3.7 do Capítulo 3), nos quais tal decomposição não é possível. No entanto, se a regressão linear for usada isoladamente, faz sentido resolver esses problemas em sua forma decomposta.

Tanto a versão consolidada quanto a decomposta dos modelos de otimização são problemas de otimização de mínimos quadrados. Esses métodos podem ser resolvidos com o uso de qualquer solucionador de otimização pronto para uso. Consulte a seção 4.4.5 do Capítulo 4 para uma discussão sobre soluções de forma fechada para problemas de regressão linear. Uma propriedade desejável da maioria desses solucionadores é que eles geralmente possuem regularização incorporada e, portanto, podem evitar o sobreajuste até certo ponto. A ideia básica na regularização é reduzir a complexidade do modelo adicionando  $\frac{1}{2}$  a cada função objetivo (decomposta)  $J_u$ , onde  $\lambda > 0$  termo  $\lambda \sum_{j=1}^m \hat{y}_j \hat{y}_j$  regularização. O termo  $\frac{1}{2}$  penaliza  $\sum_{j=1}^m \hat{y}_j \hat{y}_j$  ( $w_{user}$  é um parâmetro definido pelo usuário que regula o peso do termo de coeficientes grandes e, portanto, reduz os valores absolutos de  $\hat{y}_j \hat{y}_j$  ( $w_{user}$  é um parâmetro definido pelo usuário que regula o peso do termo de coeficientes grandes e, portanto, reduz os valores absolutos de  $\hat{y}_j \hat{y}_j$ ). Coeficientes menores resultam em modelos mais simples e reduzem o sobreajuste.

Entretanto, como discutido abaixo, às vezes não é suficiente usar apenas a regularização para reduzir o overfitting.

### 2.6.1.1 Questões de esparsidade e viés

Um problema com essa abordagem de regressão é que o tamanho do  $P_u(j)$  pode ser muito diferente para o mesmo usuário  $u$  e índices de itens variáveis (denotados por  $j$ ). Isso ocorre devido ao nível extraordinário de escassez inherente às matrizes de classificação. Como resultado, os coeficientes de regressão tornam-se fortemente dependentes do número de usuários pares que classificaram um item específico  $j$  junto com o usuário  $u$ . Por exemplo, considere um cenário em que o usuário-alvo  $u$  classificou Gladiador e Nero. Dos  $k$  vizinhos mais próximos do usuário-alvo  $u$ , apenas um usuário pode classificar o filme Gladiador, enquanto todos os  $k$  podem ter classificado Nero. Como resultado, o coeficiente de regressão  $w_{user}$  do usuário par  $v$  que classificou Gladiador será fortemente influenciado pelo fato de que ele é o único usuário que classificou Gladiador. Isso resultará em sobreajuste porque esse coeficiente de regressão (estatisticamente não confiável) pode adicionar ruído às previsões de classificação de outros filmes.

A ideia básica é alterar a função de predição e assumir que a regressão para o item  $j$  prevê apenas uma fração  $|P_u(j)|$  da classificação do usuário-alvo  $u$  para o item  $j$ . A suposição implícita é que os coeficientes de regressão são baseados em todos os pares do usuário-alvo, e é necessário interpolar informações incompletas como uma fração. Portanto, essa abordagem altera a interpretação dos coeficientes de regressão. Nesse caso, a função de predição da Equação 2.22 é modificada da seguinte forma:

$$\hat{y}_{uj} \cdot \frac{|P_u(j)|}{k} = \hat{y}_u + w_{user} \cdot \sum_{v=1}^n (\hat{y}_v - \hat{y}_v) \quad (2.24)$$

Vários outros ajustes heurísticos são, por vezes, utilizados. Por exemplo, seguindo as ideias de [312], pode-se utilizar um fator de ajuste heurístico de  $|P_u(j)|/k$ . Este fator pode

muitas vezes pode ser simplificado para  $|P_{u(j)}|$  porque fatores constantes são absorvidos pela otimização variáveis. Uma melhoria relacionada é que o deslocamento constante  $\hat{y}_v$  é substituído por um viés variável  $b_u$ , que é aprendida no processo de otimização. A previsão correspondente modelo, incluindo fatores de ajuste heurístico, é o seguinte:

$$\hat{r}_{uj} = \hat{o}_{\text{ônibus}} + \frac{v_j \hat{y}_{P_{u(j)}} \frac{\text{usuário}_{\text{você}}}{\text{usuário}_{\text{você}}} \cdot (r_{vj} - \hat{y}_{\hat{o}_{\text{ônibus}}})}{|P_{u(j)}|} \quad (2.25)$$

Observe que este modelo não é mais linear devido ao termo multiplicativo  $w_{\text{user}}$  entre duas variáveis de otimização. No entanto, é relativamente fácil usar a mesma formulação de mínimos quadrados, como no caso anterior. Além dos vieses do usuário, também é possível incorporar vieses dos itens. Nesse caso, o modelo se torna o seguinte:

$$\hat{r}_{uj} = \hat{o}_{\text{ônibus}} + b_{\text{item}}_{\text{eu}} + \frac{v_j \hat{y}_{P_{u(j)}} \frac{\text{usuário}_{\text{você}}}{\text{usuário}_{\text{você}}} \cdot (r_{vj} - \hat{y}_{\hat{o}_{\text{ônibus}}})}{|P_{u(j)}|} \quad (2.26)$$

Além disso, recomenda-se centralizar toda a matriz de classificação em torno de sua média global subtraindo-se a média de todas as entradas observadas. A média global precisa ser adicionado de volta às previsões. O principal problema com este modelo é computacional. É necessário pré-calcular e armazenar todas as relações usuário-usuário, o que é computacionalmente caro e requer  $O(m^2)$  de espaço sobre  $m$  usuários. Este problema é semelhante ao encontrado em modelos tradicionais baseados em bairros. Tais modelos são adequados em cenários nos quais o espaço dos itens muda rapidamente, mas os usuários são relativamente estáveis ao longo do tempo [312]. Um exemplo é o caso dos sistemas de recomendação de notícias.

## 2.6.2 Regressão do vizinho mais próximo baseada em itens

A abordagem baseada em itens é semelhante à abordagem baseada no usuário, exceto que a regressão aprende e aproveita correlações item-item em vez de correlações usuário-usuário. Considere a previsão baseada em itens da Equação 2.21. Pode-se substituir a similaridade (normalizada) coeficiente  $\text{AdjustedCosine}(j, t)$  com o parâmetro desconhecido  $w_{\text{item}}_{jt}$  para modelar a classificação previsão do usuário  $u$  para o item alvo  $t$ :

$$\text{rota} = \frac{\text{branco}}{\sum_j \hat{y}_{Q_t(u)}} \cdot r_{uj} \quad (2.27)$$

Os itens mais próximos em  $Q_t(u)$  podem ser determinados usando o cosseno ajustado, como em itens baseados métodos de vizinhança. O conjunto  $Q_t(u)$  representa o subconjunto dos  $k$  vizinhos mais próximos de o item alvo  $t$ , para o qual o usuário  $u$  forneceu classificações. Esta forma de definir  $Q_t(u)$  é sutilmente diferente dos métodos tradicionais baseados em vizinhança, porque o tamanho do conjunto  $Q_t(u)$  pode ser significativamente menor que  $k$ . Nos métodos tradicionais de vizinhança, determina-se o itens  $k$  mais próximos do item alvo  $t$ , para o qual o usuário  $u$  especificou classificações e, portanto, o tamanho do conjunto de vizinhança é frequentemente exatamente  $k$ . Essa mudança é necessária para poder implementar efetivamente o método baseado em regressão.

Intuitivamente, o coeficiente desconhecido  $w_{\text{item}}_{jt}$  controla a parcela da classificação do item  $t$ , que vem de sua semelhança com o item  $j$ , porque esta porção é dada por  $w_{\text{item}}_{jt} \cdot r_{uj}$ . O erro de previsão da Equação 2.27 deve ser minimizado para garantir a previsão mais robusta modelo. Pode-se usar as classificações conhecidas na matriz para configurar uma otimização de mínimos quadrados problema sobre os valores desconhecidos de  $w_{\text{item}}_{jt}$  a fim de minimizar o erro geral. A ideia é prever cada classificação (observada) do item alvo  $t$  com seus  $k$  itens mais próximos e então,

crie uma expressão para o erro dos mínimos quadrados. O problema de otimização é configurado para cada item alvo  $t$ . Seja  $U_t$  o conjunto de usuários que avaliaram o item alvo  $t$ . Os mínimos quadrados a função objetivo para o tésimo item pode ser expressa como a soma dos quadrados dos erros em prevendo cada classificação especificada em  $U_t$ :

$$\begin{aligned} \text{Minimizar } J_t &= \sum_{u \in U_t} (\text{rotina } \hat{y}_t - r_{ut})^2 \\ &= \sum_{u \in U_t} (\text{rotina } \hat{y}_{jt} - \text{branco}_{jt} \cdot r_{uj})^2 \end{aligned}$$

Observe que este problema de otimização é formulado separadamente para cada item alvo  $t$ . No entanto, pode-se somar os termos sobre vários valores do item alvo  $t$  sem nenhuma diferença para a solução de otimização, porque os coeficientes desconhecidos com eles  $j_t$  nos vários objetivos funções não se sobrepõem em diferentes valores do item alvo  $t \in \{1 \dots n\}$ . Portanto, temos a seguinte formulação consolidada:

$$\text{Minimizar } \sum_{t=1}^n (\text{rotina } \hat{y}_{jt} - \text{branco}_{jt} \cdot r_{uj})^2 \quad (2.28)$$

Este é um problema de regressão de mínimos quadrados e pode ser resolvido com o uso de qualquer solucionador disponível no mercado. Além disso, também é possível resolver cada um dos problemas de otimização menores (ou seja, função objetivo  $J_t$ ) em sua forma decomposta de forma mais eficiente sem afetar o geral solução. No entanto, a formulação consolidada tem a vantagem de poder ser combinada com outros modelos de otimização, como métodos de fatoração de matrizes (cf. seção 3.7 de Capítulo 3). Tal como no caso dos métodos baseados no utilizador, existem desafios significativos associados a o problema do overfitting. Pode-se adicionar o termo de regularização  $\lambda \sum_{u \in U} \sum_{j \in Q(u)} |j|$  para a função objetivo  $J_t$ .

Conforme discutido na seção 2.6.1.1 para o caso do modelo baseado no usuário, pode-se incorporar fatores de ajuste e variáveis de viés para melhorar o desempenho. Por exemplo, o modelo baseado no usuário O modelo de previsão da Equação 2.26 assume a seguinte forma no modelo item a item:

$$r_{ut} = \text{ônibus} + \text{bitem} + \frac{\sum_{j \in Q(u)} \text{branco}_{jt} \cdot (r_{uj} - \text{ônibus})}{|Q(u)|} \quad (2.29)$$

Além disso, presume-se que as classificações sejam centradas em torno da média global da matriz de classificações inteira. Portanto, a média global é subtraída de cada uma das classificações antes de construir o modelo. Todas as previsões são realizadas nas classificações centralizadas e, em seguida, a média global é adicionada novamente a cada previsão. Em algumas variações do modelo, o viés termos buser  $+ \text{bitem}$  entre parênteses são substituídos por um termo constante consolidado  $B_{ut}$ .

Este termo constante é derivado usando uma abordagem não personalizada descrita na seção 3.7.1

do Capítulo 3. O modelo de previsão resultante é o seguinte:

$$r_{ut} = \text{ônibus} + \text{bitem} + \frac{\sum_{j \in Q(u)} \text{branco}_{jt} \cdot (r_{uj} - B_{ut})}{|Q(u)|} \quad (2.30)$$

Um modelo de otimização de mínimos quadrados é formulado e uma abordagem de descida de gradiente é usada para resolver os parâmetros de otimização. Este é precisamente o modelo usado em [309]. O as etapas de descida de gradiente resultantes são discutidas na seção 3.7.2 do Capítulo 3. O usuário-usuário é conhecido por ter um desempenho ligeiramente melhor do que o modelo item-item [312]. No entanto, o modelo baseado em itens é muito mais eficiente em termos computacionais e de espaço em ambientes onde o número de itens é muito menor que o número de usuários.

### 2.6.3 Combinando métodos baseados em usuários e em itens

É natural combinar os modelos baseados em usuários e itens em uma estrutura de regressão unificada [312]. Portanto, uma classificação é prevista com base em sua relação com usuários e itens semelhantes. Isso é alcançado combinando as ideias das Equações 2.26 e 2.30 da seguinte forma:

$$\hat{r}_{uj} = b_{user} + \frac{w_{user} \cdot (rv_j - \bar{v}_j)}{|P_u(j)|} + \frac{\text{item} \cdot (r_{uj} - \bar{r}_{uj})}{|Q_t(u)|} \quad (2.31)$$

Como nos casos anteriores, assume-se que a matriz de classificações é centrada em torno de sua média global. Uma formulação de otimização de mínimos quadrados semelhante pode ser usada, na qual o erro quadrático sobre todas as entradas observadas é minimizado. Nesse caso, não é mais possível decompor o problema de otimização em subproblemas independentes. Portanto, um único modelo de otimização de mínimos quadrados é construído sobre todas as entradas observadas na matriz de classificações. Como nos casos anteriores, a abordagem gradiente descendente pode ser usada. Foi relatado em [312] que a fusão dos modelos baseados no usuário e nos itens geralmente tem melhor desempenho do que os modelos individuais.

### 2.6.4 Interpolação conjunta com ponderação de similaridade

O método em [72] usa uma ideia diferente para configurar o modelo baseado em vizinhança conjunta.

A ideia básica é prever cada avaliação do usuário-alvo  $u$  com o modelo baseado em usuários da Equação 2.22. Então, em vez de compará-lo com o valor observado do mesmo item, comparamos com as avaliações observadas de outros itens daquele usuário.

Seja  $S$  o conjunto de todos os pares de combinações usuário-item na matriz de classificações que foram observados:

$$S = \{(u, t) : \text{rotina é observada}\} \quad (2.32)$$

Estabelecemos uma função objetivo que é penalizada quando a classificação prevista  $\hat{r}_{uj}$  de um item  $j$  está muito distante da classificação observada dada a um item semelhante  $s$  pelo mesmo usuário alvo  $u$ .

Em outras palavras, a função objetivo para o usuário alvo  $u$  é definida da seguinte forma:

$$\begin{aligned} \text{Minimizar} & \quad \text{Cosseno ajustado}(j, s) \cdot (r_{us} - \hat{r}_{uj})^2 \\ & \quad s: (u, s) \in S, j: s \\ & = \quad \text{Cosseno ajustado}(j, s) \cdot s: \quad \bar{y}_j \bar{y}_s u + w_{user} \cdot (r_{vj} - \bar{v}_j) \bar{y}_s \\ & \quad (u, s) \in S, j: s \quad \bar{y}_j \bar{y}_s \end{aligned}$$

A regularização pode ser adicionada à função objetivo para reduzir o sobreajuste. Aqui,  $P_u(j)$  é definido como os  $k$  usuários mais próximos do usuário-alvo  $u$ , que também avaliaram o item  $j$ . Portanto, a definição convencional de  $P_u(j)$ , conforme usada em modelos baseados em vizinhança, é aproveitada neste contexto.

Ao usar o cosseno ajustado como um fator multiplicativo de cada termo individual na função objetivo, a abordagem força as avaliações do usuário-alvo de itens semelhantes a serem mais semelhantes também. Vale ressaltar que as similaridades entre usuários e itens são usadas nessa abordagem, mas de maneiras diferentes:

- As similaridades item-item são usadas como fatores multiplicativos dos termos na função objetivo para forçar as classificações previstas a serem mais semelhantes às classificações observadas de itens semelhantes.

2. As semelhanças entre usuários são usadas para prever as classificações, restringindo a regressão coeficientes de sion para o grupo de pares relevante  $P_u(j)$  do usuário alvo  $u$ .

Embora também seja possível, em princípio, alternar as funções de usuários e itens para configurar um modelo diferente, [72] afirma que o modelo resultante não é tão eficaz quanto o discutido acima. Este modelo pode ser resolvido com qualquer solucionador de mínimos quadrados disponível no mercado. Vários métodos também são discutidos em [72] para lidar com a escassez.

## 2.6.5 Modelos Lineares Esparsos (SLIM)

Um método interessante, baseado na regressão item-item da seção 2.6.2, é proposto em [455]. Essa família de modelos é denominada modelos lineares esparsos porque incentiva a esparsidade nos coeficientes de regressão com o uso de métodos de regularização. Diferentemente dos métodos em [72, 309], esses métodos trabalham com valores de classificação não negativos. Portanto, diferentemente das técnicas das seções anteriores, não será assumido que a matriz de classificações é centrada na média. Isso ocorre porque a centralização na média criará automaticamente classificações negativas, correspondentes a desgostos. No entanto, a abordagem foi projetada para trabalhar com classificações não negativas, nas quais não há mecanismo para especificar desgostos. De um ponto de vista prático, a abordagem é mais apropriada6 para matrizes de feedback implícitas (por exemplo, dados de cliques ou dados de vendas), onde apenas preferências positivas são expressas por meio de ações do usuário. Além disso, como é comum em configurações de feedback implícito, os valores ausentes são tratados como 0s para fins de treinamento na formulação de otimização. No entanto, o modelo de otimização pode eventualmente prever que alguns desses valores sejam altamente positivos, e tais combinações usuário-item são excelentes candidatas para recomendação. Portanto, a abordagem classifica os itens com base nos erros de previsão nas entradas de treinamento que foram definidas como 0.

Diferentemente da técnica da seção 2.6.2, esses métodos não restringem os coeficientes de regressão apenas à vizinhança do item alvo  $t$ . Então, a função de predição no SLIM é expressa da seguinte forma:

$$\text{rota} = \sum_{j=1}^n w_{item} \cdot r_{uj} \hat{y}_{u \{1 \dots m\}, \hat{y}_t \{1 \dots n\} j} \quad (2.33)$$

Observe a relação com a Equação 2.27 , na qual apenas a vizinhança do item-alvo é usada para construir a regressão. É importante excluir o próprio item-alvo do lado direito para evitar sobreajuste. Isso pode ser alcançado exigindo que a restrição seja 0. Seja  $R^* = [r_{uj}]$  a matriz de classificações previstas e seja  $W_{item}$  = que  $w_{item}$   $[w_{item}]$  a matriz de podemos<sub>tt</sub> regressão item-item. Portanto, se assumirmos que os elementos diagonais de  $W_{item}$  são restritos a 0, empilharas instâncias da Equação 2.33 sobre diferentes usuários e itens-alvo para criar a seguinte função de previsão baseada em matriz:

$$R^* = RW_{item}$$

$$\text{Diagonal}(W_{item})=0$$

Portanto, o objetivo principal é minimizar a norma de Frobenius  $\|R - RW_{item}\|_F^2$  juntamente com alguns termos de regularização. Esta função objetivo é disjunta em diferentes colunas de  $W$  (ou seja, itens alvo na regressão). Portanto, pode-se resolver cada problema de otimização (por

<sup>6</sup>A abordagem pode ser adaptada a matrizes de classificação arbitrárias. No entanto, as principais vantagens da abordagem são percebidas para matrizes de classificação não negativas.

um determinado valor do item alvo  $t$ ) de forma independente, ao definir  $w_{item}$  para 0. Para criar uma regressão de soma de partes mais interpretável, os vetores de peso são restringidos a não ser negativo. Portanto, a função objetivo para o item alvo  $t$  pode ser expressa como segue:

$$\begin{aligned} \text{Minimizar } J_s &= \sum_{u=1}^m (\text{rotina } \hat{y} \cdot r^u) + \hat{y} \cdot \sum_{j=1}^n (\text{compr}_j)^2 + \hat{y}_1 \cdot \sum_{j=1}^n |w_{item}| \\ &= \sum_{u=1}^m (\text{rotina } \hat{y} \cdot \text{branco}_j \cdot r_{uj}) + \hat{y} \cdot \sum_{j=1}^n (\text{compr}_j)^2 + \hat{y}_1 \cdot \sum_{j=1}^n |w_{item}| \end{aligned}$$

sujeito a:

$$\begin{aligned} \text{branco}_j &\geq 0 \quad \forall j \in \{1 \dots n\} \\ w_{item} &= 0 \end{aligned}$$

Os dois últimos termos da função objetivo correspondem ao regularizador de rede elástica, que combina regularização L1 e L2. Pode ser demonstrado [242] que o componente de regularização L1 leva a soluções esparsas para os pesos  $w_{jt}$ , o que significa que a maioria dos coeficientes  $w_{jt}$  têm valores zero. A escassez garante que cada classificação prevista pode ser expressa como um combinação linear mais interpretável das classificações de um pequeno número de outros itens relacionados. Além disso, como os pesos não são negativos, os itens correspondentes estão positivamente relacionados de uma forma altamente interpretável em termos do nível específico de impacto de cada classificação em uma regressão. O problema de otimização é resolvido usando o método de descida de coordenadas, embora qualquer solucionador pronto para uso possa ser usado em princípio. Uma série de técnicas mais rápidas são discutidas em [347]. A técnica também pode ser hibridizada [456] com informações secundárias (cf. seção 6.8.1 do Capítulo 6).

É evidente que este modelo está intimamente relacionado com a regressão baseada na vizinhança modelos discutidos nas seções anteriores. As principais diferenças do modelo SLIM em relação ao modelo de regressão linear em [309] é o seguinte:

- O método em [309] restringe os coeficientes diferentes de zero para cada alvo a no máximo  $k$  itens mais semelhantes. O método SLIM pode usar até  $|U|$  coeficientes diferentes de zero. Para exemplo, se um item for avaliado por todos os usuários, todos os coeficientes serão usados. No entanto, o valor do  $w_{item}$  é definido como 0 para evitar overfitting. Além disso, o método SLIM força escassez usando o regularizador de rede elástica, enquanto o método em [309] pré-seleciona os pesos com base no cálculo explícito da vizinhança. Em outras palavras, o trabalho em [309] usa uma abordagem heurística para seleção de recursos, enquanto a abordagem SLIM usa uma abordagem de aprendizagem (regularização) para seleção de recursos.
- O método SLIM é projetado principalmente para conjuntos de dados de feedback implícito (por exemplo, compra um item ou cliques do cliente), em vez de classificações explícitas. Nesses casos, as classificações são tipicamente unário, em que as ações do cliente são indicações de preferência positiva, mas o ato de não comprar ou clicar em um item não indica necessariamente uma reação negativa preferência. A abordagem também pode ser usada para casos em que as "classificações" são valores arbitrários que indicam apenas preferências positivas (por exemplo, quantidade de produto comprada). Note-se que tais cenários são geralmente propícios a métodos de regressão que impõem não negatividade nos coeficientes do modelo. Como você aprenderá no Capítulo 3, isso A observação também é verdadeira para outros modelos, como a fatoração de matrizes. Por exemplo, a fatoração de matriz não negativa é útil principalmente para conjuntos de dados de feedback implícito, mas não é tão útil para classificações arbitrárias. Isso ocorre, em parte, porque a decomposição não negativa da soma das partes perde sua interpretabilidade quando uma classificação indica ou uma classificação de "gostei" ou "não gostei". Por exemplo, duas classificações de "não gostei" não equivalem a uma classificação de "gostei". avaliação.

3. Os coeficientes de regressão em [309] podem ser positivos ou negativos. Por outro lado, os coeficientes no SLIM são limitados a serem não negativos. Isso ocorre porque o método SLIM é projetado principalmente para o ambiente de feedback implícito. A não negatividade costuma ser mais intuitiva nesses ambientes e os resultados são mais interpretáveis. De fato, em alguns casos, impor a não negatividade pode melhorar<sup>7</sup> a precisão. No entanto, alguns resultados experimentais limitados foram apresentados [347], sugerindo que a remoção das restrições de não negatividade proporciona um desempenho superior.
4. Embora o método SLIM também proponha um modelo de predição para as classificações (de acordo com a Equação 2.33), o uso final dos valores preditos é para classificar os itens na ordem do valor predito. Observe que a abordagem é geralmente usada para conjuntos de dados com classificações unárias e, portanto, faz sentido usar os valores preditos para classificar os itens, em vez de prever as classificações. Uma maneira alternativa de interpretar os valores preditos é que cada um deles pode ser visto como o erro de substituir uma classificação não negativa por 0 na matriz de classificações. Quanto maior o erro, maior será o valor predito da classificação. Portanto, os itens podem ser classificados na ordem do valor predito.
5. Ao contrário do trabalho em [309], o método SLIM não ajusta explicitamente o número variável de classificações especificadas com fatores de ajuste heurísticos. Por exemplo, o lado direito da Equação 2.29 usa um fator de ajuste de  $|Qt(u)|$  no denominador. Por outro lado, esse fator de ajuste não é utilizado no método SLIM. A questão do ajuste é menos urgente no caso de conjuntos de dados unários, nos quais a presença de um item geralmente é a única informação disponível. Nesses casos, substituir valores ausentes por 0s é uma prática comum, e o viés de fazê-lo é muito menor do que no caso em que as classificações indicam níveis variados de gostos ou desgostos.

Portanto, os modelos compartilham uma série de semelhanças conceituais, embora haja algumas diferenças no nível detalhado.

## 2.7 Modelos de Grafos para Métodos Baseados em Vizinhança

A escassez de classificações observadas causa um grande problema no cálculo da similaridade em métodos baseados em vizinhança. Diversos modelos de grafos são utilizados para definir similaridade em métodos baseados em vizinhança, com o uso de técnicas de transitividade estrutural ou de classificação. Os grafos são uma abstração poderosa que possibilita o uso de diversas ferramentas algorítmicas do domínio da rede. Os grafos fornecem uma representação estrutural das relações entre vários usuários e/ou itens. Os grafos podem ser construídos sobre os usuários, sobre os itens ou sobre ambos. Esses diferentes tipos de grafos resultam em uma ampla variedade de algoritmos, que utilizam

---

<sup>7</sup> Vale ressaltar que a imposição de uma restrição adicional, como a não negatividade, sempre reduz a qualidade da solução ótima nas entradas observadas. Por outro lado, a imposição de restrições aumenta o viés do modelo e reduz a variância do modelo, o que pode reduzir o sobreajuste nas entradas não observadas. De fato, quando dois modelos intimamente relacionados apresentam desempenhos relativos contraditórios nas entradas observadas e não observadas, respectivamente, isso quase sempre é resultado de níveis diferenciados de sobreajuste nos dois casos. Você aprenderá mais sobre o trade-off viés-variancia no Capítulo 6. Em geral, é mais confiável prever classificações de itens com relações item-item positivas do que com relações negativas. A restrição de não negatividade se baseia nessa observação. A incorporação de vieses do modelo na forma de tais restrições naturais é particularmente útil para conjuntos de dados menores.

Métodos de caminhada aleatória ou de caminho mais curto para recomendação. A seguir, descreveremos os algoritmos usados para realizar recomendações com vários tipos de representações gráficas de matrizes de classificação.

## 2.7.1 Gráficos de usuário-item

É possível usar medidas estruturais no gráfico usuário-item, em vez do coeficiente de correlação de Pearson, para definir vizinhanças. Essa abordagem é mais eficaz para matrizes de classificação esparsas, pois é possível usar a transitividade estrutural das arestas para o processo de recomendação.

O grafo usuário-item é definido como um grafo não direcionado e bipartido  $G = (Nu \cup Ni, A)$ , onde  $Nu$  é o conjunto de nós que representam usuários e  $Ni$  é o conjunto de nós que representam itens.

Todas as arestas no grafo existem apenas entre usuários e itens. Uma aresta não direcionada existe em  $A$  entre um usuário  $i$  e um item  $j$ , se e somente se o usuário  $i$  classificou o item  $j$ . Portanto, o número de arestas é igual ao número de entradas observadas na matriz de utilidade. Por exemplo, o grafo usuário-item para a matriz de classificações da Figura 2.3(a) é ilustrado na Figura 2.3(b). A principal vantagem dos métodos baseados em grafos é que dois usuários não precisam ter classificado muitos dos mesmos itens para serem considerados vizinhos, desde que existam muitos caminhos curtos entre os dois usuários. Portanto, essa definição permite a construção de vizinhanças com a noção de conectividade indireta entre os nós. Obviamente, se dois usuários classificaram muitos itens comuns, essa definição também os considerará vizinhos próximos. Portanto, a abordagem baseada em grafos fornece uma maneira diferente de definir vizinhanças, o que pode ser útil em ambientes esparsos.

A noção de conectividade indireta é alcançada com o uso de definições baseadas em caminhos ou caminhadas. Alguns métodos comuns para atingir esse objetivo incluem o uso de medidas de caminhada aleatória ou a medida de Katz, discutida na seção 2.7.1.2. Ambas as medidas estão intimamente relacionadas ao problema de predição de links na análise de redes sociais (cf. seção 10.4 do Capítulo 10) e demonstram o fato de que modelos gráficos de sistemas de recomendação conectam o problema de predição de links ao problema de recomendação tradicional. A seguir, discutimos diferentes maneiras de definir vizinhanças na representação gráfica.

### 2.7.1.1 Definindo Vizinhanças com Passeios Aleatórios

A vizinhança de um usuário é definida pelo conjunto de usuários que são encontrados frequentemente em uma caminhada aleatória a partir desse usuário. Como a frequência esperada de tais caminhadas aleatórias pode ser medida? A resposta para esse problema está intimamente relacionada aos métodos de caminhada aleatória, que são usados frequentemente em aplicativos de classificação da Web. Pode-se usar o método PageRank personalizado ou o método SimRank (cf. Capítulo 10) para determinar os  $k$  usuários mais semelhantes a um determinado usuário para filtragem colaborativa baseada em usuário. Da mesma forma, pode-se usar esse método para determinar os  $k$  itens mais semelhantes a um determinado item, iniciando a caminhada aleatória em um determinado item. Essa abordagem é útil para filtragem colaborativa baseada em item. As outras etapas da filtragem colaborativa baseada em usuário e da filtragem colaborativa baseada em item permanecem as mesmas.

Por que essa abordagem é mais eficaz para matrizes esparsas? No caso do coeficiente de correlação de Pearson, dois usuários precisam estar conectados diretamente a um conjunto de itens comuns para que a vizinhança seja definida de forma significativa. Em grafos esparsos de usuários-itens, essa conectividade direta pode não existir para muitos nós. Por outro lado, um método de passeio aleatório também considera a conectividade indireta, pois um passeio de um nó a outro pode usar qualquer número de passos. Portanto, desde que grandes porções dos grafos de usuários-itens estejam conectadas,

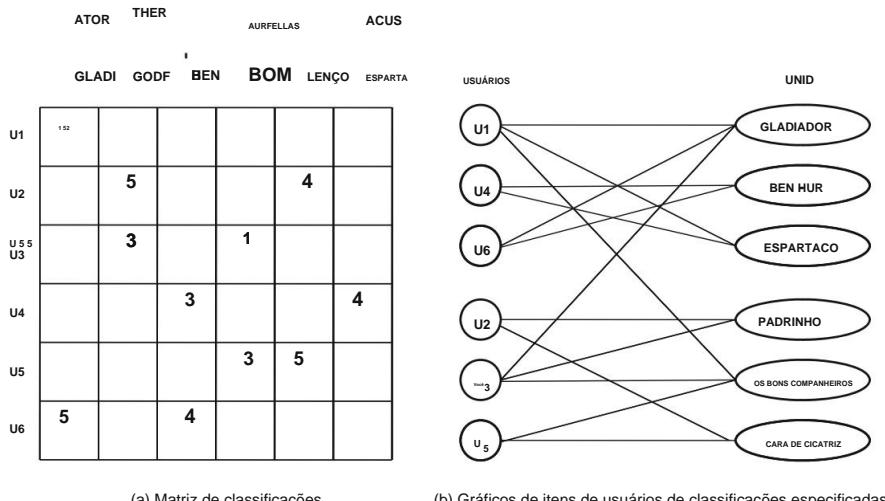


Figura 2.3: Uma matriz de classificações e um gráfico de item-usuário correspondente

é sempre possível definir bairros de forma significativa. Esses gráficos de itens de usuário também podem ser usado para prever classificações diretamente com o uso de uma variedade de modelos. Esses métodos relacionados será discutido na seção 10.2.3.3 do Capítulo 10.

#### 2.7.1.2 Definindo bairros com a medida de Katz

Em vez de usar uma medida probabilística, como caminhadas aleatórias, é possível usar a número ponderado de caminhadas entre um par de nós para determinar a afinidade entre eles. O peso de cada caminhada é um fator de desconto em  $(0, 1)$ , que normalmente é uma diminuição função do seu comprimento. O número ponderado de caminhadas entre um par de nós é referido como a medida de Katz. O número ponderado de caminhadas entre um par de nós é frequentemente usado como uma medida de previsão de link. A intuição é que se dois usuários pertencem ao mesmo bairro com base na conectividade baseada em caminhadas, então há uma propensão para que um link seja formado entre eles no gráfico usuário-item. O nível específico de propensão é medido com o número de caminhadas (com desconto) entre eles.

**Definição 2.7.1 (Medida de Katz)** Seja  $n(t)$  seja o número de caminhadas de comprimento  $t$  entre nós  $i$  e  $j$ . Então, para um parâmetro definido pelo usuário  $\gamma < 1$ , a medida de Katz entre os nós  $i$  e  $j$  são definidos da seguinte forma:

$$\text{Katz}(i, j) = \sum_{t=1}^{\gamma} \gamma^t \cdot n(t) \quad (2.34)$$

O valor de  $\gamma$  é um fator de desconto que minimiza caminhadas de maior distância. Para pequenas valores suficientes de  $\gamma$ , a soma infinita da Equação 2.34 convergirá.

Seja  $K$  a matriz  $m \times m$  dos coeficientes de Katz entre pares de usuários. Se  $A$  é o matriz de adjacência simétrica de uma rede não direcionada, então o coeficiente de Katz em pares a matriz  $K$  pode ser calculada da seguinte forma:

$$K = \sum_{i=1}^{\gamma} (\gamma A)^{-1} = (I - \gamma A)^{-1} \quad (2.35)$$

O valor de  $\gamma$  deve ser sempre selecionado como menor que o inverso do maior autovalor de  $A$  para garantir a convergência da soma infinita. A medida de Katz está intimamente relacionada aos núcleos de difusão em gráficos. De fato, vários métodos de recomendação colaborativa utilizam diretamente núcleos de difusão para fazer recomendações [205].

Uma versão ponderada da medida pode ser calculada substituindo  $A$  pela matriz de pesos do grafo. Isso pode ser útil nos casos em que se deseja ponderar as arestas do grafo usuário-item com a classificação correspondente. Os  $k$  nós superiores com as maiores medidas de Katz em relação ao nó-alvo são isolados como sua vizinhança. Uma vez determinada a vizinhança, ela é usada para realizar a previsão de acordo com a Equação 2.4. Muitas variações deste princípio básico são usadas para fazer recomendações:

1. É possível usar um limite para o comprimento máximo do caminho na Equação 2.34. Isso ocorre porque comprimentos de caminho maiores geralmente se tornam ruidosos para o processo de previsão. No entanto, devido ao uso do fator de desconto  $\gamma$ , o impacto de caminhos longos na medida é geralmente limitado.
2. Na discussão acima, a medida de Katz é usada apenas para determinar as vizinhanças dos usuários. Portanto, a medida de Katz é usada para calcular a afinidade entre pares de usuários. Após a determinação da vizinhança de um usuário, ela é usada para fazer previsões da mesma forma que qualquer outro método baseado em vizinhança.

No entanto, uma maneira diferente de realizar a previsão diretamente, sem usar métodos de vizinhança, seria medir a afinidade entre usuários e itens. A medida de Katz pode ser usada para calcular essas afinidades. Nesses casos, os links são ponderados com classificações, e o problema se reduz ao de prever links entre usuários e itens. Esses métodos serão discutidos com mais detalhes na seção 10.4.6 do Capítulo 10.

As notas bibliográficas contêm uma série de referências a vários métodos baseados em caminhos.

## 2.7.2 Gráficos de usuário-usuário

Em grafos usuário-item, a conectividade usuário-usuário é definida por um número par de saltos no grafo usuário-item. Em vez de construir grafos usuário-item, pode-se criar diretamente grafos usuário-usuário com base na conectividade de dois saltos entre usuários. A vantagem dos grafos usuário-usuário sobre os grafos usuário-item é que as arestas do grafo são mais informativas no primeiro. Isso ocorre porque a conectividade de dois saltos pode levar diretamente em consideração o número e a similaridade de itens comuns entre os dois usuários, ao criar as arestas. Essas noções, chamadas de horting e previsibilidade, serão discutidas um pouco mais adiante. O algoritmo usa a noção de horting para quantificar o número de classificações mutuamente especificadas entre dois usuários (nós), enquanto usa a noção de previsibilidade para quantificar o nível de similaridade entre essas classificações comuns.

O grafo usuário-usuário é construído da seguinte forma. Cada nó  $u$  corresponde a um dos  $m$  usuários na matriz usuário-item  $m \times n$ . Seja  $I_u$  o conjunto de itens para os quais as avaliações foram especificadas pelo usuário  $u$  e seja  $I_v$  o conjunto de itens para os quais as avaliações foram especificadas pelo usuário  $v$ . As arestas são definidas neste grafo com a noção de horting. Horting é uma relação assimétrica entre usuários, definida com base no fato de terem avaliado itens semelhantes.

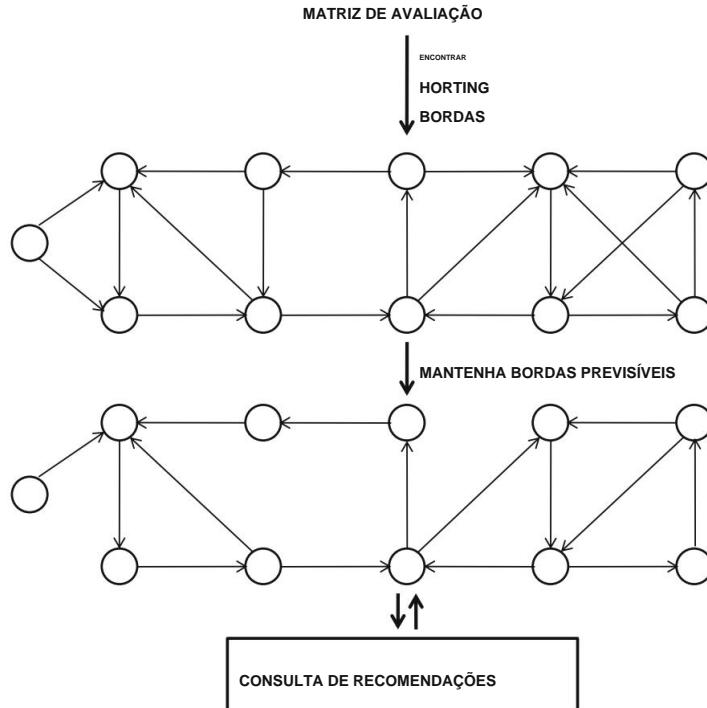


Figura 2.4: A abordagem de previsibilidade usuário-usuário

Definição 2.7.2 (Horting) Diz-se que um usuário  $u$  faz um short no usuário  $v$  no nível  $(F, G)$ , se qualquer uma das seguintes condições for verdadeira:

$$\begin{aligned} \|u - v\| &\leq F \\ \|u - v\| / \|u\| &\leq G \end{aligned}$$

Aqui,  $F$  e  $G$  são parâmetros do algoritmo. Observe que basta que uma das duas condições mencionadas seja atendida para que o usuário  $u$  faça um horting do usuário  $v$ . A noção de horting é usada para definir melhor a previsibilidade.

Definição 2.7.3 (Previsibilidade) O usuário  $v$  prevê o usuário  $u$ , se  $u$  for menor que  $v$  e existir uma função de transformação linear  $f(\cdot)$  tal que o seguinte seja verdadeiro:

$$\frac{\sum_{k=1}^n |u_k - f(v_k)|}{\|u - v\|} \leq U$$

Aqui,  $U$  é outro parâmetro do algoritmo. Vale ressaltar que a distância  $\sum_{k=1}^n |u_k - f(v_k)|$  é a distância de Manhattan entre as avaliações do usuário  $u$  e as avaliações transformadas do usuário  $v$  é uma variante  $\sum_{k=1}^n |u_k - v_k|$  da distância de Manhattan entre suas avaliações especificadas em comum. A principal diferença em relação à distância de Manhattan é que a distância é normalizada pelo número de avaliações mutuamente especificadas entre os dois usuários. Essa distância também é chamada de distância segmental de Manhattan.

As direções de horing e previsibilidade são opostas. Em outras palavras, para que o usuário v preveja o usuário u, u deve encurtar v. Um grafo direcionado G é definido, no qual existe uma aresta de u a v, se v prediz u. Este grafo é chamado de grafo de previsibilidade usuário-usuário. Cada aresta neste grafo corresponde a uma transformação linear, conforme discutido na Definição 2.7.3. A transformação linear define uma predição, onde a classificação na cabeça da aresta pode ser usada para prever a classificação na cauda da aresta. Além disso, assume-se que é possível aplicar essas transformações lineares de forma transitiva sobre um caminho direcionado para prever a classificação da origem do caminho a partir da classificação no destino do caminho.

Então, a classificação de um usuário alvo u para um item k é calculada determinando todos os caminhos mais curtos direcionados do usuário u para todos os outros usuários que classificaram o item k. Considere um caminho direcionado de comprimento r do usuário u para um usuário v que classificou o item k. Deixe  $f_1 \dots f_r$  representar a sequência de transformações lineares ao longo do caminho direcionado começando do nó u para este (v) da classificação do usuário alvo u para o item k (com base apenas no usuário v). Então, a previsão de lineares ao longo deste caminho do usuário u para v,  $\hat{r}^k(v)$  é dada aplicando a composição dos r mapeamentos para a classificação  $r^k(v)$  do usuário v no item k:

$$\hat{r}^k(v) = (f_1 \circ f_2 \dots \circ f_r)(r^k(v)) \quad (2.36)$$

A previsão de classificação  $\hat{r}^k(v)$  contém o subscrito  $v$  porque é baseada apenas na classificação do usuário v. Portanto, a previsão de classificação final  $\hat{r}^k(u)$  é computada pela média do valor de  $\hat{r}^k(v)$  sobre todos os usuários v que classificaram o item k, dentro de uma distância limite D do usuário alvo do Reino Unido u.

Dado um usuário alvo (nó) u, é necessário apenas determinar os caminhos direcionados deste usuário para outros usuários que avaliaram o item em questão. O caminho mais curto pode ser determinado com o uso de um algoritmo de largura em primeiro lugar, que é bastante eficiente. Outro detalhe importante é que um limite é imposto ao comprimento máximo do caminho que é utilizável para predição. Se nenhum usuário que tenha avaliado o item k for encontrado dentro de um comprimento limite D do nó alvo u, então o algoritmo termina com falha. Em outras palavras, a classificação do usuário alvo u para o item k simplesmente não pode ser determinada de forma robusta com a matriz de classificações disponível. É importante impor tais limites para melhorar a eficiência e também porque a transformação linear ao longo de comprimentos de caminho muito longos pode levar ao aumento da distorção na predição de classificação.

A abordagem geral é ilustrada na Figura 2.4. Observe que existe uma aresta direcionada de u para v no grafo de horing se u encurta v. Por outro lado, existe uma aresta no grafo de previsibilidade se u encurta v e v prevê u. Portanto, o grafo de previsibilidade é obtido a partir do grafo de horing, eliminando algumas arestas. Este grafo é configurado em uma fase offline e é repetidamente consultado para recomendações. Além disso, várias estruturas de dados de índice são configuradas a partir da matriz de classificações durante a fase de configuração offline. Essas estruturas de dados são usadas junto com o grafo de previsibilidade para resolver as consultas de forma eficiente. Mais detalhes sobre a abordagem de horing podem ser encontrados em [33].

Essa abordagem pode funcionar para matrizes muito esparsas, pois utiliza a transitividade para prever classificações. Um desafio importante nos métodos de vizinhança é a falta de cobertura da previsão de classificação. Por exemplo, se nenhum dos vizinhos imediatos de John tiver classificado o Terminator, é impossível fornecer uma previsão de classificação para John. No entanto, a transitividade estrutural nos permite verificar se os vizinhos indiretos de John têm classificado o Terminator. Portanto, a principal vantagem dessa abordagem é que ela tem uma cobertura melhor em comparação com métodos concorrentes.

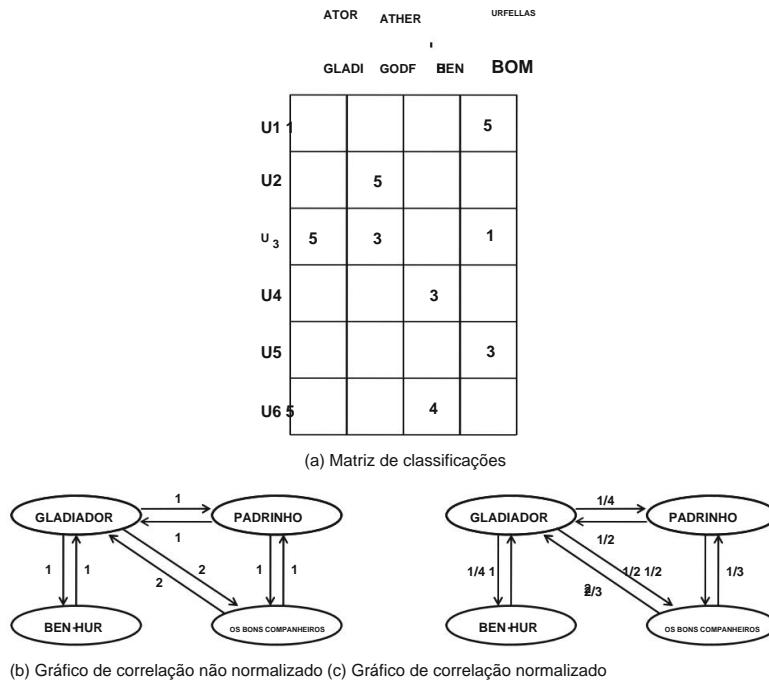


Figura 2.5: Uma matriz de classificação e seus gráficos de correlação

### 2.7.3 Gráficos Item-Item

Também é possível utilizar gráficos item-item para realizar as recomendações. Tal gráfico também é conhecido como gráfico de correlação [232]. Neste caso, um gráfico ponderado e direcionado é construída a rede  $G = (N, A)$ , na qual cada nó em  $N$  corresponde a um item, e cada aresta em  $A$  corresponde a uma relação entre itens. O peso  $w_{ij}$  está associado com cada aresta  $(i, j)$ . Se os itens  $i$  e  $j$  foram avaliados por pelo menos um usuário comum, então ambas as arestas direcionadas  $(i, j)$  e  $(j, i)$  existem na rede. Caso contrário, não existem arestas entre os nós  $i$  e  $j$ . A rede direcionada é, no entanto, assimétrica porque o peso da aresta  $(i, j)$  não é necessariamente o mesmo que o da aresta  $(j, i)$ . Seja  $U_i$  o conjunto de usuários que especificaram classificações para o item  $i$  e  $U_j$  seja o conjunto de usuários que especificaram classificações para item  $j$ . Então, o peso da aresta  $(i, j)$  é calculado usando o seguinte algoritmo simples.

Primeiro, inicializamos o peso  $w_{ij}$  de cada aresta  $(i, j)$  para  $|U_i \cap U_j|$ . Neste ponto, as arestas os pesos são simétricos porque  $w_{ij} = w_{ji}$ . Então, os pesos das arestas são normalizados, de modo que a soma dos pesos das arestas de saída de um nó seja igual a 1. Essa normalização é obtida dividindo  $w_{ij}$  pela soma dos pesos de saída do nó  $i$ .

A etapa de normalização resulta em pesos assimétricos, porque cada um dos pesos  $w_{ij}$  e  $w_{ji}$  são divididos por quantidades diferentes. Isso resulta em um gráfico no qual os pesos nas arestas correspondem a probabilidades de passeio aleatório. Um exemplo do gráfico de correlação para uma classificação A matriz é ilustrada na Figura 2.5. É claro que os pesos na correlação normalizada os gráficos não são simétricos devido à escala dos pesos para probabilidades de transição. Além disso, é importante ressaltar que os valores de classificação não são utilizados na construção do gráfico de correlação. Apenas o número de avaliações observadas em comum entre dois itens

é usado. Às vezes, isso não é desejável. É claro que é possível definir o gráfico de correlação de outras maneiras, como com o uso da função cosseno entre os vetores de classificação dos dois itens.

Conforme discutido no Capítulo 10, métodos de caminhada aleatória podem ser usados para determinar a vizinhança de um determinado item. A vizinhança resultante pode ser usada para métodos de filtragem colaborativa baseados em itens. Além disso, métodos personalizados de PageRank podem ser usados para determinar diretamente as classificações no gráfico item-item. Esse método é conhecido como ItemRank e é discutido na seção 10.2.3.3 do Capítulo 10.

## 2.8 Resumo

---

Como a filtragem colaborativa pode ser vista como uma generalização de problemas de classificação e regressão, as metodologias para as últimas classes de problemas também podem ser aplicadas às primeiras. Métodos baseados em vizinhança derivam sua inspiração de métodos de classificação e regressão do vizinho mais próximo. Em métodos baseados no usuário, o primeiro passo é determinar a vizinhança do usuário-alvo. Para calcular a vizinhança, uma variedade de funções de similaridade, como o coeficiente de correlação de Pearson ou o cosseno, são usadas. A vizinhança é usada para extrapolar as classificações desconhecidas de um registro. Em métodos baseados em itens, os itens mais semelhantes são computados em relação a um item-alvo. Em seguida, as próprias classificações do usuário sobre esses itens semelhantes são usadas para fazer uma previsão de classificação.

Métodos baseados em itens provavelmente terão recomendações mais relevantes, mas são menos propensos a produzir recomendações diversas. Para acelerar os métodos baseados em vizinhança, o agrupamento é frequentemente utilizado.

Métodos baseados em vizinhança podem ser vistos como modelos lineares, nos quais os pesos são escolhidos de forma heurística com o uso de valores de similaridade. Esses pesos também podem ser aprendidos com o uso de modelos de regressão linear. Esses métodos têm a vantagem de poderem ser combinados com outros modelos de otimização, como a fatoração de matrizes, para melhor predição. Esses métodos são discutidos no próximo capítulo.

Métodos baseados em vizinhança enfrentam inúmeros desafios devido à escassez de dados. Os usuários frequentemente especificam apenas um pequeno número de classificações. Como resultado, um par de usuários pode frequentemente ter especificado apenas um pequeno número de classificações. Tais situações podem ser abordadas efetivamente com o uso de modelos de redução de dimensionalidade e baseados em grafos. Embora os métodos de redução de dimensionalidade sejam frequentemente usados como métodos independentes para filtragem colaborativa, eles também podem ser combinados com métodos baseados em vizinhança para melhorar a eficácia e a eficiência da filtragem colaborativa. Vários tipos de grafos podem ser extraídos de padrões de classificação, como grafos usuário-item, grafos usuário-usuário ou grafos item-item. Normalmente, métodos de passeio aleatório ou caminho mais curto são usados nesses casos.

## 2.9 Notas Bibliográficas

---

Métodos baseados em vizinhança estão entre as primeiras técnicas utilizadas na área de sistemas de recomendação. Os primeiros modelos de filtragem colaborativa baseados em usuários foram estudados em [33, 98, 501, 540]. Um levantamento abrangente de sistemas de recomendação baseados em vizinhança pode ser encontrado em [183]. A escassez é um grande problema em tais sistemas, e vários sistemas baseados em grafos foram projetados para mitigar o problema da escassez [33, 204, 647].

Métodos projetados especificamente para a cauda longa em algoritmos de recomendação são discutidos em [173, 463, 648].

Métodos baseados no usuário utilizam as avaliações de usuários semelhantes sobre o mesmo item para fazer previsões. Embora tais métodos tenham sido inicialmente bastante populares, eles não são facilmente escaláveis e, às vezes, imprecisos. Posteriormente, foram propostos métodos baseados em itens [181, 360, 524], que calculam as avaliações previstas em função das avaliações do mesmo usuário sobre itens semelhantes. Os métodos baseados em itens fornecem recomendações mais precisas, porém menos diversificadas.

A noção de centralização média para aprimorar algoritmos de recomendação foi proposta em [98, 501]. Uma comparação do uso do escore Z com a centralização média é estudada em [245, 258], e esses dois estudos fornecem resultados um tanto conflitantes. Vários métodos que não utilizam classificações absolutas, mas sim focam na ordenação das classificações em termos de pesos de preferência, são discutidos em [163, 281, 282]. Os métodos de ponderação de significância para minimizar a ênfase nos vizinhos que têm poucas classificações em comum com um determinado vizinho são discutidos em [71, 245, 247, 380]. Muitas variantes diferentes da função de similaridade são usadas para calcular o vizinho. Dois exemplos são a distância média quadrática [540] e a correlação de postos de Spearman [299]. A vantagem específica dessas medidas de distância não é totalmente clara, pois resultados conflitantes foram apresentados na literatura [247, 258]. No entanto, o consenso parece ser que a correlação de classificação de Pearson fornece os resultados mais precisos [247]. Técnicas para ajuste do impacto de itens muito populares são discutidas em [98, 280]. O uso de amplificação exponenciada para predição em métodos baseados em vizinhança é discutido em [98]. Uma discussão sobre o uso de técnicas de votação em métodos de vizinho mais próximo pode ser encontrada em [183]. Os métodos de votação podem ser vistos como uma generalização direta do classificador de vizinho mais próximo, em oposição a uma generalização da modelagem de regressão de vizinho mais próximo.

Métodos para filtragem colaborativa baseada em itens foram propostos em [181, 524, 526]. Um estudo detalhado de diferentes variações de algoritmos de filtragem colaborativa baseada em itens é fornecido em [526], juntamente com uma comparação com relação aos métodos baseados no usuário. O método baseado em itens em [360] é notável porque descreve um dos métodos de filtragem colaborativa da Amazon.com. Os métodos de filtragem colaborativa baseados no usuário e em itens também foram unificados com a noção de fusão de similaridade [622]. Uma estrutura de unificação mais genérica pode ser encontrada em [613]. Métodos de agrupamento são usados frequentemente para melhorar a eficiência da filtragem colaborativa baseada em vizinhança. Vários métodos de agrupamento são descritos em [146, 167, 528, 643, 644, 647]. A extensão dos métodos de vizinhança para conjuntos de dados de grande escala foi estudada em [51].

As técnicas de redução de dimensionalidade têm uma rica história de uso em estimativas de valores ausentes [24, 472] e sistemas de recomendação [71, 72, 228, 252, 309, 313, 500, 517, 525]. De fato, a maioria dessas técnicas utiliza diretamente esses modelos latentes para prever as classificações sem depender de modelos de vizinhança. No entanto, algumas dessas técnicas de redução de dimensionalidade [71, 72, 309, 525] são projetadas especificamente para melhorar a eficácia e a eficiência das técnicas baseadas em vizinhança. Uma contribuição fundamental de [72] é fornecer uma visão sobre a relação entre métodos de vizinhança e métodos baseados em regressão. Essa relação é importante porque mostra como se pode formular métodos baseados em vizinhança como métodos baseados em modelos com uma formulação de otimização precisa. Observe que muitos outros métodos baseados em modelos, como modelos de fatores latentes, também podem ser expressos como formulações de otimização. Essa observação abre caminho para a combinação de métodos de vizinhança com modelos de fatores latentes em uma estrutura unificada [309], pois agora é possível combinar as duas funções objetivo. Outros modelos baseados em regressão para sistemas de recomendação, como preditores de declive um e métodos de mínimos quadrados ordinários, são propostos em [342, 620]. Métodos para aprender preferências em pares sobre conjuntos de itens são discutidos em [469]. Os modelos de regressão item-item também foram estudados no contexto de Modelos Lineares Esparsos (SLIM) [455], onde um regularizador de rede elástica é usado no modelo linear sem restringir os coeficientes ao

vizinhança do item. Métodos de aprendizagem esparsa de ordem superior, que modelam os efeitos de usar combinações de itens, são discutidos em [159]. Métodos eficientes para treinar modelos lineares e ajustar parâmetros de regularização são discutidos em [347]. Linear restrito métodos de regressão são discutidos em [430].

Um exame geral de classificadores lineares, como regressão de mínimos quadrados e suporte máquinas vetoriais, é fornecido em [669]. No entanto, a abordagem é projetada para feedback implícito conjuntos de dados nos quais apenas preferências positivas são especificadas. Observou-se que a colaboração a filtragem, nesses casos, é semelhante à categorização de texto. No entanto, devido ao ruído em os dados e a natureza desequilibrada da distribuição de classes, um uso direto dos métodos SVM às vezes não é eficaz. Alterações na função de perda são sugeridas em [669] para fornecer resultados mais precisos.

Muitos métodos baseados em grafos foram propostos para aprimorar algoritmos de filtragem colaborativa. A maioria desses métodos é baseada em grafos de usuário-item, mas alguns também são com base em gráficos de usuário-usuário. Uma observação importante da perspectiva de gráficos baseados em métodos é que eles mostram uma relação interessante entre os problemas de classificação, recomendação e predição de links. O uso de caminhadas aleatórias para determinar a vizinhança em sistemas de recomendação é discutido em [204, 647]. Um método, que usa o número de caminhos descontados entre um par de nós em um gráfico de item de usuário para recomendações, foi proposto em [262]. Esta abordagem é equivalente ao uso da medida de Katz entre pares de usuários para determinar se eles residem nos bairros um do outro.

Esta abordagem está relacionada com a predição de links [354], porque a medida de Katz é frequentemente usada para determinar a afinidade de ligação entre um par de nós. Uma pesquisa sobre métodos de predição de ligação pode ser encontrada em [17]. Alguns métodos baseados em gráficos não utilizam vizinhanças diretamente. Por exemplo, o método ItemRank proposto em [232] mostra como usar a classificação diretamente para fazer previsões, e o método em [261] mostra como usar métodos de previsão de link diretamente para filtragem colaborativa. Esses métodos também são discutidos no Capítulo 10 deste livro. Técnicas para alavancar gráficos usuário-usuário são discutidas em [33]. Esses métodos têm a vantagem de codificarem diretamente as relações de similaridade usuário-usuário nas bordas do gráfico. Como resultado, a abordagem oferece melhor cobertura do que métodos concorrentes.

## 2.10 Exercícios

---

1. Considere a matriz de classificações da Tabela 2.1. Preveja a classificação absoluta do item 3 para o usuário 2 usando:
  - (a) Filtragem colaborativa baseada no usuário com correlação de Pearson e centralização média
  - (b) Filtragem colaborativa baseada em itens com similaridade de cosseno ajustada

Use uma vizinhança de tamanho 2 em cada caso.

2. Considere a seguinte tabela de classificações entre cinco usuários e seis itens:

Id do item \ usuário	1	2	3	4	5	6			
1			5	6	7	4	3	?	
2			4	?	3	?	5	4	
3			?	3	4	1	1	?	
4			7	4	3	6	?	4	
5			1	?	3	2	2	5	

(a) Preveja os valores das avaliações não especificadas do usuário 2 usando algoritmos de filtragem colaborativa baseados em usuários. Use a correlação de Pearson com centralização na média. (b) Preveja os valores das avaliações não especificadas do usuário 2 usando algoritmos de filtragem colaborativa baseados em itens. Use algoritmos de filtragem. Use a similaridade de cosseno ajustada.

Suponha que um grupo de pares de tamanho máximo 2 seja usado em cada caso, e correlações negativas sejam filtradas.

3. Discuta a semelhança entre um classificador de k-vizinhos mais próximos no aprendizado de máquina tradicional e o algoritmo de filtragem colaborativa baseado em usuário. Descreva um classificador análogo à filtragem colaborativa baseada em itens.
4. Considere um algoritmo que realiza o agrupamento de usuários com base em sua matriz de avaliações e relata as avaliações médias dentro de um cluster como as avaliações de itens previstas para cada usuário dentro de um cluster. Discuta as compensações entre eficácia e eficiência dessa abordagem em comparação com um modelo de vizinhança.
5. Proponha um algoritmo que utilize caminhadas aleatórias em um gráfico usuário-usuário para realizar filtragem colaborativa baseada em vizinhança. [Esta questão requer experiência em métodos de classificação.]
6. Discuta várias maneiras pelas quais os algoritmos de agrupamento de gráficos podem ser usados para executar filtragem colaborativa baseada em vizinhança.
7. Implementar algoritmos de filtragem colaborativa baseados em usuários e itens.
8. Suponha que você tenha perfis baseados em conteúdo associados a usuários que indicam seus interesses e perfis associados a itens correspondentes às suas descrições. Ao mesmo tempo, você tenha uma matriz de classificação entre usuários e itens. Discuta como você pode incorporar as informações baseadas em conteúdo à estrutura de algoritmos baseados em gráficos.
9. Suponha que você tenha uma matriz de classificações unária. Mostre como algoritmos de filtragem colaborativa podem ser resolvidos usando métodos baseados em conteúdo, tratando as classificações de um item como suas características. Consulte o Capítulo 1 para uma descrição dos métodos baseados em conteúdo. A que tipo de classificador baseado em conteúdo corresponde um algoritmo de filtragem colaborativa baseado em itens?

---

## Capítulo 3

# Filtragem colaborativa baseada em modelo

---

“Não extinga a sua inspiração e a sua imaginação; não se torne escravo do seu modelo.” –  
Vincent van Gogh

### 3.1 Introdução

---

Os métodos baseados em vizinhança do capítulo anterior podem ser vistos como generalizações dos classificadores de k-vizinhos mais próximos, comumente usados em aprendizado de máquina. Esses métodos são métodos baseados em instâncias, nos quais um modelo não é criado especificamente para predição, exceto por uma fase opcional de pré-processamento<sup>1</sup>, necessária para garantir uma implementação eficiente. Métodos baseados em vizinhanças são generalizações de métodos de aprendizado baseados em instâncias ou métodos de aprendizado lento, nos quais a abordagem de predição é específica para a instância que está sendo prevista. Por exemplo, em métodos de vizinhança baseados em usuários, os pares do usuário-alvo são determinados para realizar a predição.

Em métodos baseados em modelos, um modelo resumido dos dados é criado antecipadamente, como nos métodos de aprendizado de máquina supervisionados ou não supervisionados. Portanto, a fase de treinamento (ou construção do modelo) é claramente separada da fase de predição. Exemplos de tais métodos no aprendizado de máquina tradicional incluem árvores de decisão, métodos baseados em regras, classificadores bayesianos, modelos de regressão, máquinas de vetores de suporte e redes neurais [22]. Curiosamente, quase todos esses modelos podem ser generalizados para o cenário de filtragem colaborativa, assim como os classificadores de k-vizinhos mais próximos podem ser generalizados para modelos baseados em vizinhança para filtragem colaborativa. Isso ocorre porque os problemas tradicionais de classificação e regressão são casos especiais do problema de complementação de matrizes (ou filtragem colaborativa).

No problema de classificação de dados, temos uma matriz  $m \times n$ , na qual as primeiras ( $n - 1$ ) colunas são variáveis de características (ou variáveis independentes), e a última (ou seja, enésima) coluna é

---

<sup>1</sup>Do ponto de vista prático, o pré-processamento é essencial para a eficiência. No entanto, pode-se implementar o método de vizinhança sem uma fase de pré-processamento, embora com latências maiores no momento da consulta.

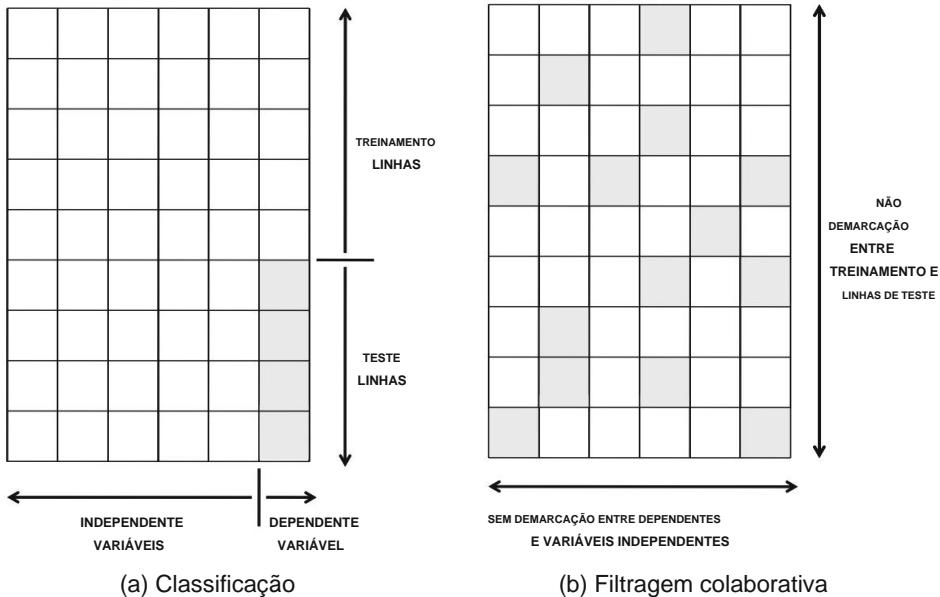


Figura 3.1: Revisitando a Figura 1.4 do Capítulo 1. Comparando o problema de classificação tradicional com a filtragem colaborativa. Entradas sombreadas estão ausentes e precisam ser previstas.

a variável de classe (ou variável dependente). Todas as entradas nas primeiras ( $n - 1$ ) colunas são totalmente especificadas, enquanto apenas um subconjunto das entradas na enésima coluna é especificado. Portanto, um subconjunto das linhas na matriz é totalmente especificado, e essas linhas são chamadas de dados de treinamento. As linhas restantes são chamadas de dados de teste. Os valores das entradas ausentes precisam ser aprendidos para os dados de teste. Este cenário é ilustrado na Figura 3.1(a), onde os valores sombreados representam as entradas ausentes na matriz.

Ao contrário da classificação de dados, qualquer entrada na matriz de classificações pode estar ausente, conforme ilustrado pelas entradas sombreadas na Figura 3.1(b). Assim, pode-se observar claramente que o problema de complementação de matrizes é uma generalização do problema de classificação (ou modelagem de regressão). Portanto, as diferenças cruciais entre esses dois problemas podem ser resumidas da seguinte forma:

1. No problema de classificação de dados, há uma separação clara entre variáveis de característica (independentes) e variáveis de classe (dependentes). No problema de complementação de matriz, essa separação clara não existe. Cada coluna é uma variável dependente e independente, dependendo de quais entradas estão sendo consideradas para a modelagem preditiva em um determinado momento.
  2. No problema de classificação de dados, há uma separação clara entre os dados de treinamento e de teste. No problema de complementação de matrizes, essa demarcação clara não existe entre as linhas da matriz. Na melhor das hipóteses, pode-se considerar as entradas especificadas (observadas) como dados de treinamento e as entradas não especificadas (ausentes) como dados de teste.
  3. Na classificação de dados, as colunas representam recursos e as linhas representam instâncias de dados.

No entanto, na filtragem colaborativa, é possível aplicar a mesma abordagem à matriz de classificação ou à sua transposta, devido à forma como as entradas ausentes são distribuídas. Por exemplo, modelos de vizinhança baseados em usuários podem ser vistos como modelos diretos.

generalizações de classificadores de vizinhos mais próximos. Quando tais métodos são aplicados ao transposição da matriz de classificação, eles são chamados de modelos de vizinhança baseados em itens. Em geral, muitas classes de algoritmos de filtragem colaborativa têm tanto o usuário e versões por item.

Essas diferenças entre classificação de dados e filtragem colaborativa são ilustradas em Figura 3.1. A maior generalidade do problema de filtragem colaborativa leva a um maior número de possibilidades algorítmicas na filtragem colaborativa, em comparação com a classificação de dados.

A semelhança entre o problema de filtragem colaborativa e a classificação de dados problema é útil ter em mente ao projetar algoritmos de aprendizagem para o primeiro. Isto é porque a classificação de dados é um campo relativamente bem estudado e os vários tipos de Soluções para classificação também fornecem dicas importantes para o design de algoritmos de filtragem colaborativa. De fato, a maioria dos algoritmos de aprendizado de máquina e classificação têm impacto direto. análogos na literatura de filtragem colaborativa. Compreendendo os sistemas de recomendação em um Uma abordagem semelhante aos modelos de classificação permite a aplicação de um número significativo de meta-algoritmos da literatura de classificação. Por exemplo, meta-algoritmos clássicos de A literatura de classificação, como bagging, boosting ou combinação de modelos, pode ser estendida à filtragem colaborativa. Curiosamente, grande parte da teoria desenvolvida para a filtragem de conjuntos Os métodos de classificação continuam a ser aplicados aos sistemas de recomendação. De fato, os métodos baseados em conjuntos [311, 704] estavam entre os métodos de melhor desempenho no desafio Netflix. Os métodos de conjunto são discutidos em detalhes no Capítulo 6.

No entanto, nem sempre é fácil generalizar modelos de classificação de dados diretamente para problema de completamento de matriz, especialmente quando a grande maioria das entradas está faltando. Além disso, a eficácia relativa dos vários modelos é diferente em diferentes contextos. Por exemplo, vários modelos recentes de filtragem colaborativa, como o fator latente modelos, são particularmente adequados para filtragem colaborativa. Tais modelos, no entanto, não são considerou modelos competitivos no contexto de classificação de dados.

Os sistemas de recomendação baseados em modelos geralmente apresentam uma série de vantagens em relação métodos baseados em vizinhança:

1. Eficiência de espaço: Normalmente, o tamanho do modelo aprendido é muito menor do que o matriz de classificação original. Assim, os requisitos de espaço são frequentemente bastante baixos. Por outro lado, Por outro lado, um método de vizinhança baseado no usuário pode ter complexidade de espaço  $O(m^2)$ , onde  $m$  é o número de usuários. Um método baseado em itens terá complexidade de espaço  $O(n^2)$ .
2. Velocidade de treinamento e velocidade de previsão: Um problema com métodos baseados em vizinhança é que a fase de pré-processamento é quadrática em termos de número de usuários ou número de itens. Os sistemas baseados em modelos são geralmente muito mais rápidos no pré-processamento fase de construção do modelo treinado. Na maioria dos casos, o modelo compacto e resumido O modelo pode ser usado para fazer previsões de forma eficiente.
3. Evitando overfitting: O overfitting é um problema sério em muitos algoritmos de aprendizado de máquina, nos quais a previsão é excessivamente influenciada por artefatos aleatórios nos dados. Esse problema também é encontrado em modelos de classificação e regressão. A abordagem de sumarização de métodos baseados em modelos pode frequentemente ajudar a evitar o sobreajuste. Além disso, métodos de regularização podem ser usados para tornar esses modelos robustos.

Embora os métodos baseados na vizinhança estivessem entre os primeiros métodos de filtragem colaborativa métodos e também estavam entre os mais populares devido à sua simplicidade, eles não são necessariamente os modelos mais precisos disponíveis hoje. Na verdade, alguns dos modelos mais precisos os métodos são baseados em técnicas baseadas em modelos em geral e em modelos de fatores latentes em especial.

Este capítulo está organizado da seguinte forma. A Seção 3.2 discute o uso de árvores de decisão e regressão para sistemas de recomendação. Métodos de filtragem colaborativa baseados em regras são discutidos na Seção 3.3. O uso do modelo Bayesiano Naïf para sistemas de recomendação é discutido na Seção 3.4. Uma discussão geral sobre como outros métodos de classificação são estendidos à filtragem colaborativa é fornecida na Seção 3.5. Modelos de fatores latentes são discutidos na Seção 3.6. A integração de modelos de fatores latentes com modelos de vizinhança é discutida na seção 3.7. Um resumo é fornecido na seção 3.8.

## 3.2 Árvores de Decisão e Regressão

---

Árvores de decisão e de regressão são frequentemente utilizadas na classificação de dados. Árvores de decisão são projetadas para os casos em que a variável dependente é categórica, enquanto árvores de regressão são projetadas para os casos em que a variável dependente é numérica. Antes de discutir a generalização das árvores de decisão para a filtragem colaborativa, discutiremos primeiro a aplicação das árvores de decisão à classificação.

Considere o caso em que temos uma matriz  $R \times n$ . Sem perda de generalidade, suponha que as primeiras colunas ( $n - 1$ ) sejam as variáveis independentes e a última coluna seja a variável dependente. Para facilitar a discussão, suponha que todas as variáveis sejam binárias. Portanto, discutiremos a criação de uma árvore de decisão em vez de uma árvore de regressão. Posteriormente, discutiremos como generalizar essa abordagem para outros tipos de variáveis.

A árvore de decisão é um particionamento hierárquico do espaço de dados com o uso de um conjunto de decisões hierárquicas, conhecidas como critérios de divisão nas variáveis independentes. Em uma árvore de decisão univariada, uma única característica é usada por vez para realizar uma divisão. Por exemplo, em uma matriz binária  $R$ , na qual os valores das características são 0 ou 1, todos os registros de dados nos quais uma variável de característica cuidadosamente escolhida assume o valor 0 ficarão em um ramo, enquanto todos os registros de dados nos quais a variável de característica assume o valor 1 ficarão no outro ramo. Quando a variável de característica é escolhida de tal forma que seja correlacionada com a variável de classe, os registros de dados dentro de cada ramo tenderão a ser mais puros.

Em outras palavras, a maioria dos registros pertencentes às diferentes classes serão separados.

Em outras palavras, um dos dois ramos conterá predominantemente uma classe, enquanto o outro ramo conterá predominantemente a outra classe. Quando cada nó em uma árvore de decisão tem dois filhos, a árvore de decisão resultante é considerada uma árvore de decisão binária.

A qualidade da divisão pode ser avaliada usando o índice de Gini médio ponderado dos nós filhos criados a partir de uma divisão. Se  $p_1 \dots p_r$  são as frações de registros de dados pertencentes a  $r$  classes diferentes em um nó  $S$ , então o índice de Gini  $G(S)$  do nó é definido da seguinte forma:

$$G(S)=1 - \sum_{i=1}^r p_i^2 \quad (3.1)$$

O índice de Gini situa-se entre 0 e 1, sendo valores menores indicativos de maior poder discriminativo. O índice de Gini geral de uma divisão é igual à média ponderada do índice de Gini dos nós filhos. Aqui, o peso de um nó é definido pelo número de pontos de dados nele contidos. Portanto, se  $S_1$  e  $S_2$  são os dois filhos do nó  $S$  em uma árvore de decisão binária, com  $n_1$  e  $n_2$  registros de dados, respectivamente, então o índice de Gini da divisão  $S \setminus [S_1, S_2]$  pode ser avaliado da seguinte forma:

$$\text{Gini}(S \setminus [S_1, S_2]) = \frac{n_1 \cdot G(S_1) + n_2 \cdot G(S_2)}{n_1 + n_2} \quad (3.2)$$

## 3.2. ÁRVORES DE DECISÃO E REGRESSÃO

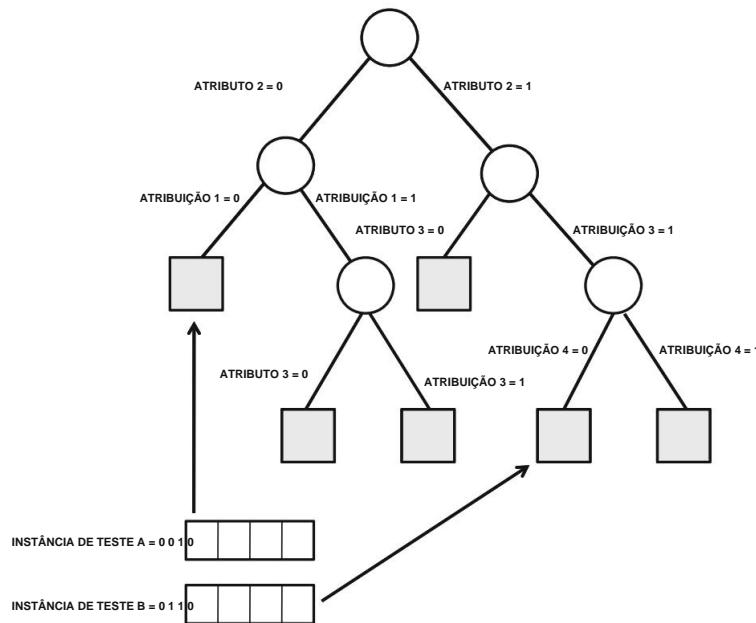


Figura 3.2: Exemplo de uma árvore de decisão com quatro atributos binários

O índice de Gini é usado para selecionar o atributo apropriado a ser usado para realizar a divisão em um determinado nível da árvore. Pode-se testar cada atributo para avaliar o índice de Gini de sua divisão de acordo com a Equação 3.2. O atributo com o menor índice de Gini é selecionado para realizar a divisão. A abordagem é executada hierarquicamente, de cima para baixo, até que cada nó contenha apenas registros de dados pertencentes a uma classe específica. Também é possível interromper o crescimento da árvore precocemente, quando uma fração mínima dos registros no nó pertence a uma classe específica. Esse nó é chamado de nó folha e é rotulado com a classe dominante dos registros naquele nó. Para classificar uma instância de teste com um valor desconhecido da variável dependente, suas variáveis independentes são usadas para mapear um caminho na árvore de decisão da raiz até a folha. Como a árvore de decisão é um particionamento hierárquico do espaço de dados, a instância de teste seguirá exatamente um caminho da raiz até a folha. O rótulo da folha é relatado como o relevante para a instância de teste. Um exemplo de árvore de decisão, construída com base em quatro atributos binários, é ilustrado na Figura 3.2. Os nós-folha da árvore estão sombreados na figura. Observe que nem todos os atributos são necessariamente usados para divisões pela árvore de decisão. Por exemplo, o caminho mais à esquerda usa os atributos 1 e 2, mas não usa os atributos 3 e 4. Além disso, diferentes caminhos na árvore de decisão podem usar diferentes sequências de atributos. Essa situação é particularmente comum com dados de alta dimensionalidade.

Exemplos de mapeamentos das instâncias de teste  $A = 0010$  e  $B = 0110$  para os respectivos nós folha são ilustrados na Figura 3.2. Cada uma dessas instâncias de teste é mapeada para um folha exclusivo devido à natureza hierárquica do particionamento de dados.

A abordagem pode ser estendida a variáveis numéricas dependentes e independentes com pequenas modificações. Para lidar com variáveis numéricas independentes (características), os valores dos atributos podem ser divididos em intervalos para realizar as divisões. Observe que essa abordagem pode resultar em uma divisão multidirecional, em que cada ramo da divisão corresponde a um intervalo diferente. A divisão é então realizada escolhendo o atributo com base no índice de Gini.

critério. Essa abordagem também se aplica a variáveis de características categóricas, em que cada valor do atributo categórico corresponde a um ramo da divisão.

Para lidar com variáveis dependentes numéricas, o critério de divisão é alterado do índice de Gini para uma medida mais adequada a atributos numéricos. Especificamente, a variância da variável dependente numérica é usada em vez do índice de Gini. Variâncias menores são mais desejáveis, pois significam que o nó contém instâncias de treinamento que são mapeadas discriminativamente na localidade da variável dependente. O valor médio no nó folha, ou um modelo de regressão linear, é usado no nó folha para realizar a predição [22].

Em muitos casos, a árvore é podada para reduzir o sobreajuste. Nesse caso, uma parte dos dados de treinamento não é utilizada durante a fase de construção da árvore. Em seguida, o efeito da poda do nó é testado na parte dos dados de treinamento que é mantida. Se a remoção do nó melhorar a precisão da previsão da árvore de decisão com base nos dados mantidos, o nó é podado. Além disso, outras variações dos critérios de divisão, como taxas de erro e entropia, são comumente utilizadas. Discussões detalhadas sobre as diversas opções de projeto na construção de árvores de decisão podem ser encontradas em [18, 22].

### 3.2.1 Estendendo Árvores de Decisão para Filtragem Colaborativa

O principal desafio na extensão de árvores de decisão para a filtragem colaborativa é que as entradas previstas e as entradas observadas não são claramente separadas em colunas como variáveis de característica e classe. Além disso, a matriz de classificações é muito escassamente preenchida, com a maioria das entradas ausentes. Isso cria desafios no particionamento hierárquico dos dados de treinamento durante a fase de construção da árvore. Além disso, como as variáveis dependentes e independentes (itens) não são claramente demarcadas na filtragem colaborativa, qual item deve ser previsto pela árvore de decisão?

Esta última questão é relativamente fácil de abordar construindo árvores de decisão separadas para prever a classificação de cada item. Considere uma matriz de classificações  $R \times n$  com  $m$  usuários e  $n$  itens. Uma árvore de decisão separada precisa ser construída, fixando cada atributo (item) como dependente e os demais como independentes. Portanto, o número de árvores de decisão construídas é exatamente igual ao número  $n$  de atributos (itens). Ao prever a classificação de um item específico para um usuário, a árvore de decisão correspondente ao item relevante é usada para a previsão.

Por outro lado, a questão da ausência de recursos independentes é mais difícil de abordar. Considere o caso em que um item específico (por exemplo, um filme específico) é usado como um atributo de divisão. Todos os usuários cuja classificação é menor que um limite são atribuídos a um ramo da árvore, enquanto os usuários cujas classificações são maiores que o limite são atribuídos ao outro ramo. Como as matrizes de classificação são esparsas, a maioria dos usuários não terá classificações especificadas para este item. A qual ramo esses usuários devem ser atribuídos? A lógica determina que esses usuários devem ser atribuídos a ambos os ramos. No entanto, nesse caso, a árvore de decisão não permanece mais um particionamento estrito dos dados de treinamento. Além disso, de acordo com essa abordagem, as instâncias de teste serão mapeadas para vários caminhos na árvore de decisão, e as previsões possivelmente conflitantes dos vários caminhos precisarão ser combinadas em uma única previsão.

Uma segunda abordagem (e mais razoável) é criar uma representação de menor dimensão dos dados usando os métodos de redução de dimensionalidade discutidos na seção 2.5.1.1 do Capítulo 2. Considere o cenário em que a classificação do j-ésimo item precisa ser prevista.

Logo no início, a matriz de classificações  $m \times (n - 1)$ , excluindo a j-ésima coluna, é convertida em uma representação  $m \times d$  de dimensão inferior, na qual  $d < n - 1$  e todos os atributos são totalmente especificados. A covariância entre cada par de itens na matriz de classificações  $m \times (n - 1)$  é estimada usando os métodos discutidos na seção 2.5.1.1 do Capítulo 2. A representação superior d

autovetores e  $e_1 \dots e_d$  da matriz de covariância estimada  $(n \times 1) \times (n \times 1)$  são determinados.

Observe que cada autovetor é um vetor contendo  $(n \times 1)$  elementos. A Equação 2.17 é usada para projetar as classificações de cada usuário nos autovetores, exceto que o  $j$ -ésimo item não está incluído no lado direito da Equação 2.17. Isso resulta em um vetor  $d$ -dimensional de classificações para cada usuário, que é completamente especificado. Essa representação reduzida é usada para construir a árvore de decisão para o  $j$ -ésimo item, tratando o problema como um problema padrão de classificação ou modelagem de regressão. Essa abordagem é repetida variando o valor de  $j$  de 1 a  $n$ , a fim de construir um total de  $n$  árvores de decisão. Portanto, a  $j$ -ésima árvore de decisão é útil apenas para prever as classificações do  $j$ -ésimo item. Tanto os autovetores quanto as árvores para cada um dos  $n$  casos são armazenados como parte do modelo.

Para prever a classificação do item  $j$  para um usuário  $i$ , a  $i$ -ésima linha da matriz  $m \times d$  é usada como instância de teste, e a  $j$ -ésima árvore de decisão/regressão é usada como modelo para prever o valor da classificação correspondente. O primeiro passo é usar os  $n - 1$  itens restantes (exceto o  $j$ -ésimo item) para criar a representação reduzida em  $d$ -dimensional da instância de teste, de acordo com a Equação 2.17. Observe que o  $j$ -ésimo conjunto de autovetores é usado para o processo de projeção e redução. Essa representação é então usada com a árvore de decisão ou regressão correspondente para o  $j$ -ésimo item para realizar a previsão. Vale ressaltar que essa abordagem mais ampla de combinar a redução de dimensionalidade com um modelo de classificação não se restringe a árvores de decisão. É relativamente fácil usar essa abordagem em conjunto com praticamente qualquer modelo de classificação. Além disso, métodos de redução de dimensionalidade também são usados isoladamente para prever classificações em sistemas de recomendação. Ambas as questões são discutidas posteriormente neste capítulo.

### 3.3 Filtragem colaborativa baseada em regras

---

A relação entre regras de associação [23] e filtragem colaborativa é natural, pois o problema das regras de associação foi proposto pela primeira vez no contexto da descoberta de relações entre dados de supermercados. As regras de associação são naturalmente definidas em dados binários, embora a abordagem possa ser estendida a dados categóricos e numéricos, convertendo esses tipos de dados em dados binários. Para os fins desta discussão, assumiremos o caso simplificado de dados unários, comuns em transações de supermercados e em conjuntos de dados de feedback implícito.

Considere um banco de dados de transações  $T = \{T_1 \dots T_m\}$ , contendo  $m$  transações, definidas em  $n$  itens  $I$ . Portanto,  $I$  é o conjunto universal de itens, e cada transação  $T_i$  é um subconjunto dos itens em  $I$ . A chave na mineração de regras de associação é determinar conjuntos de itens que estejam intimamente correlacionados no banco de dados de transações. Isso é alcançado com os conceitos de suporte e confiança. Essas medidas quantificam as relações entre conjuntos de itens.

**Definição 3.3.1 (Suporte)** O suporte de um conjunto de itens  $X \subseteq I$  é a fração de transações em  $T$ , das quais  $X$  é um subconjunto.

Se o suporte de um conjunto de itens for pelo menos igual a um limite predefinido  $s$ , então o conjunto de itens é considerado frequente. Esse limite é chamado de suporte mínimo. Esses conjuntos de itens são chamados de conjuntos de itens frequentes ou padrões frequentes. Conjuntos de itens frequentes podem fornecer insights importantes sobre correlações no comportamento de compra do cliente.

Por exemplo, considere o conjunto de dados ilustrado na Tabela 3.1. Nesta tabela, as linhas correspondem aos clientes e as colunas aos itens. Os 1s correspondem aos casos em que um determinado cliente comprou um item. Embora este conjunto de dados seja unário e os 0s correspondam a valores ausentes, uma prática comum em tais conjuntos de dados de feedback implícito é

Tabela 3.1: Exemplo de dados da cesta de compras

Item	Pão	Manteiga	Leite	Peixe	Carne	Presunto		
Cliente								
Jack	1	1	1	0	0	0		
Mary	0	1	1	0	1	0		
Jane	1		0	0	0	0		
Sayani	1	1	1	1	1	1		
John	0	0	0	1	0	1		
Tom	0	0	0	1	1	1		
Peter	0	1	0	1	1	0		

Aproxime os valores ausentes com 0s. É evidente que as colunas da tabela podem ser particionadas em dois conjuntos de itens intimamente relacionados. Um desses conjuntos é {Pão, Manteiga, Leite}, e o outro conjunto é {Peixe, Carne, Presunto}. Estes são os únicos conjuntos de itens com pelo menos 3 itens, que também têm um suporte de pelo menos 0,2. Portanto, ambos os conjuntos de itens são frequentes conjuntos de itens ou padrões frequentes. Encontrar esses padrões com alto suporte é útil para comerciante, porque ela pode usá-los para fazer recomendações e outras estratégias de marketing direcionadas decisões. Por exemplo, é razoável concluir que Maria provavelmente acabará comprando Pão, porque ela já comprou {Manteiga, Leite}. Da mesma forma, John provavelmente comprará Carne porque ele também comprou {Peixe, Presunto}. Tais inferências são muito úteis do ponto de vista visão de um sistema de recomendação.

Um nível adicional de percepção pode ser obtido em termos das direções dessas correlações usando a noção de regras de associação e confiança. Uma regra de associação é denotada na forma  $X \rightarrow Y$  onde o “ $\rightarrow$ ” pretende dar uma direção à natureza da correlação entre o conjunto de itens X e Y. Por exemplo, uma regra como {Manteiga, Leite}  $\rightarrow$  {Pão} seria muito útil recomendar Pão a Maria, porque já se sabe que ela comprou Leite e Manteiga. A força de tal regra é medida pela sua confiança.

**Definição 3.3.2 (Confiança)** A confiança da regra  $X \rightarrow Y$  é a condicional probabilidade de que uma transação em T contenha Y, dado que também contém X. Portanto, a confiança é obtida dividindo o suporte de  $X \rightarrow Y$  pelo suporte de X.

Note que o suporte de  $X \rightarrow Y$  será sempre menor que o suporte de X. Isso ocorre porque se uma transação contém X  $\rightarrow$  Y, então ela sempre conterá X. No entanto, o inverso pode não ser verdadeira. Portanto, a confiança de uma regra deve estar sempre no intervalo (0, 1). Maior Os valores da confiança são sempre indicativos de maior força da regra. Por exemplo, se uma regra  $X \rightarrow Y$  for verdadeira, então um comerciante, que sabe que um conjunto específico de clientes tem comprado o conjunto de itens X, também pode atingir esses clientes com o conjunto de itens Y. Um a regra de associação é definida com base em um suporte mínimo s e uma confiança mínima c:

**Definição 3.3.3 (Regras de Associação)** Uma regra  $X \rightarrow Y$  é considerada uma regra de associação com um suporte mínimo de s e uma confiança mínima de c, se as duas condições seguintes estão satisfeitos:

1. O suporte de  $X \rightarrow Y$  é pelo menos s.
2. A confiança de  $X \rightarrow Y$  é pelo menos c.

O processo de encontrar regras de associação é um algoritmo de duas fases. Na primeira fase, todos os conjuntos de itens que satisfazem um limite mínimo de suporte s são determinados. De cada um desses conjuntos de itens Z, todas as partições bidirecionais possíveis ( $X, Z \not\subseteq X$ ) são usadas para criar uma regra potencial  $X \not\rightarrow Z \not\rightarrow X$ . As regras que satisfazem a confiança mínima são retidas. A primeira fase de determinação dos conjuntos de itens frequentes é a computacionalmente intensiva, especialmente quando o banco de dados de transações subjacente é muito grande. Numerosos algoritmos computacionalmente eficientes foram dedicados ao problema da descoberta eficiente de conjuntos de itens frequentes. A discussão desses algoritmos está além do escopo deste livro, porque é um campo distinto da mineração de dados por si só. Leitores interessados podem consultar [23] para uma discussão detalhada da mineração de padrões frequentes. Neste livro, mostraremos como usar esses algoritmos como ferramentas para filtragem colaborativa.

### 3.3.1 Aproveitando as regras de associação para filtragem colaborativa

Regras de associação são particularmente úteis para realizar recomendações no contexto de matrizes de classificação unárias. Conforme discutido nos Capítulos 1 e 2, matrizes de classificação unárias são criadas pela atividade do cliente (por exemplo, comportamento de compra), em que há um mecanismo natural para o cliente especificar uma preferência por um item, mas nenhum mecanismo para especificar uma aversão. Nesses casos, os itens comprados por um cliente são definidos como 1, enquanto os itens ausentes são definidos como 0 como uma aproximação. Definir valores ausentes como 0 não é comum para a maioria dos tipos de matrizes de classificação, pois isso causaria viés nas previsões. No entanto, geralmente é considerada uma prática aceitável em matrizes unárias esparsas, pois o valor mais comum de um atributo geralmente é 0 nesses casos. Como resultado, o efeito do viés é relativamente pequeno e agora é possível tratar a matriz como um conjunto de dados binários.

O primeiro passo da filtragem colaborativa baseada em regras é descobrir todas as regras de associação em um nível pré-especificado de suporte mínimo e confiança mínima. O suporte mínimo e a confiança mínima podem ser vistos como parâmetros, que são ajustados<sup>2</sup> para maximizar a precisão preditiva. Apenas as regras cujo consequente contém exatamente um item são mantidas. Esse conjunto de regras é o modelo, que pode ser usado para realizar recomendações para usuários específicos. Considere um determinado cliente A ao qual se deseja recomendar itens relevantes. O primeiro passo é determinar todas as regras de associação que foram acionadas pelo cliente A.

Diz-se que uma regra de associação é disparada por um cliente A se o conjunto de itens no antecedente da regra for um subconjunto dos itens preferidos por esse cliente. Todas as regras disparadas são então classificadas em ordem decrescente de confiança. Os primeiros k itens descobertos nos consequentes dessas regras classificadas são recomendados como os principais k itens para o cliente A. A abordagem descrita aqui é uma simplificação do algoritmo descrito em [524]. Diversas outras variações dessa abordagem básica são utilizadas na literatura sobre sistemas de recomendação. Por exemplo, a escassez pode ser abordada usando métodos de redução de dimensionalidade [524].

As regras de associação mencionadas acima são baseadas em matrizes de classificação unárias, que permitem especificar gostos, mas não permitem especificar desgostos. No entanto, classificações numéricas podem ser facilmente manipuladas usando variações dessa metodologia básica. Quando o número de classificações possíveis é pequeno, cada valor da combinação classificação-item pode ser tratado como um pseudoitem. Um exemplo de pseudoitem é (Item = Pão, Classificação = Desgosto). Um novo conjunto de transações é criado em termos desses pseudoitens. As regras são então construídas em termos desses pseudoitens usando a abordagem discutida anteriormente.

---

<sup>2</sup>Métodos de ajuste de parâmetros, como hold-out e validação cruzada, são discutidos no Capítulo 7.

Portanto, tais regras poderiam aparecer da seguinte forma:

(Item = Pão, Avaliação = Curtir)  $\wedge$  (Item = Ovos, Avaliação = Curtir)

(Item = Pão, Classificação = Gostei) E (Item = Peixe, Classificação = Não gostei)  $\wedge$  (Item = Ovos, Classificação = Não gostei)

Para um determinado cliente, o conjunto de regras disparadas é determinado pela identificação das regras cujos antecedentes contêm um subconjunto dos pseudoitens para esse usuário. As regras são classificadas em ordem decrescente de confiança. Essas regras ordenadas podem ser usadas para prever classificações de itens selecionando os principais pseudoitens k nos consequentes dessas regras. Uma etapa adicional que pode ser necessária neste caso é resolver os conflitos entre as várias regras, pois diferentes pseudoitens nas regras disparadas por um cliente podem ser conflitantes. Por exemplo, os pseudoitens (Item = Pão, Classificação = Gostei) e (Item = Pão, Classificação = Não Gostei) são pseudoitens conflitantes. Tais conflitos podem ser resolvidos encontrando uma maneira de agregar as classificações nos consequentes para criar a lista final ordenada de recomendações. Também é possível agregar numericamente as classificações nos consequentes usando uma variedade de heurísticas. Por exemplo, pode-se primeiro determinar todas as regras disparadas nas quais os consequentes correspondem a um item de interesse. As classificações dos itens nas consequências dessas regras disparadas são votadas de forma ponderada para fazer uma previsão para aquela combinação usuário-item.

É possível ponderar as classificações nas regras disparadas pela confiança correspondente no processo de média. Por exemplo, se duas regras contêm a classificação "curtir" no consequente (para um item específico), com confiâncias de 0,9 e 0,8, respectivamente, então o número total de votos para "curtir" para esse item é  $0,9 + 0,8 = 1,7$ . Os votos podem ser usados para prever um valor médio da classificação para esse item. Esses valores previstos podem ser determinados para todos os itens nos consequentes das regras disparadas. Os valores resultantes podem ser usados para classificar os itens em ordem decrescente de prioridade. A abordagem de votação é mais adequada quando a granularidade da escala de classificação é muito limitada (por exemplo, gostar ou não gostar). No caso de classificações baseadas em intervalos com alta granularidade, é possível discretizar as classificações em um número menor de intervalos e, em seguida, usar a mesma abordagem discutida acima. Outros métodos heurísticos para agregar as previsões de métodos baseados em regras são discutidos em [18]. Em muitos casos, demonstrou-se que os resultados mais eficazes não são necessariamente obtidos utilizando o mesmo nível de suporte para cada item. Em vez disso, muitas vezes é desejável tornar o nível de suporte específico para o item cuja classificação está sendo prevista [358, 359, 365].

### 3.3.2 Modelos por item versus modelos por usuário

A dupla relação entre modelos baseados no usuário e no item é um tema recorrente na filtragem colaborativa. Os modelos de vizinhança do Capítulo 2 fornecem o exemplo mais conhecido dessa dualidade. Em geral, todo modelo baseado no usuário pode ser convertido em um modelo baseado no item aplicando-o à transposição da matriz de classificação e vice-versa. Às vezes, pequenos ajustes podem ser necessários para levar em conta as diferentes interpretações semânticas nos dois casos. Por exemplo, utiliza-se o cosseno ajustado para o cálculo de similaridade em modelos de vizinhança baseados em itens, em vez do coeficiente de correlação de Pearson.

A discussão acima mencionada concentra-se em modelos de itens para colaboração baseada em regras filtragem. Também é possível criar modelos baseados no usuário. Esses métodos alavancam associações de usuários em vez de associações de itens [358, 359]. Nesses casos, as regras associam os gostos dos usuários entre si, em vez de associar os gostos dos itens entre si. Portanto, trabalha-se com pseudousuários correspondentes a combinações de avaliações de usuários. Exemplos de tais

as regras são as seguintes:

(Usuário = Alice, Avaliação = Curtir)  $\wedge$  (Usuário = Bob, Avaliação = Não Curtir)

(Usuário = Alice, Avaliação = Curtir)  $E$  (Usuário = Peter, Avaliação = Não Curtir)  $\wedge$  (Usuário = John,

Avaliação = Curtir)

A primeira regra implica que Bob provavelmente não gostará de itens dos quais Alice gosta. A segunda regra implica que John provavelmente gostará de itens dos quais Alice gosta e Peter não gosta. Tais regras podem ser mineradas aplicando-se exatamente a mesma abordagem do caso anterior à transposição da matriz de transações construída a partir dos pseudousuários. Em outras palavras, cada lista de pseudousuários para um item é agora tratada como uma "transação". Regras de associação são mineradas desse banco de dados no nível mínimo de suporte e confiança exigidos. Para prever a classificação de uma combinação usuário-item, a "transação" baseada em pseudousuário para o item relevante é determinada. As regras são disparadas por essa transação quando o antecedente dessa regra contém um subconjunto dos pseudousuários na transação. Todas as regras disparadas são determinadas. Dentre essas regras disparadas, todas aquelas cujos consequentes correspondem ao usuário de interesse são determinadas. As classificações nos consequentes das regras disparadas podem ser calculadas em média ou votadas para fazer uma previsão. O processo de média pode ser ponderado com a confiança da regra correspondente para fornecer uma previsão mais robusta. Assim, a abordagem baseada no usuário é exatamente análoga à abordagem baseada em itens. Vale ressaltar que as duas maneiras de realizar a filtragem colaborativa com regras de associação compartilham uma relação complementar, que lembra os algoritmos de vizinhança baseados no usuário e nos itens.

A abordagem de regras de associação é útil não apenas para filtragem colaborativa, mas também para sistemas de recomendação baseados em conteúdo, nos quais perfis de clientes são correspondidos a itens específicos. Essas regras são chamadas de regras de associação de perfil e são usadas popularmente para recomendações baseadas em perfil. Foi demonstrado em [31, 32] como uma interface interativa eficiente pode ser construída para realizar recomendações baseadas em perfil para uma variedade de tipos diferentes de consultas.

Sistemas de recomendação baseados em regras de associação podem ser vistos como generalizações de sistemas baseados em regras que são comumente usados para o problema de classificação [18]. A principal diferença é que os consequentes das regras geradas no problema de classificação sempre contêm a variável de classe. No entanto, no caso de sistemas de recomendação, os consequentes das regras geradas podem conter<sup>3</sup> qualquer item. Além disso, as heurísticas para classificar as regras disparadas e combinar os resultados possivelmente conflitantes das regras também são semelhantes na filtragem e classificação colaborativas. Essa relação natural entre esses métodos é um resultado direto da relação entre os problemas de classificação e filtragem colaborativa. A principal distinção entre os dois casos é que não há uma demarcação clara entre as variáveis de característica e as variáveis de classe na filtragem colaborativa. É por isso que qualquer regra de associação pode ser gerada, em vez de simplesmente regras que contêm a variável de classe no consequente.

Diversos estudos comparativos demonstraram [358, 359] que sistemas de regras de associação podem fornecer resultados precisos em determinados tipos de cenários. Isso é particularmente verdadeiro para dados unários, comumente encontrados em sistemas de recomendação da Web. Sistemas baseados em regras de associação encontraram aplicações significativas em sistemas de personalização e recomendação baseados na Web [441, 552]. A abordagem é naturalmente adequada para sistemas de personalização da Web, pois é projetada especificamente para dados de transações esparsos, comumente encontrados no comportamento de cliques na Web. Tais métodos podem até ser estendidos para incluir informações temporais por meio do uso de modelos de mineração de padrões sequenciais [23].

---

<sup>3</sup>No caso de associações baseadas em usuários, os consequentes podem conter qualquer usuário.

### 3.4 Filtragem colaborativa Naive Bayes

---

A seguir, assumiremos que há um pequeno número de classificações distintas, cada uma das quais pode ser tratada como um valor categórico. Portanto, a ordenação entre as classificações será ignorada na discussão a seguir. Por exemplo, três classificações, como "Gostei", "Neutro" e "Não Gostei", serão tratadas como valores discretos não ordenados. No caso em que o número de classificações distintas é pequeno, essa aproximação pode ser razoavelmente utilizada sem perda significativa de precisão.

Suponha que existam  $l$  valores distintos das classificações, que são denotados por  $v_1 \dots v_l$ . Como no caso dos outros modelos discutidos neste capítulo, assumimos que temos uma matriz  $R$   $m \times n$  contendo as avaliações de  $m$  usuários para  $n$  itens. A entrada  $(u, j)$  da matriz é denotada por  $r_{uj}$ .

O modelo Bayes ingênuo é um modelo gerativo, comumente usado para classificação. É possível tratar os itens como características e os usuários como instâncias para inferir as entradas ausentes com um modelo de classificação. O principal desafio no uso dessa abordagem para filtragem colaborativa é que qualquer característica (item) pode ser a classe-alvo na filtragem colaborativa, e também é preciso trabalhar com variáveis de características incompletas. Essas diferenças podem ser tratadas com pequenas modificações na metodologia básica do modelo Bayesiano Naïf.

Considere o usuário  $u$ -ésimo, que especificou classificações para o conjunto de itens  $I_u$ . Em outras palavras, se a linha  $u$ -ésima tiver classificações especificadas para a primeira, terceira e quinta colunas, então temos  $I_u = \{1, 3, 5\}$ . Considere o caso em que o classificador de Bayes precisa prever a classificação não observada  $r_{uj}$  do usuário  $u$  para o item  $j$ . Observe que  $r_{uj}$  pode assumir qualquer uma das possibilidades discretas em  $\{v_1 \dots v_l\}$ . Portanto, gostaríamos de determinar a probabilidade de  $r_{uj}$  assumir qualquer um desses valores condicional às classificações observadas em  $I_u$ . Portanto, para cada valor de  $s \in \{1 \dots l\}$ , gostaríamos de determinar a probabilidade  $P(r_{uj} = vs | \text{Classificações observadas em } I_u)$ .

Esta expressão aparece na forma  $P(A|B)$ , onde  $A$  e  $B$  são eventos correspondentes ao valor de  $r_{uj}$  e aos valores das classificações observadas em  $I_u$ , respectivamente. A expressão pode ser simplificada usando a conhecida regra de Bayes na teoria da probabilidade:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} \quad (3.3)$$

Portanto, para cada valor de  $s \in \{1 \dots l\}$ , temos o seguinte:

$$P(r_{uj} = vs | \text{Classificações observadas em } I_u) = \frac{P(r_{uj} = vs) \cdot P(\text{Classificações observadas em } I_u | r_{uj} = vs)}{P(\text{Classificações observadas em } I_u)} \quad (3.4)$$

Precisamos determinar o valor de  $s$  na expressão acima mencionada para o qual o valor de  $P(r_{uj} = vs | \text{Observed ratings in } I_u)$  no lado esquerdo é o maior possível. Vale ressaltar que o denominador no lado direito da Equação 3.4 é independente do valor de  $s$ . Portanto, para determinar o valor de  $s$  no qual o lado direito assume o valor máximo, pode-se ignorar o denominador e expressar a equação acima mencionada em termos de uma constante de proporcionalidade:

$$P(r_{uj} = vs | \text{Classificações observadas em } I_u) \propto P(r_{uj} = vs) \cdot P(\text{Classificações observadas em } I_u | r_{uj} = vs) \quad (3.5)$$

Se desejado, a constante de proporcionalidade pode ser derivada garantindo que todos os valores de probabilidade resultantes  $P(r_{uj} = vs | \text{Classificações observadas em } I_u)$  para  $s \in \{1 \dots l\}$  somam 1. Uma observação importante é que todas as expressões do lado direito da Equação 3.5 podem ser estimadas facilmente de forma orientada por dados. O valor de  $P(r_{uj} = vs)$ , que também é referido

para como a probabilidade prévia de classificação  $r_{uj}$ , é estimada para a fração dos usuários que especificaram a classificação vs para o j-ésimo item. Observe que a fração é computada apenas a partir dos usuários que classificaram o item j, e os outros usuários são ignorados. A expressão  $P(\text{Classificações observadas em } lu|r_{uj} = vs)$  é estimada com o uso da suposição ingênua. A suposição ingênua é baseada na independência condicional entre as classificações. A suposição de independência condicional diz que as classificações do usuário u para vários itens em lu são independentes umas das outras, condicionadas ao fato de que o valor de  $r_{uj}$  foi observado como vs.

Esta condição pode ser expressa matematicamente da seguinte forma:

$$P(\text{Classificações observadas em } lu|r_{uj} = vs) = \frac{P(r_{uk}|r_{uj} = vs)}{k_{lu}} \quad (3.6)$$

O valor de  $P(r_{uk}|r_{uj} = vs)$  é estimado como a fração de usuários que especificaram a classificação de  $r_{uk}$  para o k-ésimo item, dado que eles especificaram a classificação do seu j-ésimo item para vs.

Ao inserir a estimativa da probabilidade anterior  $P(r_{uj} = vs)$  e a da Equação 3.6 na Equação 3.5, é possível obter uma estimativa da probabilidade posterior da classificação do item j para o usuário u da seguinte forma:

$$P(r_{uj} = vs|\text{Classificações observadas em } lu) \hat{=} P(r_{uj} = vs) \cdot \frac{P(r_{uk}|r_{uj} = vs)}{k_{lu}} \quad (3.7)$$

Esta estimativa da probabilidade posterior da classificação  $r_{uj}$  pode ser usada para estimar seu valor de uma das duas maneiras a seguir:

1. Calculando cada uma das expressões do lado direito da Equação 3.7 para cada s  $\in \{1 \dots l\}$  e determinando o valor de s em que ele é o maior, pode-se determinar o valor mais provável  $\hat{r}_{uj}$  da classificação ausente  $r_{uj}$ . Em outras palavras, temos:  $\hat{r}_{uj} = \operatorname{argmax}_{vs} P(r_{uj} = vs|\text{Classificações observadas em } lu)$

$$= \operatorname{argmax}_{vs} P(r_{uj} = vs) \cdot \frac{P(r_{uk}|r_{uj} = vs)}{k_{lu}}$$

Essa abordagem, no entanto, trata uma classificação puramente como um valor categórico e ignora toda a ordenação entre as diversas classificações. Quando o número de classificações possíveis é pequeno, essa é uma abordagem razoável.

2. Em vez de determinar a classificação que assume a probabilidade máxima, pode-se estimar o valor previsto como a média ponderada de todas as classificações, onde o peso de uma classificação é sua probabilidade. Em outras palavras, o peso da classificação vs é proporcional ao valor de  $P(r_{uj} = vs|\text{Classificações observadas em } lu)$ , conforme calculado na Equação 3.7. Observe que a constante de proporcionalidade na equação é irrelevante para o cálculo da média ponderada. Portanto, o valor estimado  $\hat{r}_{uj}$  da classificação ausente  $r_{uj}$  na matriz R é o seguinte:

$$\begin{aligned} \hat{r}_{uj} &= \frac{s=1 vs \cdot P(r_{uj} = vs|\text{Classificações observadas em } lu)}{\sum_{ls=1}^l P(r_{uj} = vs|\text{Classificações observadas em } lu)} \\ &= \frac{\sum_{lu} s=1 vs \cdot P(r_{uj} = vs) \cdot P(\text{Classificações observadas em } lu|r_{uj} = vs)}{\sum_{lu} P(r_{uj} = vs) \cdot P(\text{Classificações observadas em } lu|r_{uj} = vs)} \\ &= \frac{\sum_{lu} s=1 vs \cdot P(r_{uj} = vs) \cdot \frac{P(r_{uk}|r_{uj} = vs)}{k_{lu}}}{\sum_{lu} P(r_{uj} = vs) \cdot \frac{P(r_{uk}|r_{uj} = vs)}{k_{lu}}} \end{aligned}$$

Essa abordagem é preferível quando a granularidade da distribuição de classificações é maior.

Para um determinado usuário  $u$ , todas as suas avaliações não observadas são estimadas usando esta abordagem. Os itens com os  $k$  principais valores estimados das avaliações são relatados.

Vale ressaltar que essa abordagem calcula a probabilidade condicional de uma classificação com base nas classificações dos outros itens (ou dimensões). Portanto, essa abordagem é uma abordagem bayesiana baseada em itens. Trata-se de uma adaptação direta dos métodos tradicionais de classificação, exceto pelo fato de que a dimensão prevista (classe) é fixa na classificação tradicional, enquanto a dimensão prevista varia na filtragem colaborativa. Essa diferença ocorre porque a filtragem colaborativa é uma generalização da classificação (cf. Figura 3.1).

No caso específico da filtragem colaborativa, também é possível calcular a probabilidade de uma classificação com base nas classificações de outros usuários para o mesmo item (ver Exercício 4). Tal abordagem pode ser vista como uma abordagem Bayesiana baseada no usuário. É ainda possível combinar as previsões dos métodos Bayesianos baseados no usuário e nos itens. Em praticamente todas as formas de filtragem colaborativa, como os métodos baseados em vizinhança e em regras, é possível fornecer uma solução a partir da perspectiva do usuário, da perspectiva dos itens ou de uma combinação dos dois métodos.

### 3.4.1 Lidando com overfitting

Um problema surge quando a matriz de classificações subjacente é esparsa e o número de classificações observadas é pequeno. Nesses casos, as estimativas baseadas em dados podem não permanecer robustas. Por exemplo, a estimativa da probabilidade anterior  $P(r_{uj} = vs)$  provavelmente não será robusta se um pequeno número de usuários tiver especificado classificações para o  $j$ -ésimo item. Por exemplo, se nenhum usuário tiver especificado uma classificação para o  $j$ -ésimo item, a estimativa terá a forma 0/0, que é indeterminada. Além disso, a estimativa de cada valor  $P(r_{uk}|r_{uj} = vs)$  no lado direito da Equação 3.6 provavelmente será ainda menos robusta do que a estimativa da probabilidade anterior. Isso ocorre porque apenas uma pequena parte da matriz de classificações será condicional ao evento  $r_{uj} = vs$ . Nesse caso, a parte da matriz de classificações que precisa ser analisada é apenas a dos usuários que especificaram a classificação  $vs$  para o item  $j$ . Se o número desses usuários for pequeno, a estimativa será imprecisa e os termos multiplicativos na Equação 3.6 produzirão um grande erro.

Por exemplo, para qualquer valor de  $k \leq l_u$ , se nenhum usuário tiver especificado a classificação  $r_{uk}$  nos casos em que a classificação do  $j$ -ésimo item for definida como  $vs$ , toda a expressão da Equação 3.6 será definida como 0 devido à sua natureza multiplicativa. Este é, obviamente, um resultado errôneo e de sobreajuste, obtido devido à estimativa dos parâmetros do modelo a partir de uma pequena quantidade de dados.

Para lidar com esse problema, o método de suavização laplaciana é comumente usado. Por exemplo, seja  $q_1 \dots q_l$  o número de usuários que especificaram, respectivamente, as classificações  $v_1 \dots v_l$  para o  $j$ -ésimo item. Então, em vez de estimar  $P(r_{uj} = vs)$  de forma direta para  $qs$ ,

$t=1$  é suavizado com um parâmetro de suavização laplaciano  $\hat{y}$ :

$$P(r_{uj} = vs) = \frac{qs + \hat{y}}{t=1 + q_1 + \dots + \hat{y}} \quad (3.8)$$

Observe que, se nenhuma classificação for especificada para o  $j$ -ésimo item, essa abordagem definirá a probabilidade anterior de cada classificação possível como 1/1. O valor de  $\hat{y}$  controla o nível de suavização.

Valores maiores de  $\hat{y}$  levarão a uma suavização maior, mas os resultados se tornarão insensíveis aos dados subjacentes. Uma abordagem exatamente semelhante pode ser usada para suavizar a estimativa de  $P(r_{uk}|r_{uj} = vs)$ , adicionando  $\hat{y}$  e  $l - \hat{y}$  ao numerador e ao denominador, respectivamente.

Tabela 3.2: Ilustração do método de Bayes com uma matriz de classificações binárias

Id do item $\hat{y}$ 1			2	3	4	5	6		
		ID do usuário $\hat{y}$							
1		1	-1			-1	1	-1	
2		1	1	1?	-1	-1	-1		
		? 1	-1	1	-1	-1	1?		
3 4		-1	-1	1	-1	1		1 1	
5							1	1	

### 3.4.2 Exemplo do Método Bayes com Classificações Binárias

Nesta seção, ilustraremos o método de Bayes com uma matriz de classificações binárias em 5 usuários e 6 itens. As classificações são obtidas de  $\{v_1, v_2\} = \{\hat{y}_1, 1\}$ . Esta matriz é mostrada na Tabela 3.2. Para facilitar a discussão, não usaremos a suavização laplaciana, embora seja essencial fazê-lo na prática. Considere o caso em que desejamos prever as classificações de os dois itens não especificados do usuário 3. Portanto, precisamos calcular as probabilidades do classificações não especificadas  $r_{31}$  e  $r_{36}$  assumindo cada um dos valores de  $\{\hat{y}_1, 1\}$ , condicional em os valores observados das demais avaliações do usuário 3. Utilizando a Equação 3.7, obtemos a seguinte probabilidade posterior para a classificação do item 1 pelo usuário 3:

$$\begin{aligned} P(r_{31} = 1 | r_{32}, r_{33}, r_{34}, r_{35}) &\stackrel{\text{def}}{=} P(r_{31} = 1) \cdot P(r_{32} = 1 | r_{31} = 1) \cdot P(r_{33} = 1 | r_{31} = 1) \cdot \\ &\quad \cdot P(r_{34} = \hat{y}_1 | r_{31} = 1) \cdot P(r_{35} = \hat{y}_1 | r_{31} = 1) \end{aligned}$$

Os valores dos termos individuais no lado direito da equação acima mencionada são estimados usando os dados da Tabela 3.2 da seguinte forma:

$$\begin{aligned} P(r_{31} = 1) &= 2/4=0,5 \\ P(r_{32} = 1 | r_{31} = 1) &= 1/2=0,5 \\ P(r_{33} = 1 | r_{31} = 1) &= 1/1=1 \\ P(r_{34} = \hat{y}_1 | r_{31} = 1) &= 2/2=1 \\ P(r_{35} = \hat{y}_1 | r_{31} = 1) &= 1/2=0,5 \end{aligned}$$

Substituindo esses valores na equação acima, obtemos o seguinte:

$$P(r_{31} = 1 | r_{32}, r_{33}, r_{34}, r_{35}) \stackrel{\text{def}}{=} (0,5)(0,5)(1)(1)(0,5) = 0,125$$

Ao realizar os mesmos passos para a probabilidade de  $r_{31}$  assumir o valor de  $\hat{y}_1$ , temos obter:

$$P(r_{31} = \hat{y}_1 | r_{32}, r_{33}, r_{34}, r_{35}) \stackrel{\text{def}}{=} (0,5) \frac{0}{1} \frac{0}{2} \frac{0}{2} \frac{0}{2} = 0$$

Portanto, a classificação r31 tem maior probabilidade de assumir o valor 1, em comparação com -1, e seu valor previsto é definido como 1. Um argumento semelhante pode ser usado para demonstrar que o valor previsto da classificação r36 é -1. Portanto, em um cenário de recomendação top-1, o item 1 deve ser priorizado em relação ao item 6 em uma recomendação ao usuário 3.

### 3.5 Usando um modelo de classificação arbitrário como uma caixa preta

---

Muitos outros métodos de classificação (ou modelagem de regressão) podem ser estendidos ao caso de filtragem colaborativa. O principal desafio nesses métodos é a natureza incompleta dos dados subjacentes. No caso de alguns classificadores, é mais difícil ajustar o modelo para lidar com o caso de valores de atributos ausentes. Uma exceção é o caso de dados unários, nos quais os valores ausentes são frequentemente estimados como 0 e as entradas especificadas são definidas como 1. Portanto, a matriz subjacente se assemelha a dados binários esparsos de alta dimensionalidade. Nesses casos, os dados podem ser tratados como um conjunto de dados completo e quaisquer classificadores projetados para dados esparsos e de alta dimensionalidade podem ser usados. Felizmente, muitas formas de dados, incluindo dados de transações de clientes, dados de cliques na Web ou outros dados de atividade, podem ser formulados como uma matriz unária. Vale ressaltar que dados de texto também são esparsos e de alta dimensionalidade; como resultado, muitos dos algoritmos de classificação usados na mineração de texto podem ser diretamente adaptados a esses conjuntos de dados. De fato, foi demonstrado em [669] que é possível alavancar diretamente o sucesso das máquinas de vetores de suporte em dados de texto para filtragem colaborativa (unária), embora com uma forma quadrática da função de perda. A forma quadrática da função de perda torna o modelo mais semelhante à regressão linear regularizada. Também foi sugerido em [669] que o uso de métodos de aprendizado de classes raras pode ser eficaz na filtragem colaborativa devido à natureza desequilibrada da distribuição de classes. Por exemplo, pode-se usar diferentes funções de perda para as classes majoritárias e minoritárias enquanto se adapta a máquina de vetores de suporte ao cenário de filtragem colaborativa. Vários métodos ad hoc também foram propostos para estender vários métodos de classificação e regressão à filtragem colaborativa. Por exemplo, máquinas de vetores de suavização [638] foram usadas para estimar os valores ausentes na matriz usuário-item de forma iterativa.

Para casos em que a matriz de classificações não é unária, não é mais possível preencher as entradas ausentes da matriz com 0s sem causar viés significativo. Essa questão é discutida em detalhes na seção 2.5 do Capítulo 2. No entanto, como discutido na mesma seção, vários métodos de redução de dimensionalidade podem ser usados para criar uma representação de baixa dimensão dos dados, que seja totalmente especificada. Nesses casos, qualquer método de classificação conhecido pode ser usado efetivamente, tratando a representação de baixa dimensão como as variáveis de características dos dados de treinamento. Qualquer coluna que precise ser preenchida é tratada como a variável de classe. O principal problema com essa abordagem é a perda de interpretabilidade no processo de classificação.

Quando a representação reduzida representa uma combinação linear das colunas originais, é difícil fornecer qualquer tipo de explicação das previsões.

Para trabalhar no espaço de características original, é possível usar métodos de classificação como meta-algoritmos em conjunto com métodos iterativos. Em outras palavras, um algoritmo de classificação pronto para uso é usado como uma caixa-preta para prever as classificações de um dos itens com as classificações de outros itens. Como superar o problema de as colunas de treinamento terem sido especificadas de forma incompleta? O truque é preencher iterativamente os valores ausentes da coluna.

colunas de treinamento com refinamento sucessivo. Esse refinamento sucessivo é obtido com o uso da nossa caixa-preta, que é um algoritmo de classificação (ou modelagem de regressão) pronto para uso.

Considere um algoritmo arbitrário de modelagem de classificação/regressão A, projetado para funcionar com uma matriz completamente especificada. O primeiro passo é inicializar as entradas ausentes na matriz com médias de linha, médias de coluna ou com qualquer algoritmo simples de filtragem colaborativa. Por exemplo, pode-se usar um algoritmo simples baseado no usuário para o processo de inicialização. Como um aprimoramento opcional, pode-se centralizar cada linha da matriz de classificações como uma etapa de pré-processamento para remover o viés do usuário. Nesse caso, o viés de cada usuário precisa ser adicionado novamente aos valores previstos em uma fase de pós-processamento. A remoção do viés do usuário durante o pré-processamento geralmente torna<sup>4</sup> a abordagem mais robusta. Se o viés do usuário for removido, as entradas ausentes serão sempre preenchidas com médias de linha, que são 0.

Essas inicializações simples e métodos de remoção de viés ainda levarão a viés de previsão, quando se tenta usar os valores preenchidos artificialmente como dados de treinamento. Assim, o viés nas entradas previstas pode ser reduzido iterativamente usando a seguinte abordagem iterativa em duas etapas:

1. (Etapa iterativa 1): Use o algoritmo A para estimar as entradas ausentes de cada coluna, definindo-a como a variável de destino e as colunas restantes como as variáveis de recurso.  
Para as colunas restantes, use o conjunto atual de valores preenchidos para criar uma matriz completa de variáveis de recursos. As classificações observadas na coluna de destino são usadas para treinamento, e as classificações ausentes são previstas.
2. (Etapa iterativa 2): Atualizar todas as entradas ausentes com base na previsão do algoritmo A em cada coluna de destino.

Essas duas etapas são executadas iterativamente até a convergência. A abordagem pode ser sensível à qualidade da inicialização e do algoritmo A. No entanto, o mérito da abordagem reside no fato de ser um método simples que pode ser facilmente implementado com qualquer modelo de classificação ou regressão disponível no mercado. Classificações numéricas também podem ser tratadas com um modelo de regressão linear. O trabalho em [571] utiliza uma abordagem semelhante, na qual a matriz de classificações é imputada com entradas artificiais previstas por um conjunto de diferentes classificadores.

### 3.5.1 Exemplo: Usando uma rede neural como uma caixa preta

Nesta seção, forneceremos um exemplo simples da abordagem mencionada, utilizando redes neurais como caixas-pretas para implementá-la. Para os fins da discussão a seguir, assumiremos que o leitor já esteja familiarizado com os fundamentos das redes neurais [87]. No entanto, iremos apresentá-los brevemente para garantir a continuidade da discussão.

Redes neurais simulam o cérebro humano com o uso de neurônios, que são conectados entre si por meio de conexões sinápticas. Em sistemas biológicos, a aprendizagem é realizada pela alteração da intensidade das conexões sinápticas em resposta a estímulos externos. Em redes neurais artificiais, a unidade básica de computação também é chamada de neurônio, e as intensidades das conexões sinápticas correspondem a pesos. Esses pesos definem os parâmetros.

---

<sup>4</sup>Também é possível utilizar métodos mais sofisticados de remoção de viés para melhor desempenho. Por exemplo, o viés  $B_{ij}$ , específico do usuário i e do item j, pode ser calculado usando a abordagem discutida na seção 3.7.1. Esse viés é subtraído das entradas observadas e todas as entradas ausentes são inicializadas com 0s durante o pré-processamento. Após o cálculo das previsões, os vieses  $B_{ij}$  são adicionados novamente aos valores previstos durante o pós-processamento.

usado pelo algoritmo de aprendizagem. A arquitetura mais básica da rede neural é a perceptron, que contém um conjunto de nós de entrada e um nó de saída. Um exemplo de um perceptron é mostrado na Figura 3.3(a). Para um conjunto de dados contendo d dimensões diferentes, há sãos d unidades de entrada diferentes. O nó de saída está associado a um conjunto de pesos W, que é usado para calcular uma função  $f(\cdot)$  das d entradas. Um exemplo típico de tal função é a função linear assinada, que funcionaria bem para saída binária:

$$z_i = \text{sinal}\{W \cdot \overline{X_i} + b\} \quad (3.9)$$

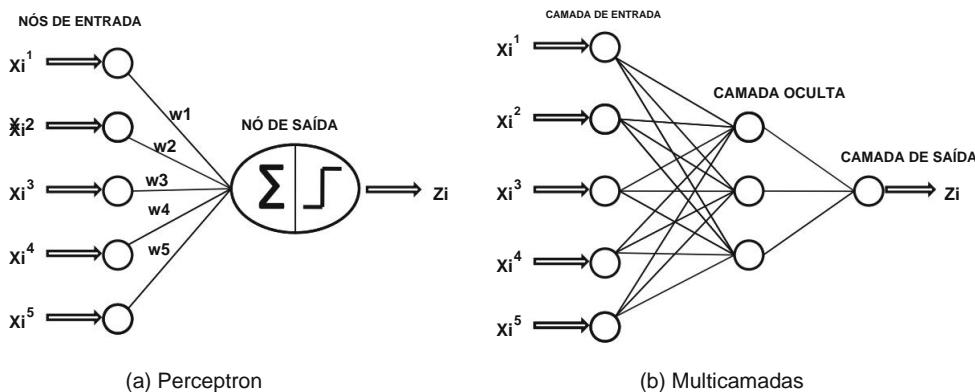


Figura 3.3: Redes neurais simples e multicamadas

		GLADIADOR		BEN-	PADRINHO	os
		U1	2		5	5
		U2		1	4	4
		U3	3		1	
		U4		5	1	
		U5	1	1	4	
		U6	5			1

CENTRALIZAÇÃO MÉDIA DE CADA LINHA  
E PREENCHA AS ENTRADAS QUE FALTARAM  
**COM ZERO**

		GLADIADOREN-		PADRINHØ		
		U1	-2	0	1	1
		U2	0	-2	1	1
		U3	1	0	-1	0
		U4	0	2	-2	0
		U5	-1	-1	2	0
		U6	2	0	0	-2

Figura 3.4: Pré-processamento da matriz de classificações. As entradas sombreadas são atualizadas iterativamente.

## 3.5. USANDO UM MODELO DE CLASSIFICAÇÃO ARBITRÁRIA COMO UMA CAIXA-PRETA 89

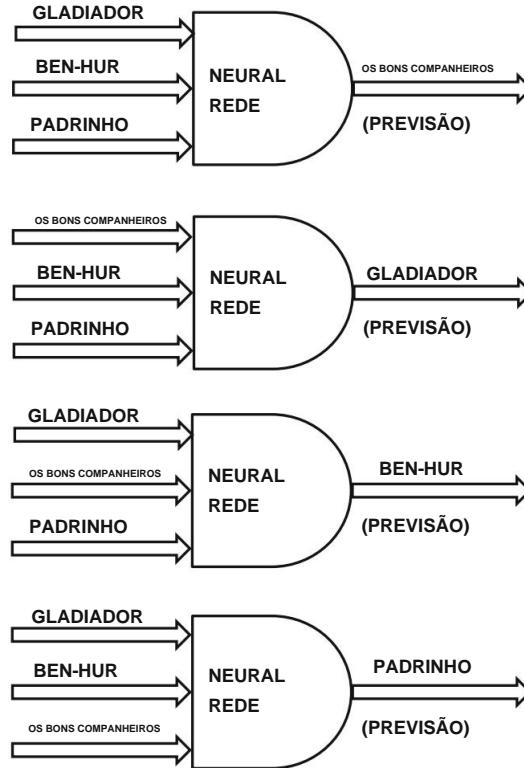


Figura 3.5: Redes neurais para previsão e atualização de entradas ausentes. Entradas sombreadas de A Figura 3.4 é atualizada iterativamente pelas redes neurais.

Aqui,  $\underline{X}_i$  é um vetor linha d-dimensional que define as d entradas da i-ésima instância de treinamento, e  $\underline{W}$  é o vetor de coeficientes. No contexto da filtragem colaborativa, as entradas d correspondem para os  $(n \times 1)$  itens, que são usados para prever a classificação do item restante. Suponha que o rótulo da i-ésima instância é  $y_i$ . No contexto da filtragem colaborativa,  $y_i$  representa as classificações observadas dos itens previstos. O parâmetro  $b$  denota o viés. Pode-se já percebermos a semelhança desta abordagem com a regressão linear embora a previsão A função é ligeiramente diferente. O valor de  $z_i$  é a saída prevista e o erro  $(y_i - z_i)^2$  desta saída prevista é usada para atualizar os pesos em  $\underline{W}$  de maneira semelhante à linear regressão. Esta atualização é semelhante às atualizações na descida do gradiente, que são feitas para Otimização por mínimos quadrados. No caso de redes neurais, a função de atualização é a seguinte:

$$\underline{\text{Peso}} + 1 = \underline{\text{Peso}} + \gamma(y_i - z_i)\underline{X}_i \quad (3.10)$$

Aqui,  $\gamma > 0$  denota a taxa de aprendizagem e  $\underline{W}_t$  é o valor do vetor de peso no t-ésima iteração. Não é difícil mostrar que o vetor de atualização incremental é o negativo gradiente de  $(y_i - z_i)^2$  em relação a  $\underline{W}$ . Iteraremos por todas as classificações observadas no item sendo previsto para fazer essas atualizações. Como foi assumido que  $y_i$  é binário, Esta abordagem foi projetada para matrizes de classificação binárias. Também é possível projetar redes neurais em que a saída não precisa ser binária e a função de previsão não precisa ser linear.

Em geral, uma rede neural pode ter múltiplas camadas, e os nós intermediários podem calcular funções não lineares. Um exemplo de tal rede neural multicamadas é ilustrado na Figura 3.3(b). É claro que tal rede teria um número maior de funções de aprendizagem.

parâmetros. O algoritmo de aprendizagem correspondente é denominado algoritmo de retropropagação [87]. A principal vantagem das redes neurais é que a arquitetura multicamadas oferece a capacidade de calcular funções não lineares complexas que não são facilmente computáveis com outros métodos de classificação. Portanto, as redes neurais também são chamadas de aproximadores de funções universais. Para dados ruidosos, como matrizes de classificação, a regularização pode ser usada para reduzir o impacto do ruído.

Considere uma matriz de classificação com quatro itens, ilustrada no lado esquerdo da Figura 3.4. Neste exemplo, os itens correspondem a filmes. O primeiro passo é centralizar cada linha, a fim de remover vieses do usuário. A matriz centrada na média resultante é mostrada no lado direito da Figura 3.4. Observe que os valores ausentes são substituídos pela média da linha correspondente, que é 0 após a centralização na média. Como há quatro itens, há quatro modelos de rede neural possíveis, em que cada modelo é construído usando as entradas de classificação dos outros três itens como colunas de treinamento e a quarta como coluna de teste. Essas quatro redes neurais são mostradas na Figura 3.5. A matriz completa da Figura 3.4 é usada para treinar cada uma dessas redes neurais na primeira iteração. Para cada coluna da matriz de classificação, a rede neural relevante na Figura 3.5 é usada para fins de previsão. As previsões resultantes feitas pelas redes neurais são então usadas para criar uma nova matriz na qual as entradas ausentes são atualizadas com os valores previstos. Em outras palavras, as redes neurais são usadas apenas para atualizar os valores nas entradas sombreadas da Figura 3.4 com o uso de um procedimento de treinamento e predição de rede neural pronto para uso. Após a atualização, as entradas sombreadas da Figura 3.4 não serão mais zeros. Esta matriz agora é usada para prever as entradas para a próxima iteração. Esta abordagem é repetida iterativamente até a convergência. Observe que cada iteração requer a aplicação de  $n$  procedimentos de treinamento, onde  $n$  é o número de itens. No entanto, não é necessário aprender os parâmetros das redes neurais do zero em cada iteração. Os parâmetros da iteração anterior podem ser usados como um bom ponto de partida. É importante usar regularização devido à alta dimensionalidade dos dados subjacentes [220].

Este modelo pode ser considerado um modelo item-wise, no qual as entradas representam as classificações de vários itens. Também é possível criar um modelo usuário-wise [679], no qual as entradas correspondem às classificações de vários usuários. O principal desafio com tal abordagem é que o número de entradas para a rede neural se torna muito grande. Portanto, é recomendado em [679] que nem todos os usuários devem ser usados como nós de entrada. Em vez disso, apenas usuários que classificaram pelo menos um número mínimo de itens são usados. Além disso, os usuários não devem ser todos muito semelhantes entre si. Portanto, heurísticas são propostas em [679] para pré-selecionar usuários mutuamente diversos na fase inicial. Esta abordagem pode ser considerada um tipo de seleção de características para redes neurais e também pode ser usada no modelo item-wise.

## 3.6 Modelos de Fatores Latentes

---

Na seção 2.5 do Capítulo 2, discutimos alguns métodos de redução de dimensionalidade para criar uma nova representação totalmente especificada de um conjunto de dados incompleto. No Capítulo 2, foram discutidos diversos métodos heurísticos que criam uma representação dimensional completa para permitir o uso de algoritmos de vizinhança [525]. Essas técnicas de redução de dados também são usadas para habilitar outros métodos baseados em modelos, que utilizam algoritmos de classificação como sub-rotina.

Portanto, em todos os métodos discutidos anteriormente, a redução de dimensionalidade desempenha apenas o papel de facilitar a criação de uma representação de dados mais conveniente para outros métodos baseados em modelos. Neste capítulo, métodos mais sofisticados serão discutidos, pois o objetivo é usar métodos de redução de dimensionalidade para estimar diretamente a matriz de dados de uma só vez.

As primeiras discussões sobre o uso de modelos de fatores latentes como um método direto para completar matrizes podem ser encontradas em [24, 525]. A ideia básica é explorar o fato de que porções significativas das linhas e colunas de matrizes de dados são altamente correlacionadas. Como resultado, os dados têm redundâncias incorporadas e a matriz de dados resultante é frequentemente aproximada muito bem por uma matriz de baixa classificação. Devido às redundâncias inerentes nos dados, a aproximação de baixa classificação totalmente especificada pode ser determinada mesmo com um pequeno subconjunto das entradas na matriz original. Essa aproximação de baixa classificação totalmente especificada frequentemente fornece uma estimativa robusta das entradas ausentes. A abordagem em [24] combina a técnica de maximização de expectativa (EM) com redução de dimensionalidade para reconstruir as entradas da matriz de dados incompleta.

Os modelos de fatores latentes são considerados o que há de mais moderno em sistemas de recomendação. Esses modelos utilizam métodos de redução de dimensionalidade bem conhecidos para preencher as entradas ausentes. Métodos de redução de dimensionalidade são comumente usados em outras áreas de análise de dados para representar os dados subjacentes em um pequeno número de dimensões. A ideia básica dos métodos de redução de dimensionalidade é rotacionar o sistema de eixos, de modo que as correlações pareadas entre as dimensões sejam removidas. A ideia-chave nos métodos de redução de dimensionalidade é que a representação reduzida, rotacionada e completamente especificada pode ser estimada de forma robusta a partir de uma matriz de dados incompleta. Uma vez que a representação completamente especificada tenha sido obtida, pode-se rotacioná-la de volta ao sistema de eixos original para obter a representação totalmente especificada [24]. Nos bastidores, os métodos de redução de dimensionalidade utilizam as correlações de linha e coluna para criar a representação totalmente especificada e reduzida.

Afinal, o uso de tais correlações é fundamental para todos os métodos de filtragem colaborativa, sejam eles métodos de vizinhança ou métodos baseados em modelos. Por exemplo, métodos de vizinhança baseados no usuário alavancam correlações entre usuários, enquanto métodos de vizinhança baseados em itens alavancam correlações entre itens. Os métodos de fatoração de matrizes fornecem uma maneira simples de alavancar todas as correlações de linha e coluna de uma só vez para estimar toda a matriz de dados.

Essa sofisticação da abordagem é uma das razões pelas quais os modelos de fatores latentes se tornaram o estado da arte em filtragem colaborativa. Para entender por que os modelos de fatores latentes são eficazes, forneceremos duas intuições, uma das quais é geométrica e a outra elucida diretamente a interpretação semântica. Ambas as intuições mostram como redundâncias em dados altamente correlacionados podem ser exploradas para criar uma aproximação de baixa classificação.

### 3.6.1 Intuição geométrica para modelos de fatores latentes

Primeiro, forneceremos uma intuição geométrica para modelos de fatores latentes, com base em uma discussão fornecida em [24]. Para entender a intuição de como as noções de baixa classificação, redundância e correlação estão relacionadas, considere uma matriz de classificações com três itens, na qual todos os três itens são positivamente correlacionados. Assuma um cenário de classificação de filme, no qual os três itens correspondem a Nero, Gladiador e Spartacus. Para facilitar a discussão, assuma que as classificações são valores contínuos, que se encontram no intervalo  $[y_1, 1]$ . Se as classificações forem positivamente correlacionadas, então o gráfico de dispersão tridimensional das classificações pode ser organizado aproximadamente ao longo de uma linha unidimensional, como mostrado na Figura 3.6. Como os dados são organizados principalmente ao longo de uma linha unidimensional, isso significa que a matriz de dados original tem uma classificação de aproximadamente 1 após a remoção das variações ruidosas. Por exemplo, a aproximação de classificação 1 da Figura 3.6 seria a linha unidimensional (ou vetor latente) que passa pelo centro dos dados e se alinha com a distribuição alongada dos dados. Observe que métodos de redução de dimensionalidade, como a Análise de Componentes Principais (ACP) e a Decomposição de Valor Singular (DV) (centrada na média), normalmente representam a projeção dos dados ao longo dessa linha como uma aproximação. Quando a matriz de classificações  $m \times n$  tem uma classificação de  $p \min\{m, n\}$  (após

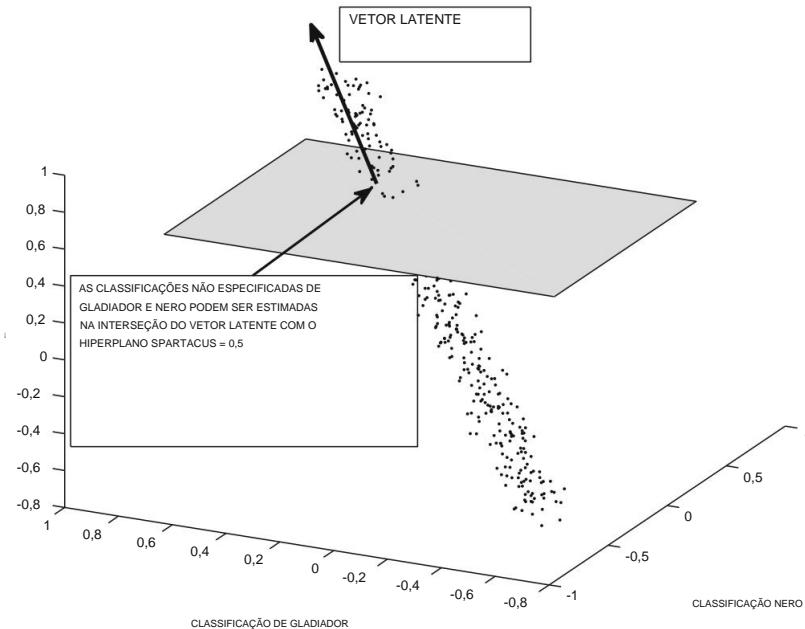


Figura 3.6: Aproveitando redundâncias baseadas em correlação na estimativa de dados ausentes para um usuário cuja única classificação especificada é um valor de 0,5 para o filme Spartacus

removendo variações ruidosas), os dados podem ser aproximadamente representados em um hiperplano  $p$ -dimensional. Nesses casos, as avaliações ausentes de um usuário podem frequentemente ser estimadas de forma robusta com apenas  $p$  entradas especificadas, desde que o hiperplano  $p$ -dimensional seja conhecido. Por exemplo, no caso da Figura 3.6, apenas uma avaliação precisa ser especificada para determinar as outras duas avaliações, porque a classificação da matriz de avaliações é apenas 1 após a remoção do ruído. Por exemplo, se a avaliação de Spartacus for fixada em 0,5, as avaliações de Nero e Gladiador podem ser estimadas<sup>5</sup> como a interseção do vetor latente unidimensional com o hiperplano paralelo ao eixo, no qual a avaliação de Spartacus é fixada em 0,5. Esse hiperplano é ilustrado na Figura 3.6. Portanto, métodos de redução de dimensionalidade, como SVD, alavancam as correlações e redundâncias entre atributos para inferir entradas não especificadas.

Neste caso, assumiu-se que uma matriz de dados especificada estava disponível para estimar o vetor latente relevante. Na prática, a matriz de dados não precisa ser totalmente especificada para estimar os vetores latentes dominantes, como a linha alinhada com a forma alongada da distribuição de dados na Figura 3.6. A capacidade de estimar esses vetores latentes com dados ausentes é a chave para o sucesso da abordagem do fator latente. A ideia básica em todos esses métodos é encontrar um conjunto de vetores latentes, no qual a distância média quadrada dos pontos de dados (representando avaliações individuais de usuários) do hiperplano definido por esses vetores latentes seja a menor possível. Portanto, devemos usar um conjunto de dados parcialmente especificado para recuperar o hiperplano de baixa dimensão no qual os dados se encontram aproximadamente. Fazendo isso, podemos capturar implicitamente as redundâncias subjacentes na estrutura de correlação dos dados e reconstruir todos os valores ausentes de uma só vez. É o conhecimento dessas redundâncias implícitas que nos ajuda a prever as entradas ausentes na matriz. Vale ressaltar que se os dados não tiverem correlações ou redundâncias, um modelo de fator latente simplesmente não funcionará.

<sup>5</sup>Uma descrição detalhada do método usado para realizar esta estimativa em vários cenários é discutida na seção 3.6.5.3.

### 3.6.2 Intuição de baixo nível para modelos de fatores latentes

A intuição geométrica da seção anterior é útil para compreender o impacto dos vetores latentes quando são mutuamente ortogonais. No entanto, os vetores latentes nem sempre são mutuamente ortogonais. Nesses casos, é útil obter alguma intuição da álgebra linear.

Uma maneira de compreender a eficácia dos modelos de fatores latentes é examinar o papel que a fatoração desempenha nessas matrizes. A fatoração é, na verdade, uma maneira mais geral de aproximar uma matriz quando ela está propensa à redução de dimensionalidade devido a correlações entre colunas (ou linhas). A maioria dos métodos de redução de dimensionalidade também pode ser expressa como fatorações de matrizes.

Primeiro, consideremos o caso simples em que todas as entradas na matriz de classificações  $R$  são observadas. A ideia principal é que qualquer matriz  $R$   $m \times n$  de classificação  $k \leq \min\{m, n\}$  pode sempre ser expressa na seguinte forma de produto de fatores de classificação  $k$ :

$$R = UV^T \quad (3.11)$$

Aqui,  $U$  é uma matriz  $m \times k$ , e  $V$  é uma matriz  $n \times k$ . Observe que o posto do espaço-linha 6 e do espaço-coluna de  $R$  é  $k$ . Cada coluna de  $U$  pode ser vista como um dos  $k$  vetores-base do espaço-coluna  $k$ -dimensional de  $R$ , e a  $j$ -ésima linha de  $V$  contém os coeficientes correspondentes para combinar esses vetores-base na  $j$ -ésima coluna de  $R$ .

Alternativamente, pode-se visualizar as colunas de  $V$  como os vetores base do espaço linha de  $R$ , e as linhas de  $U$  como os coeficientes correspondentes. A capacidade de fatorar qualquer matriz de posto- $k$  nesta forma é um fato fundamental da álgebra linear [568], e há um número infinito dessas fatorações correspondendo a vários conjuntos de vetores base. SVD é um exemplo de tal fatoração em que os vetores base representados pelas colunas de  $U$  (e as colunas de  $V$ ) são ortogonais entre si.

Mesmo quando a matriz  $R$  tem classificação maior que  $k$ , ela pode frequentemente ser expressa aproximadamente como o produto de fatores de classificação- $k$ :

$$R \approx UV^T \quad (3.12)$$

Como antes,  $U$  é uma matriz  $m \times k$  e  $V$  é uma matriz  $n \times k$ . O erro desta aproximação  $\cdot \|2$  representa a é igual a  $\|R - UV\|_F^2$ , onde  $\|\cdot\|_F$  soma dos quadrados das entradas em residual resultante ( $R - UV$  norma de  $\|\cdot\|_F^2$ ). Esta quantidade também é chamada de (quadrado) Frobenius da matriz residual. A matriz residual normalmente representa o ruído na matriz de classificações subjacente, que não pode ser modelada pelos fatores de baixa classificação. Para simplificar a discussão, vamos considerar o caso direto em que  $R$  é totalmente observado.

Primeiro, examinaremos a intuição por trás do processo de fatoração e, em seguida, discutiremos a implicação dessa intuição no contexto de matrizes com entradas ausentes.

Qual é a implicação do processo de fatoração e seu impacto em uma matriz com linhas e colunas altamente correlacionadas? Para entender esse ponto, considere a matriz de classificações ilustrada na Figura 3.7. Nesta figura, uma matriz de classificações  $7 \times 6$  com 7 usuários e 6 itens é ilustrada. Todas as classificações são derivadas de  $\{1, -1, 0\}$ , que correspondem a gostar, não gostar e neutralidade. Os itens são filmes e pertencem aos gêneros romance e história, respectivamente. Um dos filmes, intitulado Cleópatra, pertence a ambos os gêneros. Devido à natureza dos gêneros dos filmes subjacentes, os usuários também mostram tendências claras em suas classificações.

Por exemplo, os usuários de 1 a 3 geralmente gostam de filmes históricos, mas são neutros em relação ao gênero romântico. O usuário 4 gosta de filmes de ambos os gêneros. Os usuários de 5 a 7 gostam de filmes pertencentes ao gênero romântico, mas explicitamente não gostam de filmes históricos. Observe que esta matriz tem uma diferença significativa

---

60 O espaço de linhas de uma matriz é definido por todas as combinações lineares possíveis das linhas da matriz. O espaço de colunas de uma matriz é definido por todas as combinações lineares possíveis das colunas da matriz.

número de correlações entre usuários e itens, embora as avaliações de filmes pertencentes aos dois gêneros distintos pareçam ser relativamente independentes. Como resultado, essa matriz pode ser fatorada aproximadamente em fatores de classificação 2, como mostrado na Figura 3.7(a). A matriz U é uma matriz  $7 \times 2$ , que mostra a propensão dos usuários aos dois gêneros, enquanto a matriz V é uma matriz  $6 \times 2$ , que mostra a pertença dos filmes aos dois gêneros.

Em outras palavras, a matriz U fornece a base para o espaço de colunas, enquanto a matriz V fornece a base para o espaço de linhas. Por exemplo, a matriz U mostra que o usuário 1 gosta de filmes históricos, enquanto o usuário 4 gosta de ambos os gêneros. Uma inferência semelhante pode ser feita usando as linhas de V. As colunas de V correspondem aos vetores latentes, como os mostrados na Figura 3.6. Ao contrário de SVD, no entanto, os vetores latentes neste caso não são mutuamente ortogonais.

A matriz residual correspondente para a fatoração é mostrada na Figura 3.7(b). A matriz residual normalmente corresponde às avaliações dos usuários para Cleópatra, que não seguem o padrão definido. É importante ressaltar que, em aplicações do mundo real, as entradas da matriz nos fatores são tipicamente números reais (em vez de integrais). Um exemplo com fatores integrais é mostrado aqui para simplificar a visualização. Além disso, uma interpretação semântica clara dos fatores em termos de gêneros ou categorias às vezes não é possível, especialmente quando os fatores contêm valores positivos e negativos. Por exemplo, se multiplicarmos U e V por -1 na Figura 3.7, a fatoração ainda é válida, mas a interpretação se torna mais difícil. No entanto, as k colunas de U e V representam correlações-chave entre os usuários e os itens, respectivamente, e podem ser vistas abstratamente como conceitos latentes, sejam ou não semanticamente interpretáveis. Em algumas formas de fatoração, como a fatoração de matriz não negativa, a interpretabilidade desses conceitos é preservada em maior grau.

Neste exemplo, a matriz R foi totalmente especificada e, portanto, a fatoração não é particularmente útil da perspectiva da estimativa de valores ausentes. A principal utilidade da abordagem surge quando a matriz R não é totalmente especificada, mas ainda é possível estimar de forma robusta todas as entradas das matrizes U e V, respectivamente. Para valores baixos da classificação, isso ainda é possível a partir de dados esparsamente especificados. Isso ocorre porque não são necessárias muitas entradas observadas para estimar os fatores latentes a partir de dados inherentemente redundantes. Uma vez que as matrizes U e V tenham sido estimadas, toda a matriz de classificações pode ser estimada como  $\hat{R} = UV^T$ .

de uma só vez, o que fornece todas as classificações que faltam.

### 3.6.3 Princípios básicos de fatoração de matrizes

No modelo básico de fatoração matricial, a matriz de classificações  $m \times n$  R é aproximadamente fatorada em uma matriz  $m \times k$  U e uma matriz  $n \times k$  V

, do seguinte modo:

$$R \approx UV^T \quad (3.13)$$

Cada coluna de U (ou V) é chamada de vetor latente ou componente latente, enquanto cada linha de U (ou V) é chamada de fator latente. A i-ésima linha  $u_i$  de U é chamada de fator do usuário i e contém k entradas correspondentes à afinidade do usuário i em relação aos k conceitos na matriz de classificações. Por exemplo, no caso da Figura 3.7,  $u_i$  é um vetor bidimensional que contém a afinidade do usuário i em relação aos gêneros de história e romance na matriz de classificações. Da mesma forma, cada linha  $v_j$  de V é chamada de fator do item j e representa a afinidade do j-ésimo item em relação a esses k conceitos. Na Figura 3.7, o fator do item j contém a afinidade do item j em relação às duas categorias de filmes.

Da Equação 3.13, segue-se que cada classificação  $r_{ij}$  em R pode ser expressa aproximadamente como um produto escalar do i-ésimo fator do usuário i e do j-ésimo fator do item j:

$$r_{ij} \approx u_i \cdot v_j \quad (3.14)$$

## 3.6. MODELOS DE FATORES LATENTES

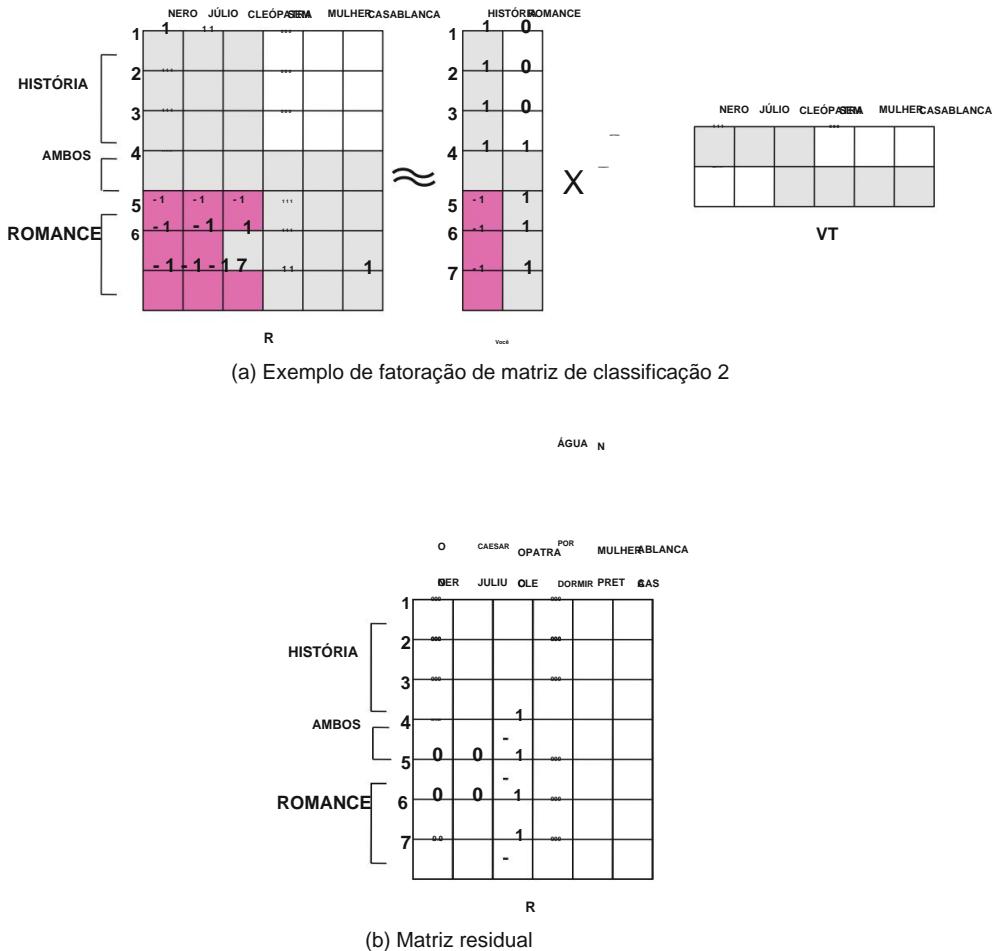


Figura 3.7: Exemplo de fatoração de uma matriz e sua matriz residual

Como os fatores latentes  $u_i = (\bar{u_{i1}} \dots \bar{u_{ik}})$  e  $v_j = (\bar{v_{j1}} \dots \bar{v_{jk}})$  podem ser vistos como as afinidades dos usuários por  $k$  conceitos diferentes, uma maneira intuitiva de expressar a Equação 3.14 seria a seguinte:

$$\begin{aligned} r_{ij} \hat{y} &= \sum_{s=1}^k \text{interfaces de usuário - } v_{js} \\ &= \sum_{s=1}^k (\text{Afinidade do usuário } i \text{ com o conceito } s) \times (\text{Afinidade do item } j \text{ com o conceito } s) \end{aligned}$$

No caso da Figura 3.7, os dois conceitos no somatório acima mencionado correspondem aos gêneros romance e histórico. Portanto, o somatório pode ser expresso da seguinte forma:

$$\begin{aligned} r_{ij} \hat{y} &= (\text{Afinidade do usuário } i \text{ com o histórico}) \times (\text{Afinidade do item } j \text{ com o histórico}) \\ &\quad + (\text{Afinidade do usuário } i \text{ com romance}) \times (\text{Afinidade do item } j \text{ com romance}) \end{aligned}$$

É importante ressaltar que a noção de conceitos muitas vezes não é semanticamente interpretável, como ilustrado na Figura 3.7. Um vetor latente pode frequentemente ser um vetor arbitrário de valores positivos e negativos, o que dificulta sua interpretação semântica. No entanto, ele representa um padrão de correlação dominante na matriz de classificações, assim como o vetor latente da Figura 3.6 representa um padrão de correlação geométrica. Como veremos mais adiante, algumas formas de fatoração, como a fatoração de matrizes não negativas, são explicitamente projetadas para alcançar maior interpretabilidade nos vetores latentes.

As principais diferenças entre os vários métodos de fatoração de matrizes surgem em termos das restrições impostas a  $U$  e  $V$  (por exemplo, ortogonalidade ou não negatividade dos vetores latentes) e da natureza da função objetivo (por exemplo, minimizar a norma de Frobenius ou maximizar a estimativa de verossimilhança em um modelo gerativo). Essas diferenças desempenham um papel fundamental na usabilidade do modelo de fatoração de matrizes em diversos cenários do mundo real.

### 3.6.4 Fatoração de matrizes irrestritas

A forma mais fundamental de fatoração de matrizes é o caso irrestrito, no qual nenhuma restrição é imposta às matrizes fatoriais  $U$  e  $V$ . Grande parte da literatura de recomendações se refere à fatoração de matrizes irrestritas como decomposição em valor singular (SVD). Estritamente falando, isso é tecnicamente incorreto; em SVD, as colunas de  $U$  e  $V$  devem ser ortogonais. No entanto, o uso do termo "SVD" para se referir à fatoração de matrizes irrestritas<sup>7</sup> é bastante difundido na literatura de recomendações, o que causa alguma confusão para profissionais de fora da área. Neste capítulo, nos desviaremos dessa prática incorreta e trataremos a fatoração de matrizes irrestritas e a SVD de uma maneira distinta.

Esta seção discutirá a fatoração de matrizes irrestrita, e a seção seguinte discutirá a SVD.

Antes de discutir a fatoração de matrizes incompletas, vamos primeiro abordar o problema da fatoração de matrizes totalmente especificadas. Como determinar as matrizes fatoriais  $U$  e  $V$ ?

---

<sup>7</sup>Em SVD [568], os vetores base também são chamados de vetores singulares, que, por definição, devem ser mutuamente ortonormais.

## 3.6. MODELOS DE FATORES LATENTES

para que a matriz R totalmente especificada corresponda a  $\bar{U}V$  o mais próximo possível? Pode-se formular um problema de otimização em relação às matrizes U e V para atingir este objetivo:

$$\text{Minimize } J = \frac{1}{2} \|R - UV\|_F^2$$

sujeito a:

Sem restrições em U e V

Aqui,  $\|\cdot\|_F^2$  representa a norma de Frobenius ao quadrado da matriz, que é igual à soma dos quadrados das entradas da matriz. Assim, a função objetivo é igual à soma dos quadrados das entradas na matriz residual  $(R - UV)^T$ . Quanto menor for a função objetivo ou seja, quanto melhor for a qualidade da fatoração  $R \approx UV$ , ela  $\|R - UV\|_F^2$  será. Esta função objetivo pode ser vista como uma função de perda quadrática, que quantifica a perda de precisão na estimativa a matriz R com o uso de fatoração de baixa classificação. Uma variedade de métodos de gradiente descendente pode ser usado para fornecer uma solução ótima para essa fatoração.

No entanto, no contexto de uma matriz com entradas ausentes, apenas um subconjunto das entradas de R são conhecidos. Portanto, a função objetivo, como escrita acima, também é indefinida. Afinal, não se pode calcular a norma de Frobenius de uma matriz na qual algumas das entradas estão faltando! A função objetivo, portanto, precisa ser reescrita apenas em termos de entradas observadas para aprender U e V. A parte boa desse processo é que uma vez os fatores latentes U e V são aprendidos, toda a matriz de classificações pode ser reconstruída como  $UV^T$  de uma só vez.

Seja o conjunto de todos os pares usuário-item  $(i, j)$ , que são observados em R, denotado por S. Aqui,  $i \in \{1 \dots m\}$  é o índice de um usuário e  $j \in \{1 \dots n\}$  é o índice de um item. Portanto, o conjunto S de pares usuário-item observados é definido da seguinte forma:

$$S = \{(i, j) : r_{ij} \text{ é observado}\} \quad (3.15)$$

Se pudermos de alguma forma fatorar a matriz incompleta R como o produto aproximado  $UV$  de matrizes totalmente especificadas  $U = [uis]_{m \times k}$  e  $V = [vjs]_{n \times k}$ , então todas as entradas em R podem ser também previsto. Especificamente, a entrada  $(i, j)$  da matriz R pode ser prevista da seguinte forma:

$$\hat{r}_{ij} = \sum_{s=1}^k u_{is} v_{js} \quad (3.16)$$

Observe o símbolo do "chapéu" (ou seja, circunflexo) na classificação do lado esquerdo para indicar que é um valor previsto e não um valor observado. A diferença entre o valor observado e o valor previsto de uma entrada especificada  $(i, j)$  é dado por  $e_{ij} = (r_{ij} - \hat{r}_{ij}) = (r_{ij} - \sum_{s=1}^k u_{is} v_{js})$ . Em seguida, a função objetivo modificada, que trabalha com matrizes incompletas, é calculada somente sobre as entradas observadas em S da seguinte forma:

$$\text{Minimizar } J = \frac{1}{2} \sum_{(i,j) \in S} e_{eu}^2 = \frac{1}{2} \sum_{(i,j) \in S} (r_{ij} - \sum_{s=1}^k u_{is} v_{js})^2$$

sujeito a:

Sem restrições em U e V

Note que a função objetivo mencionada soma o erro apenas sobre o observado entradas em S. Além disso, cada um dos termos  $(r_{ij} - \sum_{s=1}^k u_{is} v_{js})^2$  é o erro quadrado  $e_{eu}$  entre os valores observados e previstos da entrada  $(i, j)$ . Aqui,  $uis$  e  $vjs$  são os

Algoritmo GD (Matriz de classificações: R, Taxa de aprendizagem:  $\eta$ )  
 começa

  Inicializar aleatoriamente as matrizes U e V;

$S = \{(i, j) : r_{ij} \text{ é observado}\};$  enquanto

  not(convergência) começa

    Calcule cada erro  $e_{ij} \in S$  como as entradas observadas de R  $\eta$  UV para cada par usuário-componente  $(i, q)$  faça  $u_+$   
 $\sum_{j:(i,j)\in S} e_{ij} \cdot v_{jq} + \eta \cdot \sum_{j:(i,j)\in S} e_{ij} \cdot u_{iq}$   
 usuário-componente  $(j, q)$  faça  $v_+ \eta v_{jq} + \eta \cdot \sum_{i:(i,j)\in S} e_{ij} \cdot u_{iq}$   
 componente  $(i, q)$  faça  $u_{iq} \eta u_+ + iq$ ; para cada par item-componente  
 $(j, q)$  faça  $v_{jq} \eta v_+ + jq$ ;  
 Verifique a condição de convergência;

  fim

do fim

Figura 3.8: Descida do gradiente

variáveis desconhecidas, que precisam ser aprendidas para minimizar a função objetivo. Isso pode ser alcançado simplesmente com métodos de descida de gradiente. Portanto, é necessário calcular a derivada parcial de J em relação às variáveis de decisão  $u_{iq}$  e  $v_{jq}$ :

$$\begin{aligned} \frac{\partial J}{\partial u_{iq}} &= \sum_{j:(i,j)\in S} r_{ij} \eta \sum_{s=1}^k u_{is} \cdot v_{js} (\eta v_{jq}) \eta \{1 \dots m\}, q \eta \{1 \dots k\} \\ &= (e_{ij})(\eta v_{jq}) \eta \{1 \dots m\}, q \eta \{1 \dots k\} j:(i,j)\in S \\ \frac{\partial J}{\partial v_{jq}} &= \sum_{i:(i,j)\in S} r_{ij} \eta \sum_{s=1}^k u_{is} \cdot v_{js} (\eta u_{iq}) \eta \{1 \dots n\}, q \eta \{1 \dots k\} \\ &= (e_{ij})(\eta u_{iq}) \eta \{1 \dots n\}, q \eta \{1 \dots k\} i:(i,j)\in S \end{aligned}$$

Note que o vetor inteiro de derivadas parciais nos fornece o gradiente em relação ao vetor de variáveis de decisão ( $m \cdot k + n \cdot k$ ) nas matrizes U e V. Seja este vetor gradiente denotado por  $\hat{J}$ . Seja o vetor de variáveis de decisão ( $m \cdot k + n \cdot k$ ) correspondentes às entradas em U e V denotado por VAR. Então, pode-se atualizar o vetor inteiro de variáveis de decisão como VAR  $\eta$  VAR  $\eta \hat{J}$ . Aqui,  $\eta > 0$  é o tamanho do passo, que pode ser escolhido usando métodos numéricos padrão em programação não linear [76]. Em muitos casos, os tamanhos dos passos são definidos para pequenos valores constantes. As iterações são executadas até a convergência. Essa abordagem é chamada de descida do gradiente. A estrutura algorítmica para descida do gradiente é ilustrada na Figura 3.8. Vale ressaltar que as variáveis intermediárias  $u_+$  e  $v_+$  são usadas para garantir que todas as atualizações nas entradas em U e V sejam realizadas simultaneamente.

$e_{ij}$   $v_{jq}$   $s$  são

Também é possível realizar as atualizações da Figura 3.8 usando uma representação matricial. A primeira passo é calcular uma matriz de erro  $E = R - UV$  entradas que na qual as entradas não observadas de E (ou seja, não estão em S) são definidas como 0. Observe que E é uma matriz muito esparsa e faz sentido calcular o valor de  $e_{ij}$  apenas para as entradas observadas  $(i, j) \in S$  e armazenar a matriz usando

### 3.6. MODELOS DE FATORES LATENTES

uma estrutura de dados esparsa. Posteriormente, as atualizações podem ser calculadas da seguinte forma:

$$\mathbf{U} \leftarrow \mathbf{U} + \hat{\mathbf{y}}\mathbf{EV}$$

$$\mathbf{V} \leftarrow \mathbf{V} + \hat{\mathbf{y}}\mathbf{ETU}$$

Essas atualizações podem ser executadas até a convergência, tomando-se o cuidado de atualizar todas as entradas em ambas as matrizes simultaneamente com o uso de variáveis intermediárias (como na Figura 3.8).

#### 3.6.4.1 Descida do gradiente estocástico

O método mencionado é chamado de método de atualização em lote. Uma observação importante é que as atualizações são funções lineares dos erros nas entradas observadas da matriz de classificações. A atualização pode ser executada de outras maneiras, decompondo-a em componentes menores associados aos erros em entradas observadas individuais, em vez de em todas as entradas.

Esta atualização pode ser aproximada estocasticamente em termos do erro em uma entrada observada (escolhida aleatoriamente)  $(i, j)$  da seguinte forma:

$$\begin{array}{lll} \text{uiq} \leftarrow \text{uiq} - \frac{\hat{y}_j}{\hat{y}_{uiq}} & \text{Porção contribuída por } (i, j) & \hat{y}_q \leftarrow \{1 \dots k\} \\ \text{vjq} \leftarrow \text{vjq} - \frac{\hat{y}_j}{\hat{y}_{vjq}} & \text{Porção contribuída por } (i, j) & \hat{y}_q \leftarrow \{1 \dots k\} \end{array}$$

É possível percorrer as entradas observadas em  $\mathbf{R}$ , uma de cada vez (em ordem aleatória) e atualizar apenas o conjunto relevante de  $2 \cdot k$  entradas nas matrizes de fatores, em vez de todas as entradas ( $m \cdot k + n \cdot k$ ) nas matrizes de fatores. Nesse caso, as  $2 \cdot k$  atualizações específicas para a entrada observada  $(i, j) \in \mathbf{S}$  são as seguintes:

$$\begin{aligned} \text{uiq} \leftarrow \text{uiq} - \hat{y} \cdot \mathbf{eij} \cdot \mathbf{vjq} & \hat{y} \leftarrow \{1 \dots k\} \\ \cdot \mathbf{eij} \cdot \text{uiq} & \hat{y} \leftarrow \{1 \dots k\} \end{aligned}$$

Para cada classificação observada  $r_{ij}$ , o erro  $e_{ij}$  é usado para atualizar as  $k$  entradas na linha  $i$  de  $\mathbf{U}$  e as  $k$  entradas na linha  $j$  de  $\mathbf{V}$ . Observe que  $e_{ij} \cdot v_{jq}$  é o componente da derivada parcial de  $J$  em relação a  $u_{iq}$ , que é específico para uma única entrada observada  $(i, j)$ . Para melhor eficiência, cada uma dessas  $k$  entradas pode ser atualizada simultaneamente na forma vetorizada. Seja  $u_i$  a  $i$ -ésima linha de  $\mathbf{U}$  e  $v_j$  a  $j$ -ésima linha de  $\mathbf{V}$ . Então, as atualizações mencionadas acima podem ser reescritas na forma vetorizada  $k$ -dimensional da seguinte forma:

$$\begin{aligned} \mathbf{u}_i \leftarrow \mathbf{u}_i + \hat{y} \mathbf{eij} \mathbf{v}_j \\ \mathbf{v}_j \leftarrow \mathbf{v}_j + \hat{y} \mathbf{eij} \mathbf{u}_i \end{aligned}$$

Percorremos todas as entradas observadas várias vezes (ou seja, usamos múltiplas iterações) até que a convergência seja alcançada. Essa abordagem é chamada de gradiente descendente estocástico, na qual o gradiente é aproximado por aquele calculado com base no erro de uma única entrada escolhida aleatoriamente na matriz. O pseudocódigo para o método do gradiente descendente estocástico é ilustrado na Figura 3.9. Vale ressaltar que variáveis temporárias  $u_+$  são usadas para armazenar resultados intermediários durante uma atualização, de modo que as atualizações  $2 \cdot k$  não afetem umas às outras. Essa é uma abordagem geral que deve ser usada em todas as atualizações por grupo discutidas neste livro, embora possamos não declará-la explicitamente.

Na prática, a convergência mais rápida é alcançada pelo método de descida de gradiente estocástico em comparação ao método em lote, embora a convergência seja muito mais suave neste último.

```

Algoritmo SGD(Matriz de classificações: R, Taxa de aprendizagem:  $\eta$ )
begin
    Inicializa aleatoriamente as matrizes U e V; S =
    {(i, j): rij é observado}; enquanto
    not(convergência) do begin

        Embaralha aleatoriamente as entradas observadas
        em S; para cada (i, j) ∈ S em ordem embaralhada
        do begin
             $e_{ij} \leftarrow r_{ij}$ ;  $s=1$  a  $k$ ; para cada
             $q \in \{1 \dots k\}$  do  $u_{iq} = u_{iq} - \eta \cdot e_{ij} \cdot v_{jq}$ ;
            para cada  $q \in \{1 \dots k\}$  faça  $v_{jq} = v_{jq} - \eta \cdot e_{ij} \cdot u_{iq}$ ; para cada  $q \in \{1 \dots k\}$  faça  $u_{iq} = u_{iq} + \eta \cdot e_{ij} \cdot v_{jq}$ ;
            fim
            Verifique a condição de convergência;
        fim
    do fim

```

Figura 3.9: Descida do gradiente estocástico

Isso ocorre porque as entradas de U e V são atualizadas simultaneamente neste último caso, com o uso de todas as entradas observadas, em vez de uma única entrada observada escolhida aleatoriamente. Essa aproximação ruidosa da descida do gradiente estocástico pode, às vezes, impactar a qualidade da solução e a suavidade da convergência. Em geral, a descida do gradiente estocástico é preferível quando o tamanho dos dados é muito grande e o tempo computacional é o principal gargalo. Em outros métodos de "compromisso", minilotes são usados, nos quais um subconjunto de entradas observadas é usado para construir a atualização. Esses diferentes métodos fornecem diferentes compensações entre a qualidade da solução e a eficiência computacional.

À medida que se percorrem repetidamente as entradas observadas na matriz para atualizar as matrizes fatoriais, a convergência será eventualmente alcançada. Em geral, sabe-se que o método global garante a convergência, embora seja geralmente mais lento que o método local. Um valor típico do tamanho do passo (ou taxa de aprendizado) é um pequeno valor constante, como  $\eta = 0,005$ .

Uma abordagem mais eficaz para evitar mínimos locais e acelerar a convergência é usar o algoritmo de driver em negrito [58, 217] para selecionar  $\eta$  de forma adaptativa em cada iteração. Também é possível, em princípio, usar diferentes tamanhos de passo para diferentes fatores [586]. Uma observação interessante sobre alguns desses modelos é que executá-los até a convergência por muitas iterações pode, às vezes, levar a uma ligeira piora na qualidade da solução nas entradas não observadas.

Portanto, às vezes é aconselhável não definir os critérios de convergência de forma muito rígida.

Outro problema com esses modelos de fatores latentes é a inicialização. Por exemplo, é possível inicializar as matrizes de fatores com números pequenos em  $(\bar{y}_1, 1)$ . No entanto, a escolha da inicialização pode afetar a qualidade da solução final. É possível usar diversas heurísticas para melhorar a qualidade. Por exemplo, é possível usar algumas heurísticas simples baseadas em SVD, discutidas posteriormente nesta seção, para criar uma inicialização aproximada.

### 3.6.4.2 Regularização

Um dos principais problemas dessa abordagem surge quando a matriz de classificação R é esparsa e relativamente poucas entradas são observadas. Isso ocorre quase sempre em cenários reais.

Nesses casos, o conjunto S de classificações observado é pequeno, o que pode causar sobreajuste. Observe que o sobreajuste também é um problema comum na classificação quando os dados de treinamento são limitados. Uma abordagem comum para lidar com esse problema é usar a regularização. A regularização reduz a tendência do modelo a se ajustar excessivamente, introduzindo um viés<sup>8</sup> no modelo.

Na regularização, a ideia é desencorajar valores muito grandes dos coeficientes em U e V ( $\|U\|_F^2 + \|V\|_F^2$ ), adicionados Portanto, um termo de regularização, para a função objetivo, onde  $\tilde{\gamma} > 0$  é o parâmetro para incentivar a estabilidade. de regularização. Aqui,  $\|\cdot\|_F^2$  denota a norma de Frobenius (ao quadrado) da matriz. A ideia básica é criar um viés em favor de soluções mais simples, penalizando coeficientes altos. Esta é uma abordagem padrão, usada em muitas formas de classificação e regressão, e também alavancada pela filtragem colaborativa.

O parâmetro  $\tilde{\gamma}$  é sempre não negativo e controla o peso do termo de regularização. O método para escolher  $\tilde{\gamma}$  será discutido posteriormente nesta seção.

Como no caso anterior, suponha que  $e_{ij} = (r_{ij} \sum_{k=1}^K u_{ik} v_{kj}) - y_{ij}$  representa a diferença entre o valor observado e o valor previsto da entrada especificada  $(i, j)$   $\in S$ . A função objetivo regularizada é a seguinte:

$$\begin{aligned} \text{Minimizar } J = & \frac{1}{2} \sum_{(i,j) \in S} e_{ij}^2 + \frac{\tilde{\gamma}}{2} \left( \sum_{i=1}^m \sum_{s=1}^k u_{is}^2 + \sum_{j=1}^n \sum_{s=1}^k v_{js}^2 \right) \\ & = \frac{1}{2} \sum_{(i,j) \in S} r_{ij} (\hat{y}_{ij} - \sum_{s=1}^k u_{is} v_{js})^2 + \frac{\tilde{\gamma}}{2} \left( \sum_{i=1}^m \sum_{s=1}^k u_{is}^2 + \sum_{j=1}^n \sum_{s=1}^k v_{js}^2 \right) \end{aligned}$$

Ao tomar a derivada parcial de  $J$  em relação a cada uma das variáveis de decisão, obtém-se quase o mesmo resultado que o caso não regularizado, exceto que os termos  $\tilde{\gamma} u_{is}$  e  $\tilde{\gamma} v_{js}$ , respectivamente, são adicionados aos gradientes correspondentes nos dois casos.

$$\begin{aligned} \frac{\partial J}{\partial u_{is}} &= \sum_{j=1}^n r_{ij} (\hat{y}_{ij} - \sum_{s=1}^k u_{is} v_{js}) v_{js} \\ &= (e_{ij}) (\hat{y}_{ij}) + \tilde{\gamma} u_{is} \quad \forall i \in \{1 \dots m\}, s \in \{1 \dots k\} \\ \frac{\partial J}{\partial v_{js}} &= \sum_{i=1}^m r_{ij} (\hat{y}_{ij} - \sum_{s=1}^k u_{is} v_{js}) u_{is} \\ &= (e_{ij}) (\hat{y}_{ij}) + \tilde{\gamma} v_{js} \quad \forall j \in \{1 \dots n\}, s \in \{1 \dots k\} \end{aligned}$$

As etapas para executar a descida do gradiente permanecem semelhantes às discutidas no caso sem regularização. Tanto o método em lote quanto o método local podem ser usados. Por exemplo, considere o método de atualização global. Seja o vetor de variáveis de decisão  $(m \cdot k + n \cdot k)$  correspondentes às entradas em U e V denotado por  $VAR$  e seja o vetor gradiente correspondente denotado por  $\hat{J}$ . Então, pode-se atualizar todo o vetor de variáveis de decisão como  $VAR \leftarrow VAR - \hat{J}$ . Isso pode ser efetivamente alcançado modificando o

<sup>8</sup>Consulte o Capítulo 6 para uma discussão sobre o trade-off entre viés e variância.

Atualizações (não regularizadas) na Figura 3.8 para incluir termos de regularização. As atualizações modificadas podem ser escritas da seguinte forma:

uiq ŷ uiq + ŷ	ŷ ŷ j:(i,j)ŷS	eij · vjq ŷ ŷ · uiq ŷ ŷ	ŷq ŷ {1 ...k}
vjq ŷ vjq + ŷ	ŷ ŷ i:(i,j)ŷS	eij · uiq ŷ ŷ · vjq ŷ ŷ	ŷq ŷ {1 ...k}

As atualizações podem ser executadas até a convergência. Também é possível escrever essas atualizações em termos da matriz de erro  $m \times n$   $E = [e_{ij}]$ , na qual as entradas não observadas de  $E$  são definidas como 0:

você  $\hat{y}$  você(1  $\hat{y}$   $\hat{y}$  ·  $\hat{y}$ ) +  $\hat{y}$ EV

$$\nabla \cdot \nabla (1 - \nabla \cdot \nabla) + \nabla \cdot \nabla \times$$

Observe que o termo multiplicativo  $(1 - \hat{y}_j \cdot \hat{y}_i)$  reduz os parâmetros em cada etapa, o que é resultado da regularização. Se a forma matricial for usada para atualizações, deve-se tomar cuidado para calcular e usar representações esparsas de  $E$ . Faz sentido calcular o valor de  $e_{ij}$  apenas para as entradas observadas  $(i, j) \in S$  e armazenar  $E$  usando uma estrutura de dados esparsa.

No caso de atualizações locais (ou seja, descida de gradiente estocástico), as derivadas parciais são calculadas em relação ao erro em uma entrada observada escolhida aleatoriamente ( $i, j$ ), em vez de em todas as entradas. As seguintes  $2 \cdot k$  atualizações podem ser executadas para cada entrada observada ( $i, j$ ) e  $S$ , que são processadas em ordem aleatória:

uiq ý uiq + ý(eij · vjq ý ý · uiq) ýq ý {1 ...k} vjq ý vjq + ý(eij  
uiq ý ý · vjq) ýq ý {1 ...k}

Para melhor eficiência, essas atualizações são executadas de forma vetorizada sobre os vetores de fatores k-dimensionalas do usuário i e do item j, da seguinte forma:

ui ÿ ui + ÿ(eijvj ÿ ÿui) vj ÿ vj +  
ÿ(eijui ÿ ÿvj ) — —

Essas atualizações são usadas dentro da estrutura do algoritmo descrito na Figura 3.9. Vale ressaltar que as atualizações locais não são exatamente equivalentes às atualizações globais vitorizadas em termos de como o termo de regularização é tratado. Isso ocorre porque os componentes de regularização das atualizações, que são  $\hat{y}_{uiq}$  e  $\hat{y}_{vjq}$ , são usados várias vezes em um ciclo de atualizações locais por meio de todas as entradas observadas; as atualizações são executadas em  $uiq$  para cada entrada observada na linha  $i$  e as atualizações são executadas em  $vjq$  para cada entrada observada na coluna  $j$ . Além disso, diferentes linhas e colunas podem ter números diferentes de entradas observadas, o que pode afetar ainda mais o nível relativo de regularização de vários usuários e

Uma atualização mais precisa deve ser  $\bar{u}_i = \bar{u}_i + \bar{\delta}(e_{ij} - \bar{y}_{ui}/n_{user})$  e  $\bar{v}_j = \bar{v}_j + \bar{\delta}(e_{ij} - \bar{y}_{vj}/n_{item})$ , representando o número de entradas observadas para o usuário/item. Alguns tipos de regularização para vários fatores de usuário/item são divididos igualmente entre as entradas observadas correspondentes para vários usuários/itens. Na prática, as regras de atualização heurística (mais simples) discutidas no capítulo são frequentemente usadas. Escolhemos usar essas regras (mais simples) ao longo deste capítulo para ser consistente com a literatura de pesquisa sobre sistemas de recomendação. Com o ajuste adequado dos parâmetros,  $\bar{\delta}$  se ajustará automaticamente a um valor menor no caso das regras de atualização mais simples.

Fatores de item. No método global vetorializado, a regularização é feita de forma mais suave e uniforme, pois cada entrada  $uiq$  e  $vjq$  é atualizada apenas uma vez. No entanto, como  $\hat{y}$  é escolhido de forma adaptativa durante o ajuste de parâmetros, o método de atualização local selecionará automaticamente valores menores de  $\hat{y}$  do que o método global. Do ponto de vista heurístico, os dois métodos fornecem resultados aproximadamente semelhantes, mas com diferentes compensações entre qualidade e eficiência.

Como antes,  $\hat{y} > 0$  representa o tamanho do passo e  $\hat{y} > 0$  é o parâmetro de regularização. Por exemplo, sabe-se que um pequeno valor constante de  $\hat{y}$ , como 0,005, funciona razoavelmente bem no caso do conjunto de dados do Prêmio Netflix. Alternativamente, pode-se usar o algoritmo de driver bold [58, 217] para selecionar  $\hat{y}$  de forma adaptativa em cada iteração, a fim de evitar ótimos locais e acelerar a convergência. Resta discutir como o parâmetro de regularização  $\hat{y}$  é selecionado.

O método mais simples é manter uma fração das entradas observadas na matriz de classificações e não usá-las para treinar o modelo. A precisão da predição do modelo é testada sobre esse subconjunto de entradas mantidas. Diferentes valores de  $\hat{y}$  são testados, e o valor de  $\hat{y}$  que fornece a maior precisão é usado. Se desejado, o modelo pode ser retrainado em todo o conjunto de entradas especificadas (sem retenções), uma vez que o valor de  $\hat{y}$  é selecionado. Este método de ajuste de parâmetros é chamado de método de retenção. Uma abordagem mais sofisticada é usar um método chamado validação cruzada. Este método é discutido no Capítulo 7 sobre avaliação de sistemas de recomendação. Para melhores resultados, diferentes parâmetros de regularização  $\hat{y}_1$  e  $\hat{y}_2$  podem ser usados para os fatores do usuário e fatores do item.

Frequentemente, pode ser dispendioso testar diferentes valores de  $\hat{y}$  no conjunto de retenção para determinar o valor ótimo. Isso restringe a capacidade de testar muitas opções de  $\hat{y}$ . Como resultado, os valores de  $\hat{y}$  frequentemente não são bem otimizados. Uma abordagem, proposta em [518], é tratar as entradas das matrizes  $U$  e  $V$  como parâmetros, e os parâmetros de regularização como hiperparâmetros, que são otimizados em conjunto com uma abordagem probabilística. Uma abordagem de amostragem de Gibbs é proposta em [518] para aprender conjuntamente os parâmetros e hiperparâmetros.

#### 3.6.4.3 Treinamento de Componentes Latentes Incrementais

Uma variante desses métodos de treinamento é treinar os componentes latentes incrementalmente. Em outras palavras, primeiro realizamos as atualizações  $uiq \leftarrow uiq + \hat{y}(eij \cdot vjq - \hat{y} \cdot uiq)$  e  $vjq \leftarrow vjq + \hat{y}(eij \cdot uiq - \hat{y} \cdot vjq)$  apenas para  $q = 1$ . A abordagem percorre repetidamente todas as entradas observadas em  $S$  enquanto realiza essas atualizações para  $q = 1$  até que a convergência seja alcançada.

Portanto, podemos aprender o primeiro par de colunas,  $U_1$  e  $V_1$ , de  $U$  e  $V$ , respectivamente.

Então, a matriz  $m \times n$  produto externo<sup>10</sup>  $U_1 V_1$  é subtraída de  $R$  (para entradas observadas).

Posteriormente, as atualizações são realizadas para  $q = 2$  com a matriz de classificações (residuais) para  $U_2 V_2$ . Aprenda o segundo par de colunas,  $U_2$  e  $V_2$ , de  $U$  e  $V$ , respectivamente. Em seguida, subtraia  $U_2 V_2$  de  $R$ . Esse processo é repetido a cada vez com a matriz residual até que  $q = k$ .

A abordagem resultante fornece a fatoração de matriz necessária porque a fatoração geral de classificação  $k$  pode ser expressa como a soma de  $k$  fatorações de classificação 1:

$$R \leftarrow UV^T = \sum_{q=1}^k U_q V_q^T \quad (3.17)$$

---

<sup>10</sup>O produto interno de dois vetores-coluna  $x$  e  $y$  é dado pelo escalar  $x^T y$ , enquanto o produto externo é dado pela matriz de posto 1  $x y^T$ . Além disso,  $x$  e  $y$  não precisam ter o mesmo tamanho para calcular um produto externo.

Algoritmo ComponentWise-SGD (Matriz de classificações: R, Taxa de aprendizagem:  $\hat{y}$ ) começar

```

Iniciar aleatoriamente as matrizes U e V;
S = {(i, j) : rij é observado}; para q = 1
a k comece enquanto
não
    (convergência) comece

        Embaralhe aleatoriamente as entradas observadas em S;
        para cada (i, j) ∈ S em ordem embaralhada comece eij ← rij
        ← uiqvjq;
        v̄t ←
        uiq ← u+ iq; v̄q ← v+ jq;
    fim

    Verifique a condição de convergência;
fim
{ Implementação elemento a elemento de R ← R ← Uq Vq para cada
(i, j) ∈ S do rij ← rij ← uiqvjq;
fim
fim

```

Figura 3.10: Implementação componente a componente da descida do gradiente estocástico

Uma descrição deste procedimento é ilustrada na Figura 3.10. As diferenças desta abordagem em relação à versão discutida anteriormente podem ser entendidas em termos das diferenças em suas estruturas de loop aninhadas. O treinamento de componentes incrementais percorre vários valores de  $q$  nos loops mais externos e percorre as entradas observadas repetidamente nos loops internos para atingir a convergência para cada valor de  $q$  (cf. Figura 3.10). O método anterior percorre as entradas observadas repetidamente para atingir a convergência nos loops externos e percorre vários valores de  $q$  no loop interno (cf. Figura 3.9). Além disso, o método incremental precisa ajustar a matriz de classificações entre duas execuções do loop externo. Essa abordagem leva a uma convergência mais rápida e estável em cada componente porque um número menor de variáveis é otimizado ao mesmo tempo.

Vale ressaltar que diferentes estratégias para descida de gradiente levarão a soluções com propriedades distintas. Essa forma específica de treinamento incremental fará com que os componentes latentes anteriores sejam os dominantes, o que proporciona um efeito semelhante ao do SVD. No entanto, as colunas resultantes em  $U$  (ou  $V$ ) podem não ser mutuamente ortogonais. Também é possível forçar a ortogonalidade mútua das colunas de  $U$  (e  $V$ ) usando a descida do gradiente projetada para  $q > 1$ . Especificamente, o vetor gradiente em relação às variáveis na coluna  $Uq$  (ou  $Vq$ ) é projetado em uma direção ortogonal às colunas ( $q \neq 1$ ) de  $U$  (ou  $V$ ) encontradas até o momento.

### 3.6.4.4 Mínimos Quadrados Alternados e Descida de Coordenadas

O método do gradiente estocástico é uma metodologia eficiente para otimização. Por outro lado, é bastante sensível, tanto à inicialização quanto à maneira como os tamanhos dos passos são escolhidos. Outros métodos de otimização incluem o uso de mínimos quadrados alternados (ALS) [268, 677], que é geralmente mais estável. A ideia básica desta abordagem é usar o seguindo a abordagem iterativa, começando com um conjunto inicial de matrizes U e V:

1. Mantendo U fixo, resolvemos cada uma das n linhas de V tratando o problema como um problema de regressão de mínimos quadrados. Somente as classificações observadas em S podem ser usadas para construindo o modelo de mínimos quadrados em cada caso. Seja  $v_j$  a j-ésima linha de V. Para determinar o vetor ótimo  $v_j$ , desejamos minimizar  $\sum_{(i,j) \in S} (v_{ij} - u_{ij})^2$ , que é um problema de regressão de mínimos quadrados em  $v_j$ . Os termos  $u_{ij}$  são tratados como valores constantes, enquanto  $v_j$  são tratados como variáveis de otimização. Portanto, os k componentes do fator latente em  $v_j$  para o j-ésimo item são determinados com regressão de mínimos quadrados. Um total de n desses problemas de mínimos quadrados precisam ser executados, e cada problema de mínimos quadrados tem k variáveis. Como o problema de mínimos quadrados para cada item é independente, esta etapa pode ser paralelizada facilmente.
  
2. Mantendo V fixo, resolva para cada uma das m linhas de U, tratando o problema como um problema de regressão de mínimos quadrados. Somente as classificações especificadas em S podem ser usadas para construir o modelo dos mínimos quadrados em cada caso. Seja  $u_i$  a i-ésima linha de U. Para determinar o vetor ótimo  $u_i$ , desejamos minimizar um problema de regressão de mínimos quadrados em  $u_i$ , que é  $\sum_{(i,j) \in S} (u_{ij} - v_{ij})^2$ , que é um problema de mínimos quadrados em  $u_i$ . Os termos  $v_{ij}$  são tratados como valores constantes, enquanto  $u_i$  são tratados como variáveis de otimização. Portanto, os k componentes do fator latente para o i-ésimo usuário são determinados com regressão. Um total de m desses problemas de mínimos quadrados precisam ser executados, e cada problema dos mínimos quadrados tem k variáveis. Como o problema dos mínimos quadrados para cada usuário é independente, esta etapa pode ser paralelizada facilmente.

Essas duas etapas são iteradas até a convergência. Quando a regularização é usada no objetivo função, equivale a usar a regularização de Tikhonov [22] na abordagem dos mínimos quadrados. O valor do parâmetro de regularização  $\lambda > 0$  pode ser fixado em todos os independentes problemas de mínimos quadrados, ou pode ser escolhido de forma diferente. Em ambos os casos, pode ser necessário determinar o valor ótimo de  $\lambda$  usando uma metodologia de validação cruzada ou de retenção. Uma breve discussão sobre regressão linear com regularização de Tikhonov é fornecida na seção 4.4.5 do Capítulo 4. Embora a discussão sobre regressão linear no Capítulo 4 seja fornecida no contexto de modelos baseados em conteúdo, a metodologia básica de regressão é invariante em todo o diferentes cenários em que é usado.

Curiosamente, uma versão ponderada do ALS é particularmente adequada para feedback implícito configurações nas quais se supõe que a matriz esteja totalmente especificada com muitos valores zero. Além disso, as entradas diferentes de zero costumam receber maior peso nessas configurações. Nesses casos, casos, a descida do gradiente estocástico torna-se muito cara. Quando a maioria das entradas são zeros, alguns truques podem ser usados para tornar o ALS ponderado uma opção eficiente. O leitor é referido em [260].

A desvantagem do ALS é que ele não é tão eficiente quanto a descida de gradiente estocástico em cenários de larga escala com classificações explícitas. Outros métodos, como a descida de coordenadas, podem efetivamente abordar o trade-off entre eficiência e estabilidade [650]. Na descida de coordenadas, a abordagem de fixar um subconjunto de variáveis (como no ALS) é levada ao extremo. Aqui, todas as entradas em U e V são fixadas, exceto por uma única entrada (ou coordenada) em uma das duas matrizes, que é otimizada usando a função objetivo da seção 3.6.4.2. A solução de otimização resultante pode ser demonstrada como tendo forma fechada porque é uma função objetivo quadrática em uma única variável. O valor correspondente de  $u_{iq}$  (ou  $v_{jq}$ ) pode ser determinado eficientemente de acordo com uma das duas atualizações a seguir:

$$\begin{aligned} u_{iq} \hat{y} &= \frac{(i,j)\hat{y}_S + (e_{ij} + u_{iq}v_{jq})v_{jq}}{\sum_j (i,j)\hat{y}_S v_{jq}} \\ v_{jq} \hat{y} &= \frac{i:(i,j)\hat{y}_S - (e_{ij} + u_{iq}v_{jq})u_{iq}}{\sum_i (i,j)\hat{y}_S u_{iq}} \end{aligned}$$

Aqui, S denota o conjunto de entradas observadas na matriz de classificações e  $e_{ij} = r_{ij} - \hat{r}_{ij}$  é o erro de previsão da entrada  $(i, j)$ . Percorre-se os parâmetros  $(m + n) \cdot k$  em U e V com essas atualizações até que a convergência seja alcançada. Também é possível combinar a descida de coordenadas com o treinamento incremental de componentes latentes, assim como a descida de gradiente estocástico é combinada com o treinamento de componentes incrementais (cf. seção 3.6.4.3).

### 3.6.4.5 Incorporando vieses de usuário e item

Uma variação do modelo irrestrito foi introduzida por Paterek [473] para incorporar variáveis que podem aprender vieses de usuários e itens. Suponha, para fins de discussão, que a matriz de classificações seja centrada na média, subtraindo a média global  $\bar{y}$  de toda a matriz de classificações de todas as entradas como uma etapa de pré-processamento. Após prever as entradas com o modelo de fator latente, o valor  $\hat{y}$  é adicionado novamente aos valores previstos como uma etapa de pós-processamento. Portanto, nesta seção, simplesmente assumiremos que a matriz de classificações R já foi centrada dessa forma e ignoraremos as etapas de pré-processamento e pós-processamento.

Associada a cada usuário  $i$ , temos uma variável  $o_i$ , que indica o viés geral dos usuários em classificar itens. Por exemplo, se o usuário  $i$  for uma pessoa generosa, que tende a classificar todos os itens como muito bons, então a variável  $o_i$  será uma quantidade positiva. Por outro lado, o valor de  $o_i$  será negativo para um rabugento que classifica a maioria dos itens negativamente. Da mesma forma, a variável  $p_j$  denota o viés nas classificações do item  $j$ . Itens muito apreciados (por exemplo, um sucesso de bilheteria) tenderão a ter valores maiores (positivos) de  $p_j$ , enquanto itens globalmente desaprovados terão valores negativos de  $p_j$ . É função do modelo fatorial aprender os valores de  $o_i$  e  $p_j$  de forma orientada por dados. A principal mudança no modelo fatorial latente original é que uma parte do  $(i, j)$ -ésimo

A classificação é explicada por  $o_i + p_j$  e o restante pela entrada  $(i, j)$  do produto UV das matrizes de fatores latentes. Portanto, o valor previsto da classificação da entrada  $(i, j)$  é dado por:

$$\hat{r}_{ij} = o_i + p_j + \sum_{s=1}^k \text{interfaces de usuário - vjs} \quad (3.18)$$

## 3.6. MODELOS DE FATORES LATENTES

Assim, o erro  $e_{ij}$  de uma entrada observada  $(i, j)$  é  $S$  é dado pelo seguinte:

$$e_{ij} = r_{ij} - \hat{r}_{ij} = r_{ij} - o_i - p_j \quad \text{interfaces de usuário - vjs} \quad (3.19)$$

$$\sum_{s=1}^k$$

Observe que os valores  $o_i$  e  $p_j$  também são variáveis que precisam ser aprendidas de forma orientada por dados, juntamente com as matrizes de fatores latentes  $U$  e  $V$ . Então, a função objetivo de minimização  $J$  pode ser formulada agregando os erros quadrados sobre as entradas observadas da matriz de classificações (ou seja, conjunto  $S$ ) da seguinte forma:

$$J = \frac{1}{2} \sum_{(i,j) \in S} \left[ r_{ij} - \sum_{s=1}^k \left( u_{es} + \frac{\sum_{j=1}^n v_{js}}{2} \right) \right]^2 + \sum_{i=1}^m \left( o_i^2 + \frac{\sum_{j=1}^n p_{ej}}{2} \right)^2$$

$$= \frac{1}{2} \sum_{(i,j) \in S} \left[ r_{ij} - \sum_{s=1}^k \left( u_{es} + \frac{\sum_{j=1}^n v_{js}}{2} \right) \right]^2 + \sum_{i=1}^m \left( o_i^2 + \frac{\sum_{j=1}^n p_{ej}}{2} \right)^2$$

Acontece que esse problema difere da fatoração de matrizes irrestrita apenas em um grau menor. Em vez de ter variáveis de viés separadas,  $o_i$  e  $p_j$ , para usuários e itens, podemos aumentar o tamanho das matrizes fatoriais para incorporar essas variáveis de viés. Precisamos adicionar duas colunas adicionais a cada matriz fatorial  $U$  e  $V$  para criar matrizes fatoriais maiores, de tamanho  $m \times (k+2)$  e  $n \times (k+2)$ , respectivamente. As duas últimas colunas de cada matriz fatorial são especiais, pois correspondem aos componentes de viés. Especificamente, temos:

$$\begin{aligned} u_{i,k+1} &= o_i \\ u_{i,k+2} &= 1 \\ v_{j,k+1} &= 1 \\ v_{j,k+2} &= p_j \end{aligned}$$

Observe que as condições  $u_{i,k+2} = 1$  e  $v_{j,k+1} = 1$  são restrições nas matrizes de fatores.

Em outras palavras, precisamos restringir a última coluna da matriz fator-usuário a todos os valores 1, e a penúltima coluna da matriz fator-item a todos os valores 1. Este cenário é ilustrado na Figura 3.11. Assim, o problema de otimização modificado com essas matrizes fatoriais ampliadas é o seguinte:

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j) \in S} \left[ r_{ij} - \sum_{s=1}^{k+2} \left( u_{es} + \frac{\sum_{j=1}^n v_{js}}{2} \right) \right]^2$$

sujeito a:

$$\begin{aligned} (k+2)^{\text{a}} \text{ coluna de } U &\text{ contém apenas 1s} \\ 1^{\text{a}} \text{ coluna de } V &\text{ contém apenas 1s} \end{aligned}$$

Vale ressaltar que as somas no objetivo são até  $(k+2)$  e não até  $k$ .

Observe que este problema é virtualmente idêntico ao caso irrestrito, exceto pelo menor

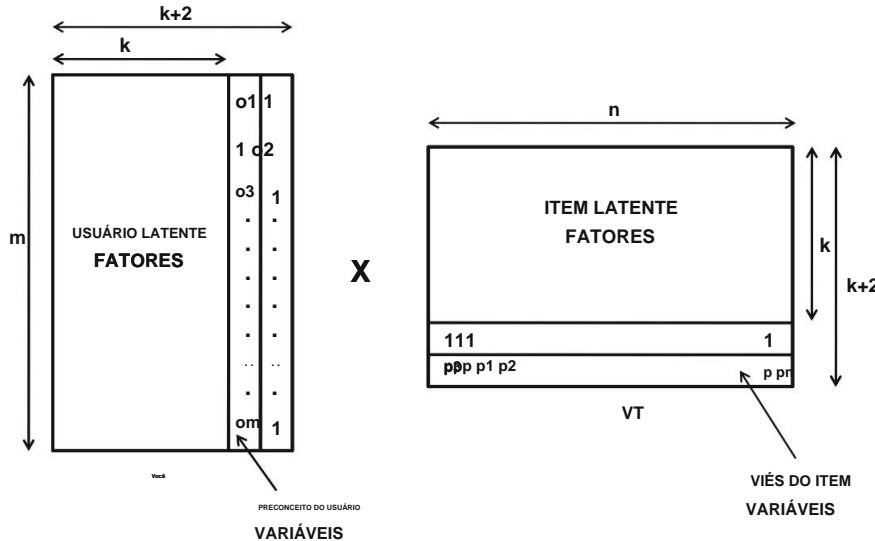


Figura 3.11: Incorporando vieses de usuário e item no modelo de fator latente

restrições aos fatores. A outra mudança é o aumento no tamanho das matrizes de fatores para incorporar as variáveis de viés do usuário e do item. Devido à pequena alteração na formulação do problema, basta fazer as alterações correspondentes no método de gradiente descendente.

Para inicialização, a coluna  $(k + 1)$  de  $V$  e a coluna  $(k + 2)$  de  $U$  são definidas como 1s.

Exatamente as mesmas regras de atualização (locais) são usadas no caso irrestrito, exceto que as duas entradas perturbadas na coluna  $(k + 1)$  de  $V$  e na coluna  $(k + 2)$  de  $U$  são redefinidas para seus valores fixos após cada atualização (ou simplesmente não são atualizadas). As seguintes atualizações podem ser executadas percorrendo cada entrada especificada  $(i, j) \in S$ :

$$uiq \leftarrow uiq + \hat{y}(eij \cdot vjq - uiq) \hat{y} \{1 \dots k + 2\} vjq \leftarrow vjq + \hat{y}(eij \cdot$$

$$uiq \hat{y} \hat{y} \cdot vjq) \hat{y} \{1 \dots k + 2\}$$

Redefinir entradas perturbadas na coluna  $(k + 2)$  de  $U$  e na coluna  $(k + 1)$  de  $V$  para 1s

Este conjunto de atualizações é realizado simultaneamente como um grupo. Também é possível utilizar o método dos mínimos quadrados alternados com pequenas variações (ver Exercício 11). A discussão mencionada utiliza os mesmos parâmetros de regularização e taxas de aprendizado para cada tipo de variável. Às vezes, recomenda-se o uso de diferentes parâmetros de regularização e taxas de aprendizado para os vieses do usuário, vieses dos itens e variáveis fatoriais [586]. Isso pode ser alcançado com pequenas modificações nas atualizações mencionadas.

Uma questão natural que surge é por que essa formulação deve ter um desempenho melhor do que a fatoração de matrizes sem restrições. A adição de restrições nas duas últimas colunas das matrizes de fatores deve apenas reduzir a qualidade global da solução, pois agora se está otimizando sobre um espaço menor de soluções. No entanto, em muitos casos, a adição de tais restrições enviesa a solução, ao mesmo tempo que reduz o sobreajuste. Em outras palavras, a adição de tais restrições intuitivas pode frequentemente melhorar a generalização do algoritmo de aprendizagem para entradas não vistas, mesmo que o erro sobre as entradas especificadas possa ser maior. Isso é particularmente útil quando o número de avaliações observadas para um usuário ou para um item é pequeno [473]. Variáveis de viés adicionam um componente às avaliações que são globais para os usuários ou para os itens. Tais globais

Propriedades são úteis quando há dados limitados disponíveis. Como exemplo específico, considere o caso em que um usuário forneceu classificações para apenas um pequeno número (1 ou 2) itens. Nesses casos, muitos algoritmos de recomendação, como métodos baseados em vizinhança, não fornecerão previsões confiáveis para o usuário. Por outro lado, as previsões (não personalizadas) das variáveis de viés do item serão capazes de fornecer previsões razoáveis. Afinal, se um determinado filme for um sucesso de bilheteria globalmente, o usuário relevante também terá maior probabilidade de apreciá-lo.

As variáveis de viés também refletirão esse fato e o incorporarão ao algoritmo de aprendizagem.

De fato, foi demonstrado [73, 310, 312] que o uso apenas de variáveis de viés (ou seja,  $k = 0$ ) pode frequentemente fornecer previsões de classificação razoavelmente boas. Este ponto foi enfatizado como uma das lições práticas aprendidas no concurso Netflix Prize [73]:

Das inúmeras novas contribuições algorítmicas, gostaria de destacar uma: os modestos preditores de linha de base (ou vieses), que capturam os efeitos principais nos dados. Embora a literatura se concentre principalmente nos aspectos algorítmicos mais sofisticados, aprendemos que um tratamento preciso dos efeitos principais é provavelmente tão significativo quanto a descoberta de avanços na modelagem.

Isso significa que uma parte significativa das classificações pode ser explicada pela generosidade do usuário e pela popularidade dos itens, em vez de quaisquer preferências personalizadas específicas dos usuários por itens. Esse modelo não personalizado é discutido na seção 3.7.1, o que equivale a definir  $k = 0$  no modelo mencionado anteriormente. Como resultado, apenas os vieses dos usuários e dos itens são aprendidos, e uma classificação de base  $B_{ij}$  é prevista para o usuário  $i$  e o item  $j$  pela soma de seus vieses. Pode-se usar essa classificação de base para aprimorar qualquer modelo de filtragem colaborativa pronto para uso. Para isso, pode-se simplesmente subtrair cada  $B_{ij}$  da entrada  $(i, j)$ -ésima (observada) da matriz de classificações antes de aplicar a filtragem colaborativa. Esses valores são adicionados novamente em uma fase de pós-processamento aos valores previstos. Essa abordagem é especialmente útil para modelos nos quais não é possível parametrizar facilmente as variáveis de viés. Por exemplo, modelos de vizinhança (tradicional) alcançam essas metas de correção de viés com centralização média por linha, embora o uso de  $B_{ij}$  para corrigir as entradas da matriz seja uma abordagem mais sofisticada porque ajusta os vieses do usuário e do item.

#### 3.6.4.6 Incorporando Feedback Implícito

Geralmente, cenários de feedback implícito correspondem ao uso de matrizes de classificação unárias nas quais os usuários expressam seus interesses comprando itens. No entanto, mesmo nos casos em que os usuários classificam os itens explicitamente, a identidade dos itens que eles classificam pode ser vista como um feedback implícito. Em outras palavras, um valor preditivo significativo é capturado pela identidade dos itens que os usuários avaliam, independentemente dos valores reais das avaliações. Um artigo recente [184] descreve esse fenômeno com elegância no contexto do domínio musical:

Intuitivamente, um processo simples poderia explicar os resultados [mostrando o valor preditivo do feedback implícito]: os usuários escolhem avaliar as músicas que ouvem e ouvem músicas que esperam gostar, evitando gêneros dos quais não gostam. Portanto, a maioria das músicas que receberiam uma avaliação ruim não são avaliadas voluntariamente pelos usuários.  
Como as pessoas raramente ouvem músicas aleatórias ou raramente assistem a filmes aleatórios, devemos esperar observar em muitas áreas uma diferença entre a distribuição de classificações para itens aleatórios e a distribuição correspondente para os itens selecionados pelos usuários.”

Diversas estruturas, como modelos de fatores assimétricos e SVD++, foram propostas para incorporar feedback implícito. Esses algoritmos utilizam duas matrizes fatoriais de itens diferentes:  $V$

e  $Y$ , correspondendo ao feedback explícito e implícito, respectivamente. Os fatores latentes do usuário são total ou parcialmente derivados usando uma combinação linear dessas linhas da matriz de fatores latentes do item (implícito)  $Y$  que correspondem aos itens avaliados do usuário. A ideia é que os fatores do usuário correspondem às preferências do usuário e, portanto, as preferências do usuário devem ser influenciadas pelos itens que eles escolheram avaliar. Na versão mais simples dos modelos de fatores assimétricos, uma combinação linear dos vetores de fatores (implícitos) dos itens avaliados é usada para criar os fatores do usuário. Isso resulta em uma abordagem assimétrica na qual não temos mais variáveis independentes para os fatores do usuário. Em vez disso, temos dois conjuntos de fatores de itens independentes (ou seja, explícito e implícito), e os fatores do usuário são derivados como uma combinação linear dos fatores de itens implícitos. Muitas variantes [311] dessa metodologia são discutidas na literatura, embora a ideia original seja creditada a Paterek [473]. O modelo SVD++ combina ainda mais essa abordagem assimétrica com fatores de usuário (explícitos) e uma estrutura de fatoração tradicional. A abordagem assimétrica pode, portanto, ser vista como um precursor simplificado do SVD++. Para maior clareza na exposição, discutiremos primeiro brevemente o modelo assimétrico.

**Modelos de Fatores Assimétricos:** Para capturar as informações de feedback implícitas, primeiro derivamos uma matriz de feedback implícita da matriz de classificações explícita. Para uma matriz de classificações  $m \times n R$ , a matriz de feedback implícita  $m \times n F = [f_{ij}]$  é definida definindo-a como 1, se o valor  $r_{ij}$  for observado, e 0, se estiver ausente. A matriz de feedback  $F$  é subsequentemente normalizada para que a norma L2 de cada linha seja 1. Portanto, se  $r_{ij}$  for o conjunto de índices dos itens avaliados pelo usuário  $i$ , então cada entrada diferente de zero na  $i$ -ésima linha é  $1 / \|r_i\|$ . Um exemplo de uma matriz de classificações  $R$  juntamente com sua matriz de feedback implícita correspondente  $F$  é ilustrado abaixo:

$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & ? & ? & ? & ? & ? \\ 1 & 2 & 2 & 2 & 2 & 2 \\ 0 & ? & ? & ? & ? & ? \end{matrix}$	$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 1/1 & 1/1 & 1/1 & 1/1 & 1/1 & 1/1 \\ 1/1 & 1/1 & 1/1 & 1/1 & 1/1 & 1/1 \\ 1/1 & 1/1 & 1/1 & 1/1 & 1/1 & 1/1 \\ 1/1 & 1/1 & 1/1 & 1/1 & 1/1 & 1/1 \end{matrix}$
$\underline{\underline{R}}$	$\underline{\underline{F}}$	$\underline{\underline{F}}$

Uma matriz  $n \times k Y = [y_{ij}]$  é usada como a matriz implícita item-fator e a matriz  $F$  fornece os coeficientes de combinação linear para criar uma matriz usuário-fator a partir dela. As variáveis em  $Y$  codificam a propensão de cada combinação fator-item para contribuir para o feedback implícito. Por exemplo, se  $|y_{ij}|$  for grande, significa que simplesmente o ato de classificar o item  $i$  contém informações significativas sobre a afinidade dessa ação para o  $j$ -ésimo componente latente, não importa qual seja o valor real da classificação. No modelo assimétrico simplificado, os fatores do usuário são codificados como combinações lineares dos fatores implícitos dos itens avaliados; a ideia básica é que combinações lineares de ações do usuário são usadas para definir suas preferências (fatores). Especificamente, o produto da matriz  $FY$  é uma matriz usuário-fator  $m \times k$ , e cada linha (específica do usuário) nela é uma combinação linear (específica do usuário) de fatores implícitos dos itens, dependendo dos itens avaliados pelo usuário. A matriz  $FY$  é usada em vez da matriz fatorial do usuário  $U$ , e a matriz de classificações é fatorada como  $R \approx [FY]V$ , onde  $V$  é a matriz fatorial do item explícita  $n \times k$ . Se desejado, variáveis de viés podem ser incorporadas ao modelo centralizando a média da matriz de classificações e acrescentando duas colunas adicionais a cada um dos parâmetros  $Y$  e  $V$ . Essa abordagem simples frequentemente fornece resultados excelentes<sup>11</sup> porque reduz a redundância nos fatores do usuário, derivando-os como combinações lineares de fatores do item. O básico , conforme discutido na seção 3.6.4.5 (ver Exercício 13).

<sup>11</sup>Em muitos casos, essa abordagem pode superar o SVD++, especialmente quando o número de classificações observadas é pequeno.

A ideia aqui é que dois usuários terão fatores de usuário semelhantes se eles tiverem avaliado itens semelhantes, independentemente dos valores das classificações. Observe que a matriz  $n \times k$  Y contém menos parâmetros do que uma matriz de fatores do usuário  $m \times k$  U porque  $n < m$ . Outra vantagem disso é que é possível incorporar outros tipos de feedback implícito independente (como comportamento de compra ou navegação) incorporando-o na matriz de feedback implícita F. Nesses casos, a abordagem geralmente pode ter um desempenho melhor do que a maioria das outras formas de fatoração de matrizes (com classificações explícitas) devido à sua capacidade de usar classificações explícitas e implícitas. No entanto, mesmo nos casos em que não há feedback implícito independente disponível, este modelo parece ter um desempenho melhor do que variações diretas de fatoração de matriz para muitas matrizes esparsas com um grande número de usuários (em comparação com o número de itens). Uma vantagem adicional deste modelo é que não são necessárias parametrizações do usuário; portanto, a o modelo pode funcionar bem para usuários fora da amostra, embora não possa ser usado para usuários fora da amostra itens. Em outras palavras, o modelo é pelo menos parcialmente indutivo, ao contrário da maioria dos métodos de fatoração de matrizes. Omitimos a discussão das etapas de descida do gradiente deste modelo, porque A generalização deste modelo é discutida na próxima seção. As etapas correspondentes são, no entanto, enumeradas na declaração do problema do Exercício 13.

A parametrização baseada em itens de modelos de fatores assimétricos também lhe confere o mérito de explicabilidade. Observe que se pode reescrever a fatoração  $[FY]V^T$  como  $F[YV^T]$ . O matriz  $V^T$  pode ser visto como uma matriz de previsão item a item  $n \times n$  na qual  $[YV^T]_{ij}$  nos diz quanto o ato de classificar o item  $i$  contribui para a classificação prevista do item  $j$ . A matriz F fornece os coeficientes  $m \times n$  usuário-item correspondentes e, portanto, multiplicando F com  $[YV^T]$  fornece previsões do usuário para o item. Portanto, agora é possível explicar quais itens previamente consumido/avaliado pelo usuário contribuiu mais para a previsão em F[O tipo de explicabilidade]. Esse  $YV^T$  é inerente aos modelos centrados em itens.

**SVD++:** A derivação de fatores do usuário puramente com base nas identidades dos itens classificados parece um uso bastante extremo de feedback implícito em modelos de fatores assimétricos. Isto é porque tal abordagem não discrimina de forma alguma entre pares de usuários que avaliaram exatamente o mesmo conjunto de itens, mas com valores observados muito diferentes das classificações. Dois desses usuários receberão exatamente a mesma previsão de classificação para um item que não foi classificado por ambos.

Em SVD++, uma abordagem mais sutil é usada. A matriz de fator de usuário implícita FY é usado apenas para ajustar a matriz explícita de fatores do usuário U, em vez de criá-la. Portanto, FY precisa ser adicionado a U antes de multiplicar com V a matriz  $R^T$ . Então, as classificações  $m \times n$  reconstruídas é dada por  $(U + FY)V$  a classificação é dada  $R^T$ , e o componente de feedback implícito do previsto por  $(FY)V$  é que o número de  $R^T$ . O preço pela flexibilidade de modelagem adicional no SVD++ parâmetros é aumentado, o que pode causar overfitting em muitos esparsos matrizes de classificação. A matriz de feedback implícita pode ser derivada da matriz de classificação (como em modelos de fatores assimétricos), embora outras formas de feedback implícito (por exemplo, compra ou comportamento de navegação) também podem ser incluídos.

Os vieses do usuário e do item são incluídos neste modelo de maneira semelhante à seção 3.6.4.5. Podemos assumir, sem perda de generalidade, que a matriz de classificação é centrada na média em torno de a média global  $\bar{y}$  de todas as entradas. Portanto, trabalharemos com  $m \times (k+2)$  e  $n \times (k+2)$  matrizes fatoriais U e V, respectivamente, nas quais as duas últimas colunas contêm 1s ou variáveis de viés de acordo com a seção 3.6.4.5. Também assumimos13 que Y é uma matriz  $n \times (k+2)$ ,

12Para matrizes que não são centradas na média, a média global pode ser subtraída durante o pré-processamento e depois adicionado novamente no momento da previsão.

13Usamos uma notação ligeiramente diferente do artigo original [309], embora a abordagem descrita aqui é equivalente. Esta apresentação simplifica a notação introduzindo menos variáveis e viés de visualização

e as duas últimas colunas de  $Y$  contêm 0s. Isso ocorre porque o componente de viés já é abordado pelas duas últimas colunas de  $U$ , mas precisamos das duas últimas colunas fictícias em  $Y$  para garantir que possamos adicionar  $U$  e  $FY$  como matrizes das mesmas dimensões. Portanto, a classificação prevista  $\hat{r}_{ij}$  pode ser expressa em termos dessas variáveis da seguinte forma:

$$\begin{aligned} \hat{r}_{ij} &= \sum_{s=1}^{k+2} (\text{uis} + [FY]_s) \cdot v_{js} \\ &= \frac{\sum_{s=1}^{k+2} \frac{y_{hs}}{\|i\|} \cdot v_{js}}{\sum_{s=1}^{k+2} h_{li}} \end{aligned} \quad (3.20)$$

O primeiro termo  $\sum_{s=1}^{k+2} u_{is} v_{js}$  no lado direito da equação acima mencionada é o  $v_{js}$  é o  $(i, j)$ -ésimo termo de  $(i, j)$ -ésimo termo do  $T$ , e o segundo termo  $\sum_{s=1}^{k+2} h_{li} \hat{y}_{hs} \|i\|$  é o modelo  $T$ . Observe que a entrada  $(i, s)$ -ésima de  $[FY]$  é dada por  $h_{li} \hat{y}_{hs} \|i\|$ . Pode-se ver isto

UV  $[FY]V$  como uma combinação do modelo de fatoração de matriz irrestrita (com vieses) e do modelo de fatoração assimétrica discutido na seção anterior. Portanto, ele combina os pontos fortes de ambos os modelos.

O problema de otimização correspondente, que minimiza o erro quadrático agregado  $= (r_{ij} - \hat{r}_{ij})^2$  sobre todas as entradas observadas (denotadas pelo conjunto  $S$ ) na matriz de classificações, pode ser declarado da seguinte forma:

$$\text{Mín. } \sum_{(i,j) \in S} \left( r_{ij} - \sum_{s=1}^{k+2} \frac{y_{hs}}{\|i\|} \cdot v_{js} \right)^2 + \sum_{s=1}^{k+2} \sum_{j=1}^{n_s} \left( \frac{\hat{y}_{js}}{2} - \sum_{i=1}^{m_s} \frac{y_{mi}}{\|i\|} \cdot v_{js} \right)^2$$

sujeito a:

$(k+2)^2$  coluna de  $U$  contém apenas 1s ( $k+1)^2$

coluna de  $V$  contém apenas 1s

As duas últimas colunas de  $Y$  contêm apenas 0s

Observe que esta formulação de otimização difere daquela da seção anterior por possuir um termo de feedback implícito juntamente com seu regularizador. Pode-se usar a derivada parcial desta função objetivo para derivar as regras de atualização para as matrizes  $U$  e  $V$ , bem como para as variáveis em  $Y$ . As regras de atualização são então expressas em termos dos valores de erro  $e_{ij} = r_{ij} - \hat{r}_{ij}$  das entradas observadas. As seguintes atualizações<sup>14</sup> podem ser usadas para cada

---

variáveis como restrições no processo de fatoração.

14A literatura frequentemente descreve essas atualizações de forma vetorializada. Essas atualizações podem ser aplicadas às linhas de  $U$ ,  $V$  e  $Y$  da seguinte forma:

$$\begin{aligned} \bar{u}_i &= \bar{y}_i + \frac{\sum_j e_{ij} v_{j|i}}{\|i\|} \\ \bar{v}_j &= \bar{y}_j + \frac{\sum_i e_{ij} \bar{u}_{i|j}}{\|j\|} \\ \bar{y}_i &= \bar{y}_i + \frac{\sum_{ij} e_{ij} \bar{u}_{i|j} \bar{v}_{j|i}}{\|i\|} \end{aligned}$$

Redefinir entradas perturbadas em colunas fixas de  $U$ ,  $V$ , e  $Y$

entrada observada ( $i, j$ ) é  $S$  na matriz de classificações:

$$\begin{aligned} uiq &= uiq + \hat{y}(eij \cdot vjq - \hat{y} \cdot uiq) \quad \hat{y} \in \{1 \dots k+2\} \\ vjq &= vjq + \hat{y} eij \cdot uiq + \frac{\hat{y} hq}{\|h\|_i} \quad \hat{y} \in \{1 \dots k+2\} \\ yhq &= yhq + \hat{y} \frac{e ij \cdot vjq}{\|v\|_j} \quad \hat{y} \in \{1 \dots k+2\}, \hat{y} \in \{1 \dots k+2\} \\ \end{aligned}$$

Redefinir entradas perturbadas em colunas fixas de  $U, V$ , e  $Y$

As atualizações são executadas repetindo o loop sobre todas as classificações observadas em  $S$ . As entradas perturbadas nas colunas fixas de  $U, V$  e  $Y$  são redefinidas por essas regras para 1s e 0s. Uma alternativa mais eficiente (e prática) seria simplesmente não atualizar as entradas fixas, mantendo-as sob controle durante a atualização. Além disso, essas colunas são sempre inicializadas com valores fixos que respeitam as restrições do modelo de otimização. A estrutura de loop aninhado da descida do gradiente estocástico é semelhante em toda a família de métodos de fatoração de matrizes. Portanto, a estrutura básica descrita na Figura 3.9 pode ser usada, embora as atualizações sejam baseadas na discussão acima mencionada. Melhores resultados podem ser obtidos usando diferentes parâmetros de regularização para diferentes matrizes de fatores. Uma variação rápida da descida do gradiente estocástico é descrita em [151]. Também é possível desenvolver uma abordagem de mínimos quadrados alternados para resolver o problema acima mencionado (ver Exercício 12).

Embora este modelo seja chamado de SVD++ [309], o nome é um pouco enganoso, pois os vetores base das matrizes fatoradas não são ortogonais. De fato, o termo "SVD" é frequentemente aplicado de forma vaga na literatura sobre modelos de fatores latentes. Na próxima seção, discutiremos o uso da decomposição em valores singulares com vetores ortogonais.

### 3.6.5 Decomposição de Valor Singular

A decomposição em valor singular (DVS) é uma forma de fatoração matricial na qual as colunas de  $U$  e  $V$  são restrinvidas a serem mutuamente ortogonais. A ortogonalidade mútua tem a vantagem de que os conceitos podem ser completamente independentes uns dos outros e podem ser interpretados geometricamente em diagramas de dispersão. No entanto, a interpretação semântica dessa decomposição é geralmente mais difícil, pois esses vetores latentes contêm quantidades positivas e negativas e são restrinvidos por sua ortogonalidade a outros conceitos.

Para uma matriz totalmente especificada, é relativamente fácil realizar a SVD com o uso de métodos de autodecomposição. Primeiro, recapitularemos brevemente a discussão sobre decomposição em valores singulares na seção 2.5.1.2 do Capítulo 2.

Considere o caso em que a matriz de classificações é totalmente especificada. É possível fatorar aproximadamente a matriz de classificações  $R$  usando a SVD truncada de classificação  $k \min\{m, n\}$ . A SVD truncada é calculada da seguinte forma:

$$R \approx Qk \tilde{y} k P^T \quad (3.22)$$

Aqui,  $Qk$ ,  $\tilde{y} k$  e  $Pk$  são matrizes de tamanho  $m \times k$ ,  $k \times k$  e  $n \times k$ , respectivamente. As matrizes  $Qk$  e  $Pk$  contêm, respectivamente, os  $k$  maiores autovetores de  $R^T R$  e  $R R^T$ , enquanto a matriz (diagonal)  $\tilde{y} k$  contém as raízes quadradas (não negativas) dos  $k$  maiores autovalores de qualquer matriz ao longo de sua diagonal. É digno de nota que os autovalores diferentes de zero de  $R^T R$  e  $R R^T$  são os mesmos, embora tenham um número diferente de autovalores nulos quando  $m = n$ . A matriz  $Pk$  contém os autovetores superiores de  $R^T R$ , que é a matriz reduzida

representação básica necessária para a redução da dimensionalidade do espaço de linhas. Esses autovetores contêm informações sobre as direções das correlações item-item entre as classificações e portanto, eles fornecem a capacidade de representar cada usuário em um número reduzido de dimensões em um sistema de eixo rotacionado. Por exemplo, na Figura 3.6, o autovetor superior corresponde ao vetor latente que representa as direções dominantes das correlações item-item. Além disso, a matriz  $Q\tilde{y}k$  contém a representação  $m \times k$  transformada e reduzida do original matriz de classificações na base correspondente a  $P_k$ . Portanto, na Figura 3.6, a matriz  $Q\tilde{y}k$  seria um vetor de coluna unidimensional contendo as coordenadas das classificações ao longo da vetor latente dominante.

É fácil ver pela Equação 3.22 que SVD é inherentemente definida como uma fatoração de matrizes. É claro que a fatoração aqui é em três matrizes em vez de duas. No entanto, a matriz diagonal  $\tilde{y}k$  pode ser absorvida nos fatores do usuário  $Q_k$  ou nos fatores do item  $P_k$ . Por convenção, os fatores do usuário e os fatores do item são definidos da seguinte forma:

$$U = Q\tilde{y}k$$

$$V = P_k$$

Como antes, a fatoração da matriz de classificação  $R$  é definida como  $R = UV$ . As matrizes  $U$  e  $V$  são compostas por fatores de usuário e item têm colunas ortogonais, é fácil converter o resultado da fatoração em uma forma compatível com SVD (ver Exercício 9). Portanto, o objetivo do processo de fatoração é descobrir matrizes  $U$  e  $V$  com colunas ortogonais. Portanto, SVD pode ser formulado como o seguinte problema de otimização sobre as matrizes  $U$  e  $V$ :

$$\text{Minimizar } J = \frac{1}{2} \|R - UV\|_F^2$$

sujeito a:

As colunas de  $U$  são mutuamente ortogonais

As colunas de  $V$  são mutuamente ortogonais

É fácil ver que a única diferença em relação ao caso da fatoração irrestrita é a presença de restrições de ortogonalidade. Em outras palavras, a mesma função objetivo está sendo otimizado em um espaço menor de soluções em comparação à fatoração de matriz irrestrita. Embora fosse de se esperar que a presença de restrições aumentasse o erro  $J$  de uma aproximação, verifica-se que o valor ótimo de  $J$  é idêntico no caso de SVD e fatoração de matriz irrestrita, se a matriz  $R$  for totalmente especificada e regularização não é usado. Portanto, para matrizes totalmente especificadas, a solução ótima para SVD é uma de os ótimos alternativos da fatoração de matrizes irrestritas. Isso não é necessariamente verdade em os casos em que  $R$  não é totalmente especificado e a função objetivo  $J$  é calculada apenas  $\frac{1}{2} \|R - UV\|_F^2$  sobre as entradas observadas. Nesses casos, a fatoração de matrizes irrestrita normalmente fornecerá menor erro nas entradas observadas. No entanto, o desempenho relativo nas entradas não observadas podem ser imprevisíveis devido aos vários níveis de generalização de diferentes modelos.

### 3.6.5.1 Uma abordagem iterativa simples para SVD

Nesta seção, discutimos como resolver o problema de otimização quando a matriz  $R$  é incompletamente especificada. O primeiro passo é centralizar cada linha de  $R$  subtraindo a classificação média  $\bar{y}_i$  do usuário  $i$  a partir dele. Essas médias por linha são armazenadas porque elas irão eventualmente ser necessárias para reconstruir as classificações brutas das entradas ausentes. Deixe o centralizado

matriz pode ser denotada por  $R_c$ . Então, as entradas ausentes de  $R_c$  são definidas como 0. Esta abordagem define efetivamente as entradas ausentes para a classificação média do usuário correspondente, porque as entradas ausentes da matriz centralizada são definidas como 0. SVD é então aplicado a  $R_c$  para obter a decomposição  $R_c = Q \tilde{y} k P^T$ . Os fatores do usuário e os fatores do item resultantes são dados por  $U = Q \tilde{y} k$  e  $V = P^T k$ . Seja a  $i$ -ésima linha de  $U$  o vetor  $k$ -dimensional denotado por  $u_i$  e a  $j$ -ésima linha de  $V$  seja o vetor  $k$ -dimensional denotado por  $v_j$ . Então, a classificação  $\hat{r}_{ij}$  de um usuário  $i$  para o item  $j$  é estimado como o seguinte produto escalar ajustado de  $u_i$  e  $v_j$ :

$$\hat{r}_{ij} = u_i \cdot v_j + \bar{y}_i \quad (3.23)$$

Observe que a média  $\bar{y}_i$  do usuário  $i$  precisa ser adicionada à classificação estimada para levar em conta a centralização média aplicada a  $R$  na primeira etapa.

O principal problema com esta abordagem é que a substituição de entradas ausentes por suas médias por linha podem levar a um viés considerável. Um exemplo específico de como as médias por coluna  $A$  substituição leva ao viés é apresentada na seção 2.5.1 do Capítulo 2. O argumento para a substituição por linha é exatamente semelhante. Existem várias maneiras de reduzir esse viés. Uma delas é usar a estimativa de máxima verossimilhança [24, 472], que é discutida na seção 2.5.1.1 do Capítulo 2. Outra abordagem é usar um método que reduz o viés iterativamente, melhorando a estimativa das entradas ausentes. A abordagem utiliza as seguintes etapas:

1. Inicialização: Inicialize as entradas faltantes na  $i$ -ésima linha de  $R$  para serem a média  $\bar{y}_i$  de essa linha para criar  $R_f$ .
2. Etapa iterativa 1: Execute o SVD de classificação  $k$  de  $R_f$  na forma  $Q \tilde{y} k P^T$ .
3. Etapa iterativa 2: reajuste apenas as entradas (originalmente) ausentes de  $R_f$  aos valores correspondentes em  $Q \tilde{y} k P^T$ . Vá para a etapa iterativa 1.

As etapas iterativas 1 e 2 são executadas até a convergência. Neste método, embora a etapa de inicialização cause viés nas iterações iniciais do SVD, as iterações posteriores tendem a fornecer estimativas robustas. Isso ocorre porque a matriz  $Q \tilde{y} k P^T$  diferirá de  $R$  em maior grau nas entradas tendenciosas. A matriz de classificação final é então dada por  $Q \tilde{y} k P^T$  na convergência.

A abordagem pode ficar presa em um ótimo local quando o número de entradas ausentes é grande. Em particular, o ótimo local na convergência pode ser sensível à escolha de inicialização. Também é possível usar o preditor de linha de base discutido na seção 3.7.1 para realizar uma inicialização mais robusta. A ideia é calcular um valor previsto inicial  $B_{ij}$  para usuário  $i$  e item  $j$  com o uso de vieses aprendidos de usuário e item. Esta abordagem é equivalente para aplicar o método na seção 3.6.4.5 em  $k = 0$ , em seguida, adicionar o viés do usuário  $i$  a  $B_{ij}$  para derivar  $B_{ij}$ . O valor de  $B_{ij}$  é subtraído de cada entrada observada  $(i, j)$  na matriz de classificação, e as entradas ausentes são definidas como 0 na inicialização. O mencionado abordagem iterativa é aplicada a esta matriz ajustada. O valor de  $B_{ij}$  é adicionado novamente a entrada  $(i, j)$  no momento da previsão. Tal abordagem tende a ser mais robusta devido à melhor inicialização.

A regularização pode ser usada em conjunto com o método iterativo mencionado anteriormente. A ideia é realizar SVD regularizado de  $R_f$  em cada iteração em vez de usar apenas SVD vanilla. Como a matriz  $R_f$  é totalmente especificada em cada iteração, é relativamente fácil para aplicar métodos SVD regularizados a essas matrizes intermediárias. Singular regularizado métodos de decomposição de valor para matrizes completas são discutidos em [541]. O ótimo os valores dos parâmetros de regularização  $\gamma_1$  e  $\gamma_2$  são escolhidos de forma adaptativa usando o métodos de validação cruzada ou hold-out.

### 3.6.5.2 Uma abordagem baseada em otimização

A abordagem iterativa é bastante dispendiosa porque funciona com matrizes totalmente especificadas. É simples de implementar para matrizes menores, mas não escala bem em ambientes de larga escala. Uma abordagem mais eficiente é adicionar restrições de ortogonalidade ao modelo de otimização das seções anteriores. Uma variedade de métodos de gradiente descendente pode ser usada para resolver o modelo.

Seja  $S$  o conjunto de entradas especificadas na matriz de classificações. O problema de otimização (com regularização) é enunciado da seguinte forma:

$$\text{Minimizar } J = \frac{1}{2} \sum_{(i,j) \in S} r_{ij} \hat{y}_{ij}^2 + \frac{\|U\|_F^2 + \|V\|_F^2}{2}$$

sujeito a:

As colunas de  $U$  são mutuamente ortogonais

As colunas de  $V$  são mutuamente ortogonais

A principal diferença deste modelo em relação à fatoração de matrizes irrestrita é a adição de restrições de ortogonalidade, o que torna o problema mais difícil. Por exemplo, se alguém tentar usar diretamente as equações de atualização da seção anterior sobre fatoração de matrizes irrestrita, as restrições de ortogonalidade serão violadas. No entanto, existe uma variedade de métodos de atualização modificados para lidar com este caso. Por exemplo, pode-se usar um método de descida de gradiente projetada [76], em que todos os componentes de uma coluna específica de  $U$  ou  $V$  são atualizados ao mesmo tempo. Na descida de gradiente projetada, a direção da descida para a coluna  $p$ -ésima de  $U$  (ou  $V$ ), conforme indicado pelas equações da seção anterior, é projetada em uma direção ortogonal às primeiras ( $p-1$ ) colunas de  $U$  (ou  $V$ ). Por exemplo, a implementação da seção 3.6.4.3 pode ser adaptada para aprender fatores ortogonais projetando cada fator em uma direção ortogonal aos aprendidos até então em cada etapa. É possível incorporar facilmente vieses de usuários e itens calculando as previsões de base  $B_{ij}$  (discutidas na seção anterior) e subtraindo-as das entradas observadas na matriz de classificações antes da modelagem. Posteriormente, os valores de base podem ser adicionados novamente aos valores previstos como uma etapa de pós-processamento.

### 3.6.5.3 Recomendações fora da amostra

Muitos métodos de complementação de matrizes, como a fatoração de matrizes, são inherentemente transdutivos, nos quais as previsões podem ser feitas apenas para usuários e itens já incluídos na matriz de classificações no momento do treinamento. Muitas vezes, não é fácil fazer previsões para novos usuários e itens a partir dos fatores  $U$  e  $V$  se eles não estiverem incluídos na matriz de classificações original  $R$  no momento da fatoração. Uma vantagem dos vetores de base ortogonais é que eles podem ser aproveitados mais facilmente para realizar recomendações fora da amostra para novos usuários e itens.

Esse problema também é chamado de completação de matriz indutiva.

A interpretação geométrica fornecida na Figura 3.6 é útil para entender por que vetores de base ortogonais são úteis na previsão de classificações ausentes. Uma vez que os vetores latentes tenham sido obtidos, pode-se projetar as informações nas classificações especificadas nos vetores latentes correspondentes; isso é muito mais fácil quando os vetores são mutuamente ortogonais. Considere uma situação em que SVD obteve fatores latentes  $U$  e  $V$ , respectivamente. As colunas de  $V$  definem um hiperplano  $k$ -dimensional,  $H_1$ , passando pela origem. Na Figura 3.6, o número de fatores latentes é 1 e, portanto, o único vetor latente (ou seja, hiperplano unidimensional) é mostrado. Se dois fatores tivessem sido usados, teria sido um plano.

Agora imagine um novo usuário cujas avaliações foram adicionadas ao sistema. Observe que esse novo usuário não está representado nos fatores latentes em U ou V. Considere o cenário em que o novo usuário especificou um total de  $h$  avaliações. O espaço de possibilidades de avaliações para esse usuário é um hiperplano ( $n \times h$ )-dimensional no qual os valores de  $h$  são fixos. Um exemplo é ilustrado na Figura 3.6, onde uma avaliação para Spartacus é fixa e o hiperplano é definido nas outras duas dimensões. Seja esse hiperplano denotado por  $H_2$ . O objetivo é então determinar o ponto em  $H_2$  que é o mais próximo possível de  $H_1$ . Esse ponto em  $H_2$  produz os valores de todas as outras avaliações. Três possibilidades surgem:

1. H1 e H2 não se intersectam: O ponto em H2 mais próximo de H1 é retornado. A menor distância entre um par de hiperplanos pode ser formulada como um problema simples de otimização de soma de quadrados.
  2. H1 e H2 se cruzam em um único ponto: Este caso é semelhante ao da Figura 3.6. Nesse caso, os valores das classificações do ponto de intersecção podem ser usados.
  3. H1 e H2 se cruzam em um hiperplano t-dimensional, onde  $t \geq 1$ : Todas as avaliações que sejam o mais próximas possível do hiperplano t-dimensional devem ser encontradas. Os valores médios das avaliações dos usuários correspondentes são retornados. Observe que esta abordagem combina métodos de fator latente e de vizinhança. A principal diferença em relação aos métodos de vizinhança é que a vizinhança é descoberta de forma mais refinada com o uso do feedback de modelos de fator latente.

A ortogonalidade apresenta vantagens significativas em termos de interpretabilidade geométrica. A capacidade de descobrir recomendações fora da amostra é um exemplo dessa vantagem.

#### 3.6.5.4 Exemplo de decomposição de valor singular

Para ilustrar o uso da decomposição em valores singulares, vamos aplicar essa abordagem ao exemplo da Tabela 3.2. Usaremos a abordagem iterativa de estimar as entradas ausentes repetidamente. O primeiro passo é preencher as entradas ausentes com a média de cada linha. Como resultado, a matriz de classificações  $Rf$  preenchida se torna:

$$\tilde{y} \quad 1 \tilde{y}1 \ 1 \tilde{y}1 \ 1 \tilde{y}1 \ 1 \tilde{y}0,2 \tilde{y}1 \tilde{y}1 \tilde{y}1 \ 01 \ 1 \tilde{y}1 \\ \tilde{y} \quad \tilde{y}1 \ 0 \tilde{y}1 \tilde{y}1 \tilde{y}1111 \tilde{y}1 \ 0,2 \tilde{y}1111 \quad \tilde{y}$$

Ao aplicar SVD truncado de classificação 2 à matriz e absorver a matriz diagonal dentro dos fatores do usuário, obtemos o seguinte:

$Rf$ $\ddot{y}$ $=$	$1,129 \quad 2,152 \quad 1,937$ $0,640 \quad 1,539 \quad 0,873 \quad \ddot{y}$ $\ddot{y}2,400 \quad \ddot{y}0,341$ $\ddot{y}2,105 \quad 0,461$	$0,431 \quad 0,246$ $\ddot{y}0,249 \quad 0,124 \quad \ddot{y}0,578 \quad 0,266$	$0,386 \quad \ddot{y}0,518 \quad \ddot{y}0,390 \quad \ddot{y}0,431 \quad \ddot{y}0,266 \quad 0,668$
		$9999$	
			$\ddot{y}$

Note que mesmo após a primeira iteração, uma estimativa razoável é obtida da falta de entradas. Em particular, os valores estimados são  $\hat{r}_{23} \approx 0,5581$ ,  $\hat{r}_{31} \approx 0,43$ ,  $\hat{r}_{36} \approx 0,43$ , e  $\hat{r}_{52} \approx 0,2095$ . É claro que essas entradas são tendenciosas pelo fato de que as entradas preenchidas inicialmente foram baseadas nas médias das linhas e, portanto, não refletiram com precisão as médias corretas. Portanto, na próxima iteração, preenchemos esses quatro valores ausentes no original matriz para obter a seguinte matriz:

$$\hat{R}_f = \begin{matrix} & 1 & -1 & 1 & 1 & -1 \\ \hat{y} & 1 & & 10,5581 & \hat{y} & -1 \\ & & & \hat{y} & \hat{y} & \hat{y} \\ & 0,43 & 1 & \hat{y} & \hat{y} & 0,43 \\ & & \hat{y} & 1 & \hat{y} & 1 \\ & & & \hat{y} & 1 & 1 \\ \cdots & & & \hat{y} & 0,2095 & \hat{y} \\ & & & & \hat{y} & \cdots \end{matrix}$$

Esta matriz ainda é tendenciosa, mas é melhor do que preencher entradas ausentes com médias de linhas. Na próxima iteração, aplicamos SVD com esta nova matriz, que é claramente um melhor ponto de partida. Ao aplicar todo o processo de SVD de classificação 2 novamente, obtemos a seguinte matriz na próxima iteração:

$$\hat{R}_f = \begin{matrix} & 1 & \hat{y} & 1 & \hat{y} & 1 & \hat{y} \\ \hat{y} & 1 & 10,9274 & \hat{y} & \hat{y} & 1 & \hat{y} \\ & 0,6694 & 1 & 1 & \hat{y} & 1 & \hat{y} \\ & & \hat{y} & 1 & \hat{y} & 1 & 1 \\ & & & \hat{y} & 1 & 0,5088 & \hat{y} \\ & & & & \hat{y} & 1 & \cdots \end{matrix}$$

Observe que as novas entradas estimadas foram alteradas na próxima iteração.

Os novos valores estimados são  $\hat{r}_{23} \approx 0,9274$ ,  $\hat{r}_{31} \approx 0,6694$ ,  $\hat{r}_{36} \approx 0,6694$  e  $\hat{r}_{52} \approx 0,5088$ .

Além disso, as entradas foram alteradas em menor grau do que na primeira iteração.

Aplicando o processo para mais uma iteração ao último valor de  $R_f$ , obtemos o seguinte:

$$\hat{R}_f = \begin{matrix} & 1 & -1 & 1 & \hat{y} & -1 \\ \hat{y} & 1 & & 10,9373 & \hat{y} & 1 & \hat{y} \\ & 0,7993 & & 1 & & \hat{y} & 0,7993 \\ & & -1 & & \hat{y} & 1 & \hat{y} \\ & & & \hat{y} & 1 & 0,6994 & \hat{y} \\ & & & & \hat{y} & 1 & \cdots \end{matrix}$$

Os valores estimados são agora  $\hat{r}_{23} \approx 0,9373$ ,  $\hat{r}_{31} \approx 0,7993$ ,  $\hat{r}_{36} \approx 0,7993$  e  $\hat{r}_{52} \approx 0,6994$ .

Observe que a mudança é ainda menor do que na iteração anterior. Na verdade, a mudança em a entrada  $\hat{r}_{23}$  é muito pequena. Ao longo de iterações sucessivas, as mudanças nas entradas tendem a se tornar cada vez menor, até que a convergência seja alcançada. As entradas resultantes podem ser usadas como valores previstos. Normalmente, não é necessário um grande número de iterações no processo. De fato, para classificar os itens de um determinado usuário, apenas 5 a 10 iterações podem ser suficientes. Exemplo particular, pode-se classificar corretamente as duas classificações ausentes para o usuário 3 após a primeira iteração. A abordagem também pode ser aplicada após a centralização média das linhas ou colunas, ou ambos. Essa abordagem tem o efeito de remover vieses do usuário e do item antes da estimativa process. A aplicação desses métodos de correção de viés geralmente tem um efeito positivo na previsão.

Não há garantia de que a abordagem converja para um ótimo global, especialmente se a pobreza pontos de inicialização foram usados. Isso é especialmente verdadeiro quando uma grande fração do entradas na matriz estão faltando. Nestes casos, o viés inicial pode ser significativo o suficiente para afetam a qualidade da solução final. Portanto, às vezes é aconselhável usar um método simples heurística, como um modelo de vizinhança, a fim de obter uma primeira estimativa da falta entradas. A escolha de uma estimativa tão robusta como ponto de partida acelerará a convergência,

e também levará a resultados mais precisos. Além disso, todo esse processo poderia ser facilmente aplicado à decomposição em valores singulares regularizada das matrizes preenchidas. A principal diferença é que cada iteração utiliza a decomposição em valores singulares regularizada da matriz atual, que é preenchida com os valores estimados. O trabalho em [541] pode ser usado como a sub-rotina relevante para a decomposição em valores singulares regularizada.

### 3.6.6 Fatoração de matrizes não negativas

A fatoração de matrizes não negativas (NMF) pode ser usada para matrizes de classificação não negativas. A principal vantagem dessa abordagem não é necessariamente a precisão, mas o alto nível de interpretabilidade que ela proporciona na compreensão das interações usuário-item.

A principal diferença em relação a outras formas de fatoração matricial é que os fatores U e V devem ser não negativos. Portanto, a formulação de otimização na fatoração matricial não negativa é apresentada da seguinte forma:

$$\text{Minimizar } J = \frac{1}{2} \|R - UV^T\|_F^2$$

sujeito a:

$$U \geq 0$$

$$V \geq 0$$

Embora a fatoração de matrizes não negativas possa ser usada para qualquer matriz de classificações não negativas (por exemplo, classificações de 1 a 5), suas maiores vantagens em termos de interpretabilidade surgem em casos em que os usuários têm um mecanismo para especificar uma preferência por um item, mas nenhum mecanismo para especificar uma não preferência. Tais matrizes incluem matrizes de classificações unárias ou matrizes nas quais as entradas não negativas correspondem à frequência da atividade. Esses conjuntos de dados também são chamados de conjuntos de dados de feedback implícito [260, 457]. Alguns exemplos dessas matrizes são os seguintes:

1. Em dados de transações de clientes, a compra de um item corresponde à expressão de uma preferência por um item. No entanto, não comprar um item não implica necessariamente uma não preferência, pois o usuário pode tê-lo comprado em outro lugar ou pode não estar ciente da existência do item. Quando valores são associados a transações, a matriz R pode conter números arbitrários não negativos. No entanto, todos esses números especificam o grau de preferência por um item, mas não indicam não preferência. Em outras palavras, as quantidades numéricas no feedback implícito indicam confiança, enquanto as quantidades numéricas no feedback explícito indicam preferência.
2. Semelhante ao caso da compra de um item, a navegação por um item pode ser indicativa de uma curtida. Em alguns casos, a frequência do comportamento de compra ou navegação pode ser quantificada como um valor não negativo.
3. Nos dados de cliques da Web, a seleção de um item corresponde a uma classificação unária de gostar de um item.
4. Um botão “curtir” no Facebook pode ser considerado um mecanismo para fornecer uma classificação unitária para um item.

A configuração de feedback implícito pode ser considerada o análogo de complementação de matriz para o problema de aprendizagem positivo-não rotulado (PU) em modelagem de classificação e regressão. Em modelagem de classificação e regressão, resultados razoáveis podem frequentemente ser obtidos tratando as entradas não rotuladas como pertencentes à classe negativa quando a classe positiva já é conhecida.

ser uma classe minoritária muito pequena. Da mesma forma, um aspecto útil de tais matrizes e problemas configurações é que muitas vezes é razoavelmente possível definir as entradas não especificadas como 0, em vez de tratá-los como valores ausentes. Por exemplo, considere um conjunto de dados de transações de clientes, em cujos valores indicam quantidades compradas por um cliente. Nesse caso, é razoável para definir um valor como 0, quando esse item não foi comprado pelo cliente. Portanto, neste caso, basta realizar a fatoração de matriz não negativa de uma matriz totalmente especificada, que é um problema padrão na literatura de aprendizado de máquina. Este problema também é conhecido como uma filtragem colaborativa de classe. Embora alguns trabalhos recentes argumentem que a falta os valores não devem ser definidos como 0 nesses casos [260, 457, 467, 468] para reduzir o viés, um considerável quantidade de trabalho na literatura mostra que soluções razoavelmente robustas podem ser obtidas por tratando as entradas ausentes como 0 no processo de modelagem. Este é especialmente o caso quando a probabilidade anterior de uma entrada ser 0 é muito grande. Por exemplo, no supermercado cenário, um cliente normalmente nunca compraria a grande maioria dos itens na loja. nesses casos, definindo os valores ausentes como 0 (na matriz inicial para fins de fatoração mas não na previsão final) resultaria em uma pequena quantidade de viés, mas tratar explicitamente as entradas como não especificadas na matriz inicial levará a uma maior complexidade da solução. A complexidade desnecessária sempre leva ao sobreajuste. Esses efeitos são especialmente significativos<sup>15</sup> em conjuntos de dados menores.

Observe que a formulação de otimização da fatoração de matrizes não negativas é uma formulação de otimização com restrições, que pode ser resolvida usando métodos padrão, como a relaxação La-grangiana. Embora uma derivação detalhada do algoritmo usado para fatoração de matrizes não negativas A fatoração de matrizes está além do escopo deste livro, portanto, remetemos o leitor a [22] para obter detalhes. Aqui, apresentamos apenas uma breve discussão sobre como a fatoração de matrizes não negativas é realizado.

Uma abordagem iterativa é usada para atualizar as matrizes U e V. Sejam  $u_{ij}$  e  $v_{ij}$ , respectivamente, as (i, j)-ésimas entradas das matrizes U e V. A seguinte atualização multiplicativa regras para  $u_{ij}$  e  $v_{ij}$  são usadas:

$$u_{ij} \leftarrow \frac{(RV)_{ij} u_{ij}}{(UV^T)_{ij} + } \quad \forall i \in \{1 \dots m\}, \forall j \in \{1 \dots k\} \quad (3.24)$$

$$v_{ij} \leftarrow \frac{(RTU)_{ij} v_{ij}}{(V^T U)_{ij} + } \quad \forall i \in \{1 \dots n\}, \forall j \in \{1 \dots k\} \quad (3.25)$$

Aqui, um pequeno valor como  $10^{-9}$  é usado para aumentar a estabilidade numérica. Todas as entradas em U e V no lado direito das equações de atualização são fixados aos valores obtidos no fim da iteração anterior durante o curso de uma iteração específica. Em outras palavras, todas as entradas em U e V são atualizadas “simultaneamente”. Às vezes, pequenos valores são adicionados a o denominador das equações de atualização para evitar a divisão por 0. As entradas em U e V são inicializados com valores aleatórios em (0, 1), e as iterações são executadas até a convergência. É possível obter melhores soluções realizando a inicialização de forma mais criteriosa caminho [331, 629].

Como no caso de outros tipos de fatoração de matrizes, a regularização pode ser usada para melhorar a qualidade da solução subjacente. A ideia básica é adicionar as penalidades  $\frac{\gamma_1 \|U\|_2^2}{2} + \frac{\gamma_2 \|V\|_2^2}{2}$  para a função objetivo. Aqui  $\gamma_1 > 0$  e  $\gamma_2 > 0$  são a regularização

<sup>15</sup>Esses efeitos são melhor compreendidos em termos de compensação entre viés e variância no aprendizado de máquina [22]. Definir os valores não especificados como 0 aumenta o viés, mas reduz a variância. Quando um grande número de entradas não são especificados e a probabilidade anterior de uma entrada ausente ser 0 é muito alta, os efeitos de variância podem dominar.

parâmetros. Isso resulta em uma modificação [474] das equações de atualização da seguinte forma:

$$\frac{uij \ddot{y} \text{ máx.}}{(RV)ij \ddot{y} \ddot{y}1uij} = \frac{0 \ddot{y} \ddot{y} \{1 \dots m\}, \ddot{y}j \ddot{y} \{1 \dots k\} uij}{(UV TV)ij +}, \quad (3.26)$$

$$\frac{vij \ddot{y} \text{ máx.}}{(RTU)ij \ddot{y} \ddot{y}2vij} = \frac{vij, 0}{(V UTU)ij +} \quad \ddot{y}i \ddot{y} \{1 \dots n\}, \ddot{y}j \ddot{y} \{1 \dots k\} \quad (3.27)$$

A função de maximização é usada para impor a não negatividade e o pequeno termo aditivo  $\ddot{y}10\ddot{y}9$  no denominador é usado para garantir a estabilidade numérica. Os parâmetros  $\ddot{y}1$  e  $\ddot{y}2$  podem ser determinados usando a mesma abordagem descrita anteriormente. Em vez de usar o método de gradiente descendente, também se pode usar métodos de mínimos quadrados alternados nos quais A regressão linear não negativa é utilizada. A regularização de Tikhonov pode ser usada dentro do modelo de regressão para evitar sobreajuste. Detalhes do método dos mínimos quadrados alternados para A fatoração de matriz não negativa pode ser encontrada em [161, 301]. Os principais desafios com A desvantagem desses métodos prontos para uso é a falta de eficiência computacional com grandes matrizes de classificação, pois todas as entradas são tratadas como observadas. Na seção 3.6.6.3, discutiremos como essas questões podem ser abordadas.

### 3.6.6.1 Vantagens da Interpretabilidade

A principal vantagem da fatoração de matrizes não negativas é que um alto grau de interpretabilidade é alcançado na solução. É sempre útil parear sistemas de recomendação com explicações para as recomendações, e isso é fornecido por uma matriz não negativa fatoração. Para melhor compreender este ponto, considere uma situação em que a A matriz de preferências contém quantidades de itens comprados pelos clientes. Um exemplo de um brinquedo A Figura 3.12 ilustra uma matriz  $6 \times 6$  com 6 itens e 6 clientes . É evidente que são duas classes de produtos correspondentes aos laticínios e às bebidas, respectivamente. É claro que o comportamento de compra do cliente é altamente correlacionado com base nas classes de itens, embora todos os clientes pareçam gostar de suco. Essas classes de itens são chamadas de aspectos. As matrizes fatoriais correspondentes também fornecem uma interpretabilidade clara sobre o afinidade dos clientes e itens com esses aspectos. Por exemplo, os clientes de 1 a 4 gostam de laticínios produtos, enquanto os clientes de 4 a 6 gostam de bebidas. Isso se reflete claramente na matriz fator-usuário U  $6 \times 2$ . Neste exemplo simplificado, mostramos todos os valores fatorados em U e V para ser integral para simplicidade visual. Na prática, os valores ótimos são quase sempre reais números. A magnitude da entrada de um usuário em cada uma das duas colunas quantifica sua nível de interesse no aspecto relevante. Da mesma forma, a matriz fatorial V mostra como os itens estão relacionados aos vários aspectos. Portanto, neste caso, a condição  $r_{ij} \ddot{y} s=1 u_{is} \cdot v_{js}$  pode ser semanticamente interpretado em termos dos aspectos  $k = 2$ :

$$r_{ij} \ddot{y} (\text{Afinidade do usuário } i \text{ com o aspecto lácteo}) \times (\text{Afinidade do item } j \text{ com o aspecto lácteo}) \\ + (\text{Afinidade do usuário } i \text{ com o aspecto de bebidas}) \times (\text{Afinidade do item } j \text{ com o aspecto de bebidas})$$

Esta forma de prever o valor de  $r_{ij}$  mostra uma decomposição de "soma de partes" da matriz. Cada uma dessas partes também pode ser vista como um coagrupamento de itens de usuário. Este também é um dos as razões pelas quais a fatoração de matrizes não negativas é frequentemente usada em agrupamentos. Na prática aplicações, muitas vezes é possível olhar para cada um desses clusters e interpretar semanticamente as associações entre usuários e itens. Quando rótulos semânticos podem ser anexados manualmente para os vários clusters, o processo de fatoração fornece uma explicação clara das classificações em termos das contribuições de vários "gêneros" semânticos de itens.

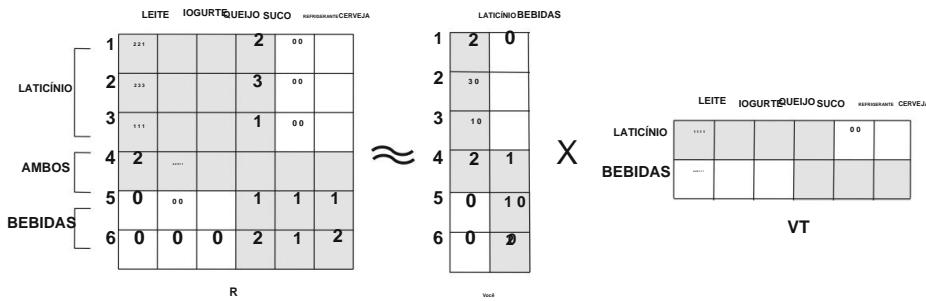


Figura 3.12: Um exemplo de fatoração de matriz não negativa

Esta decomposição da soma das partes pode ser representada matematicamente da seguinte forma.

Fatoração da matriz T rank-k Produto da matriz UV em termos pode ser decomposto em k componentes expressando o

das k colunas  $U_i$  e  $V_i$ , respectivamente, de  $U$  e  $V$ :

$$UV^T = \sum_{i=1}^k \frac{U_i V_i}{U_i V_i} T \quad (3.28)$$

Cada matriz  $m \times n$   $U_i V_i^T$  é uma matriz de classificação 1 que corresponde a um aspecto nos dados.

Devido à natureza interpretável da decomposição não negativa, é fácil mapear estes aspectos dos clusters. Por exemplo, os dois componentes latentes do exemplo acima mencionado correspondentes a produtos lácteos e bebidas, respectivamente, são ilustrados na Figura 3.13.

Observe que a Equação 3.28 decompõe a fatoração em termos das colunas de U e V, enquanto a Equação 3.14 é uma maneira diferente de entender a fatoração em termos de linhas de U e V. Para uma determinada combinação de usuário-item, a previsão de classificação é dada por soma das contribuições desses aspectos, e pode-se até obter uma melhor compreensão por que uma classificação é prevista de uma certa maneira pela abordagem.

### 3.6.6.2 Observações sobre fatoração com feedback implícito

A fatoração de matrizes não negativas é particularmente adequada para matrizes de feedback implícitas em que as classificações indicam preferências positivas. Ao contrário dos conjuntos de dados de feedback explícito, não é possível ignorar as entradas ausentes no modelo de otimização devido à falta de feedback negativo em tais dados. Vale ressaltar que a fatoração de matriz não negativa o modelo trata as entradas ausentes como feedback negativo, definindo-as como 0s. Não fazer isso seria aumentar consideravelmente o erro nas entradas não observadas. Para entender este ponto, considere uma matriz de classificações unárias na qual as curtidas são especificadas por 1s. A fatoração mostrada na Figura 3.14 fornecerá 100% de precisão em uma matriz unária arbitrária quando calculada apenas sobre entradas observadas. Isso ocorre porque a multiplicação de  $U$  e  $V$  na Figura 3.14<sup>T</sup> leva a uma matriz contendo apenas 1s e nenhum 0s. É claro que tal fatoração terá erro muito alto nas entradas não observadas porque muitas entradas não observadas podem corresponder a preferências negativas. Este exemplo é uma manifestação de overfitting causado pela falta de dados de feedback negativo. Portanto, para matrizes de classificação em que as preferências negativas são ausentes e sabe-se que as preferências negativas superaram em muito as positivas, é importante tratar as entradas ausentes como 0s. Por exemplo, em um conjunto de dados de transações de clientes,

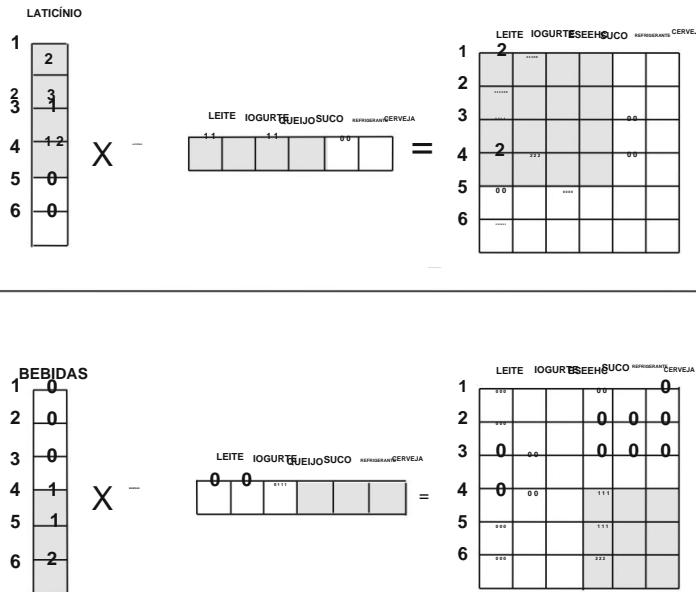


Figura 3.13: Interpretação da soma das partes do NMF

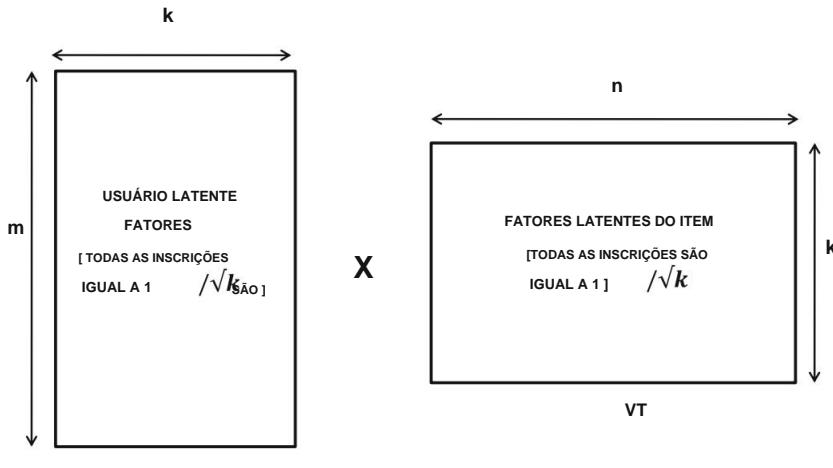


Figura 3.14: Overfitting causado pela ignorância de entradas ausentes em uma matriz unária

se os valores indicarem as quantidades compradas por vários usuários e a maioria dos itens não for comprada por padrão, então é possível aproximar o valor de uma entrada ausente como 0.

### 3.6.6.3 Problemas computacionais e de ponderação com feedback implícito

O tratamento de entradas ausentes como 0s leva a desafios computacionais com matrizes grandes. Existem várias soluções para esse dilema. Por exemplo, uma amostra das entradas ausentes pode ser tratada como 0s. A solução de gradiente descendente para o caso amostrado é semelhante à discutida na próxima seção. É possível melhorar ainda mais a precisão com uma abordagem de conjunto. A matriz é fatorada várias vezes com uma amostra diferente de 0s, e cada fatoração é usada para prever um valor (ligeiramente diferente) da classificação. As diferentes previsões de uma classificação específica são então calculadas para criar o resultado final. Usando amostras de tamanhos variados, também é possível ponderar as entradas de feedback negativo de forma diferente das entradas de feedback positivo. Tal abordagem pode ser importante em cenários sensíveis a custos, onde falsos positivos e falsos negativos são ponderados de forma diferente. Normalmente, as entradas zero devem ter uma ponderação menor do que as entradas diferentes de zero e, portanto, a subamostragem das entradas zero é útil.

Também é possível incorporar tais pesos diretamente na função objetivo e tratar todas as entradas ausentes como 0s. Os erros nas entradas zero devem ser ponderados menos do que aqueles nas entradas diferentes de zero na função objetivo para evitar que as entradas zero dominem a otimização. Os pesos relativos podem ser determinados usando validação cruzada em relação a uma medida de precisão específica. Alternativamente, o trabalho em [260] sugere a seguinte heurística para selecionar o peso  $w_{ij}$  da entrada  $(i, j)$ :

$$w_{ij} = 1 + \bar{y} \cdot r_{ij} \quad (3.29)$$

Observe que todos os valores ausentes de  $r_{ij}$  são tratados como 0s na Equação 3.29, e um valor típico de  $\bar{y}$  é 40. Essa abordagem também funciona em cenários onde as classificações  $r_{ij}$  representam quantidades que são compradas, em vez de indicadores binários. Nesses casos, os pesos  $w_{ij}$  são calculados tratando essas quantidades como as classificações na Equação 3.29, mas a matriz fatorada é uma matriz indicadora binária derivada  $R_I$  da matriz de quantidade  $R = [r_{ij}]$ . Essa matriz indicadora  $R_I$  é derivada de  $R$  copiando as entradas zero e substituindo as entradas diferentes de zero por 1s. Essa abordagem de fatoração ponderada da matriz indicadora é, portanto, ligeiramente diferente do exemplo da Figura 3.12, que foi apresentada apenas para fins ilustrativos.

Ao trabalhar com entradas ponderadas, é possível modificar métodos de descida de gradiente estocástico com pesos (cf. seção 6.5.2.1 do Capítulo 6). No entanto, o problema é que as matrizes de feedback implícitas são totalmente especificadas, e muitos dos métodos de descida de gradiente não são mais computacionalmente viáveis em cenários de larga escala. Um método ALS (ponderado) eficiente foi proposto em [260] para o processo de fatoração, a fim de evitar o desafio computacional de lidar com o grande número de entradas nulas. Embora essa abordagem não imponha não negatividade aos fatores, ela pode ser facilmente generalizada para o cenário não negativo.

### 3.6.6.4 Classificações com gostos e desgostos

Nossa discussão sobre fatoração de matrizes não negativas até agora se concentrou apenas em matrizes de feedback implícitas, nas quais há um mecanismo para especificar uma preferência por um item, mas nenhum mecanismo para especificar uma aversão. Como resultado, as matrizes de "classificações" subjacentes são sempre não negativas. Embora seja possível usar a fatoração de matrizes não negativas para classificações nominalmente não negativas (por exemplo, de 1 a 5), que especificam explicitamente tanto as preferências quanto as aversões, ha-

Não há vantagens especiais de interpretabilidade no uso da fatoração matricial não negativa nesses casos. Por exemplo, a escala de classificação pode variar de 1 a 5, em que um valor de 1 indica extrema antipatia. Nesse caso, não se pode tratar as entradas não especificadas como 0, e deve-se trabalhar apenas com o conjunto de entradas observadas. Como antes, denotamos o conjunto de entradas observadas na matriz de classificações  $R = [r_{ij}]$  por  $S$ :

$$S = \{(i, j) : r_{ij} \text{ é observado}\} \quad (3.30)$$

O problema de otimização (com regularização) é declarado em termos dessas entradas observadas da seguinte forma:

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j) \in S} r_{ij} (\hat{y}_i - \sum_{s=1}^k u_{is} v_{js})^2 + \frac{\lambda}{2} \sum_{i=1}^m \sum_{s=1}^k u_{is}^2 + \sum_{j=1}^n \sum_{s=1}^k v_{js}^2$$

sujeito a:

$$U \geq 0$$

$$V \geq 0$$

Esta formulação é semelhante à formulação regularizada na fatoração de matrizes irrestrita. A única diferença é a adição das restrições de não negatividade. Nesses casos, são necessárias modificações nas equações de atualização usadas para a fatoração de matrizes irrestrita. Primeiro, deve-se inicializar as entradas de  $U$  e  $V$  para valores não negativos em  $(0, 1)$ . Então, uma atualização semelhante pode ser feita, como na seção sobre fatoração de matrizes irrestrita. De fato, as equações de atualização na seção 3.6.4.2 podem ser usadas diretamente. A principal modificação é garantir que a não negatividade seja mantida durante as atualizações. Se qualquer componente de  $U$  ou  $V$  violar a restrição de não negatividade como resultado da atualização, então ela é definida como 0. As atualizações são realizadas para convergência como em todos os métodos de descida de gradiente estocástico.

Outras metodologias de solução são frequentemente utilizadas para calcular soluções ótimas para tais modelos. Por exemplo, é possível adaptar uma abordagem de mínimos quadrados alternados à fatoração de matrizes não negativas. A principal diferença é que os coeficientes da regressão de mínimos quadrados são limitados a serem não negativos. Uma ampla variedade de métodos de descida de gradiente projetada, descida de coordenadas e programação não linear também estão disponíveis para lidar com tais modelos de otimização [76, 357].

No cenário em que as classificações especificam gostos e desgostos, a fatoração matricial não negativa não apresenta vantagens especiais sobre a fatoração matricial irrestrita em termos de interpretabilidade. Isso ocorre porque não é mais possível interpretar a solução a partir de uma perspectiva de soma de partes. Por exemplo, a adição de três classificações de desgosto não pode ser interpretada como levando a uma classificação de gosto. Além disso, devido à adição de restrições não negativas, a qualidade da solução é reduzida em relação à fatoração matricial irrestrita quando calculada sobre as entradas observadas. No entanto, isso nem sempre significa que a qualidade da solução será pior quando calculada sobre as entradas não observadas. Em cenários reais, relacionamentos positivos entre usuários e itens são mais importantes do que relacionamentos negativos entre usuários e itens. Como resultado, restrições não negativas frequentemente introduzem um viés que é benéfico para evitar o sobreajuste. Como no caso da fatoração matricial irrestrita, também é possível incorporar vieses de usuário e item para melhorar ainda mais o desempenho da generalização.

### 3.6.7 Compreendendo a família de fatoração de matrizes

É evidente que as várias formas de fatoração de matrizes nas seções anteriores compartilham uma muito em comum. Todas as formulações de otimização mencionadas minimizam o Frobenius normas da matriz residual ( $R \ddot{\gamma} UV^T$ ) sujeito a várias restrições sobre o fator matrizes  $U$  e  $V$ . Observe que o objetivo da função objetivo é tornar  $UV$  aproximado  $R^T$  a matriz de classificação  $R$  o mais próximo possível. As restrições nas matrizes fatoriais alcançam diferentes propriedades de interpretabilidade. De fato, a família mais ampla de modelos de fatoração de matrizes pode usar qualquer outra função objetivo ou restrição para forçar uma boa aproximação. Isto família mais ampla pode ser escrita da seguinte forma:

$$\begin{aligned} & \text{Otimizar } J = [\text{Função objetivo quantificando a correspondência entre } R \text{ e } UV^T] \\ & \text{sujeito a:} \end{aligned}$$

Restrições em  $U$  e  $V$

A função objetivo de um método de fatoração de matriz é algumas vezes chamada de função de perda, quando está na forma de minimização. Observe que a formulação de otimização pode ser um problema de minimização ou de maximização, mas o objetivo do objetivo é sempre forçar  $R$  a corresponder o mais próximo possível a  $UV$ . A norma de Frobenius é uma exemplo de um objetivo de minimização e alguns métodos de fatoração de matriz probabilística usar uma formulação de maximização, como a função objetivo de máxima verossimilhança. Na maioria casos, regularizadores são adicionados à função objetivo para evitar overfitting. Os vários restrições muitas vezes impõem diferentes tipos de interpretabilidade aos fatores. Dois exemplos de tal interpretabilidade são a ortogonalidade (que fornece interpretabilidade geométrica) e não negatividade (que fornece interpretabilidade de soma de partes). Além disso, embora restrições aumentam o erro nas entradas observadas, às vezes podem melhorar a erros nas entradas não observadas quando elas têm uma interpretação semântica significativa. Isso ocorre porque as restrições reduzem a variância<sup>16</sup> nas entradas não observadas, ao mesmo tempo que aumentam viés. Como resultado, o modelo tem melhor generalização. Por exemplo, corrigindo as entradas em uma coluna em cada um de  $U$  e  $V$  para uns quase sempre resulta em melhor desempenho (cf. seção 3.6.4.5). A seleção das restrições corretas a serem usadas geralmente depende dos dados e requer insights sobre o domínio de aplicação em questão.

Existem outras formas de fatoração nas quais se pode atribuir interpretabilidade probabilística a os fatores. Por exemplo, considere um cenário em que uma matriz de classificações unárias não negativas  $R$  é tratado como uma distribuição de frequência relativa, cujas entradas somam 1.

$$\sum_{i=1}^m \sum_{j=1}^n r_{ij} = 1 \quad (3.31)$$

Observe que é fácil escalar  $R$  para somar 1, dividindo-o pela soma de suas entradas. Tal uma matriz pode ser fatorada de maneira semelhante a SVD:

$$\begin{aligned} R &\ddot{\gamma} (Qk\ddot{\gamma} k)P^T \\ &= UV^T \end{aligned}$$

Como em SVD, a matriz diagonal  $\ddot{\gamma} k$  é absorvida na matriz de fatores do usuário  $U = Qk\ddot{\gamma} k$ , e a matriz fatorial do item  $V$  é definida como  $Pk$ . A principal diferença de SVD é que as colunas de  $Qk$  e  $Pk$  não são ortogonais, mas são valores não negativos que somam 1. Além disso, as entradas da matriz diagonal  $\ddot{\gamma} k$  não são negativas e também somam 1.

<sup>16</sup>Consulte o Capítulo 6 para uma discussão sobre a compensação entre viés e variância na filtragem colaborativa.

Tabela 3.3: A família de métodos de fatoração de matrizes

Restrições de Método		Vantagens/Desvantagens Objetivas
Sem restrições	Sem restrições	Frobenius Solução de altíssima qualidade + regularizador A Bom para a maioria das matrizes regularização previne sobreajuste Interpretabilidade ruim
SVD	Base Ortogonal	Frobenius Boa + regularizador B Interpretabilidade visual Recomendações fora da amostra Bom para matrizes densas Interpretabilidade semântica pobre Subótimo em matrizes esparsas
Margem Máxima	Sem restrições	Perda de dobradiça + regularizador de margem Solução de altíssima qualidade Resiste ao overfitting Semelhante a irrestrito Interpretabilidade ruim Bom para classificações discretas
NMF	Não-negatividade	Frobenius Solução de boa qualidade + regularizador perde interpretabilidade com avaliações de gosto/não gosto Menos overfitting em alguns casos Melhor para feedback implícito
PLSA	Solução de Máxima Qualidade Não Negativa	Probabilidade Alta interpretabilidade semântica + regularizador perde interpretabilidade com avaliações de gosto/não gosto Menos overfitting em alguns casos Melhor para feedback implícito

Tal fatoração tem uma interpretação probabilística; as matrizes  $Q_k$ ,  $P_k$  e  $\hat{y}_k$  contêm os parâmetros probabilísticos de um processo gerativo que cria a matriz de classificações. A função objetivo aprende os parâmetros deste processo gerativo para que a probabilidade do processo gerativo que cria a matriz de classificação é o maior possível. Portanto, o A função objetivo está na forma de maximização. Curiosamente, esse método é conhecido como Análise Semântica Latente Probabilística (PLSA) e pode ser visto como uma variante probabilística da fatoração de matrizes não negativas. Claramente, a natureza probabilística desta fatoração fornece-lhe um tipo diferente de interpretabilidade. Uma discussão detalhada do PLSA pode ser encontrado em [22]. Em muitas dessas formulações, técnicas de otimização, como gradiente descendida (ou subida) são úteis. Portanto, a maioria desses métodos usa ideias muito semelhantes em termos de formulação do problema de otimização e da metodologia de solução subjacente.

Da mesma forma, a fatoração de margem máxima [180, 500, 569, 624] toma emprestadas ideias do suporte máquinas vetoriais para adicionar um regularizador de margem máxima à função objetivo e alguns de suas variantes [500] são particularmente eficazes para classificações discretas. Esta abordagem compartilha uma número de semelhanças conceituais com o método de fatoração de matriz regularizada discutido na seção 3.6.4. Na verdade, o regularizador de margem máxima não é muito diferente daquele usado na fatoração de matriz irrestrita. No entanto, a perda de dobradiça é usada para quantificar a erros na aproximação, em vez da norma de Frobenius. Embora esteja além do escopo deste livro para discutir essas variantes em detalhes, uma discussão pode ser encontrada em [500, 569]. O o foco na maximização da margem geralmente fornece fatoração de qualidade superior a alguns os outros modelos na presença de dados propensos a overfitting. Na Tabela 3.3, fornecemos uma lista de vários modelos de fatoração e suas características. Na maioria dos casos, a adição de

Restrições como a não negatividade podem reduzir a qualidade da solução subjacente nas entradas observadas, pois reduzem o espaço de soluções viáveis. É por isso que se espera que a fatoração irrestrita e a fatoração com margem máxima apresentem a mais alta qualidade de ótimos globais. No entanto, como o ótimo global não pode ser facilmente encontrado na maioria dos casos pelos métodos (iterativos) disponíveis, um método com restrições pode, às vezes, ter um desempenho melhor do que um método sem restrições. Além disso, a precisão sobre entradas observadas pode ser diferente daquela sobre entradas não observadas devido aos efeitos do sobreajuste. De fato, restrições de não negatividade podem, às vezes, melhorar a precisão sobre entradas não observadas em alguns domínios. Algumas formas de fatoração, como a NMF, não podem ser aplicadas a matrizes com entradas negativas. Claramente, a escolha do modelo depende da configuração do problema, do ruído nos dados e do nível desejado de interpretabilidade. Não existe uma solução única que possa atingir todos esses objetivos. Uma compreensão cuidadosa do domínio do problema é importante para a escolha do modelo correto.

### 3.7 Integrando modelos de fatoração e vizinhança

---

Métodos baseados em vizinhança são geralmente considerados inherentemente diferentes de outros modelos de otimização devido à sua natureza heurística. No entanto, foi demonstrado na seção 2.6 do Capítulo 2 que métodos de vizinhança também podem ser entendidos no contexto de modelos de otimização. Esta é uma estrutura bastante conveniente, pois abre caminho para a integração de modelos de vizinhança com outros modelos de otimização, como modelos de fatores latentes. A abordagem em [309] integra o modelo item a item da seção 2.6.2 do Capítulo 2 com o modelo SVD++ da seção 3.6.4.6.

Suponha que a matriz de classificações  $R$  seja centrada na média. Em outras palavras, a média global  $\bar{y}$  da matriz de classificações já foi subtraída de todas as entradas, e todas as previsões serão realizadas com base em valores centrados na média. A média global  $\bar{y}$  pode ser adicionada novamente aos valores previstos em uma fase de pós-processamento. Com essa suposição na matriz de classificações  $R = [r_{ij}]$ , revisitaremos as várias partes do modelo.

#### 3.7.1 Estimador de linha de base: um modelo centrado em viés não personalizado

O modelo não personalizado centrado em viés prevê as avaliações (centradas na média) em  $R$  puramente como uma adição de vieses do usuário e do item. Em outras palavras, as avaliações são explicadas completamente pela generosidade do usuário e pela popularidade do item, em vez de interesses específicos e personalizados dos usuários serem a variável de viés para o usuário  $i$  e item nos itens. Seja  $b_{user}$  a variável de viés para o item  $j$ . Então, a previsão desse modelo é a seguinte:

$$= b_{user} + \text{item } r^*_{ij} \quad (3.32)$$

Sejam  $S$  os pares de índices correspondentes às entradas observadas na matriz de classificações.

$$S = \{(i, j) : r_{ij} \text{ é observado}\} \quad (3.33)$$

Então,  $b_{user}$  pode ser determinado formulando uma função objetivo sobre os  $j$  erros  $e_{ij} = r_{ij} - \bar{y} - r^*_{ij}$  nas entradas observadas da seguinte forma:

$$\text{Minimizar } J = \frac{1}{2} \sum_{(i,j) \in S} (r_{ij} - \bar{y} - r^*_{ij})^2 + \frac{\bar{y}}{2} \sum_{i=1}^n (b_{user i})^2 + \frac{\bar{y}}{2} \sum_{j=1}^m (b_{item j})^2$$

Este problema de otimização pode ser resolvido por meio da descida do gradiente usando a seguinte atualização de regras sobre cada entrada observada  $(i, j)$  em  $S$  em um método de descida de gradiente estocástico:

$$\begin{aligned} \hat{y}_{\text{ônibus}} &= \hat{y}_{\text{ônibus}} + \hat{y}(e_{ij} \hat{y}_{\text{buser}}) \\ \hat{y}_{\text{item}} &= \hat{y}_{\text{item}} + \hat{y}(e_{ij} \hat{y}_{\text{bitem}}) \end{aligned}$$

A estrutura básica do método de gradiente descendente é semelhante à da Figura 3.9, exceto para as diferenças na escolha de variáveis de otimização e etapas de atualização correspondentes.

Curiosamente, um modelo puramente centrado no viés pode frequentemente fornecer previsões razoáveis, apesar de sua natureza não personalizada. Isso é especialmente o caso quando a quantidade de dados de classificação é limitado. Após resolver os valores de  $b_{user}$  e  $b_{item}$ , definimos  $B_{ij}$  para o valor previsto de  $r_{ij}$  de acordo com a Equação 3.32. Este valor de  $B_{ij}$  é então tratado como uma constante ao longo desta seção e não como uma variável. Portanto, a primeira etapa do modelo integrado a solução é determinar o valor constante  $B_{ij}$  resolvendo o modelo não personalizado. Isso O modelo não personalizado também pode ser visto como um estimador de linha de base porque  $B_{ij}$  é um modelo aproximado estimativa de base para os valores da classificação  $r_{ij}$ . Em geral, subtraindo o valor de  $B_{ij}$  de cada entrada observada  $r_{ij}$  resulta em uma nova matriz que muitas vezes pode ser estimada de forma mais robusta pela maioria dos modelos discutidos nas seções e capítulos anteriores. Esta seção fornece um exemplo específico de como os modelos de vizinhança podem ser ajustados com o uso da linha de base estimador embora sua aplicabilidade seja muito mais ampla.

### 3.7.2 Porção de vizinhança do modelo

Replicamos a relação de previsão baseada na vizinhança da Equação 2.29 (cf. seção 2.6.2 do Capítulo 2) da seguinte forma:

$$r_{ij} = \hat{y}_{\text{ônibus}} + b_{item} + \frac{\sum_{l \in Q_j(i)} \hat{y}_{\text{bitem}}}{|Q_j(i)|} \cdot (r_{il} \hat{y}_{\text{ônibus}} - \hat{y}_{\text{bitem}}) \quad (3.34)$$

Embora a equação acima mencionada seja a mesma da Equação 2.29 do Capítulo 2, as notações de subscrito foram alteradas para garantir a consistência com os modelos de fatores latentes nesta seção. Aqui  $b_{user}$  é o viés do usuário e o  $b_{item}$  é o viés do item. A variável  $w_{item}$  representa o coeficiente de regressão item-item entre o item  $i$  e o item  $j$ . O conjunto  $Q_j(i)$  representa o subconjunto dos  $K$  itens mais próximos do item  $j$ , que foram avaliados pelo usuário  $i$ . Além disso, uma das ocorrências de  $b_{user}$  +  $b_{item}$  na Equação 3.34 é substituída por o valor constante  $B_{il}$  (derivado usando a abordagem da seção anterior). O resultado da previsão é a seguinte:

$$r_{ij} = \hat{y}_{\text{ônibus}} + b_{item} + \frac{\sum_{l \in Q_j(i)} \hat{y}_{\text{bitem}}}{|Q_j(i)|} \cdot (r_{il} \hat{y}_{\text{Bil}}) \quad (3.35)$$

Vale ressaltar que as variáveis de viés  $b_{user}$  e  $b_{item}$  são parâmetros a serem otimizados, enquanto  $B_{il}$  é uma constante. Pode-se configurar um modelo de otimização que some o quadrado de erros  $\sum_{i,j} (r_{ij} - \hat{y}_{ij})^2$  além dos termos de regularização. Uma descida de gradiente estocástica abordagem  $ij$  pode ser usada para determinar uma solução para a parte de vizinhança do modelo.

<sup>17</sup>Observe que usamos a variável maiúscula  $K$  para representar o tamanho da vizinhança que define  $Q_j(i)$ . Este é um desvio da seção 2.6.2 do Capítulo 2. Usamos a variável  $k$  em minúsculas para representar a dimensionalidade das matrizes fatoriais. Os valores de  $k$  e  $K$  são geralmente diferentes.

Os passos resultantes da descida do gradiente são os seguintes:

$$\begin{aligned} \hat{o}_{\text{ônibus}} & \hat{y}_{\text{ônibus}} + \hat{y}(e_{ij} \hat{y}_{\text{ybuser}}) \\ \hat{b}_{\text{item}} & \hat{y}_{\text{item}} + \hat{y}(e_{ij} \hat{y}_{\text{ybitem}}) \\ \hat{b}_{\text{branco}} & \hat{y}_{\text{witem}} + \hat{y}_2 \frac{e_{ij} \cdot (r_{il} \hat{y}_{\text{Bil}})}{|Q_j(i)|} - \hat{y}_2 \cdot \hat{b}_{\text{branco}} \hat{y}_{\text{Qj(i)}} \end{aligned}$$

Este modelo de vizinhança pode ser melhorado ainda mais com feedback implícito, introduzindo variáveis de feedback implícitas item-item  $c_{lj}$ . A ideia básica é que se o item  $j$  for avaliado em conjunto com muitos itens vizinhos do mesmo usuário  $i$ , então isso deve ter um impacto na classificação prevista  $\hat{r}_{ij}$ . Este impacto é independente dos valores reais das classificações destes itens vizinhos de  $j$ . Este impacto é igual a  $\frac{\hat{y}_{Qj(i)} c_{lj}}{|Q_j(i)|}$ . Observe que a escala do ex-  $\hat{y}|Q_j(i)|$

A compressão com  $|Q_j(i)|$  é feita para ajustar os vários níveis de escassez em diferentes combinações usuário-item. Então, o modelo de vizinhança com feedback implícito pode ser escrito da seguinte forma:

$$\hat{r}_{ij} = \hat{o}_{\text{ônibus}} + \hat{b}_{\text{item}} + \frac{\hat{b}_{\text{branco}} \cdot (r_{il} \hat{y}_{\text{Bil}})}{|Q_j(i)|} + \frac{\hat{y}_{Qj(i)} c_{lj}}{|Q_j(i)|} \quad (3.36)$$

Ao criar um modelo de otimização de mínimos quadrados com relação ao erro  $e_{ij} = r_{ij} - \hat{r}_{ij}$ , é possível calcular o gradiente e derivar os passos estocásticos de descida do gradiente. Isso resulta em o seguinte conjunto modificado de atualizações:

$$\begin{aligned} \hat{o}_{\text{ônibus}} & \hat{y}_{\text{ônibus}} + \hat{y}(e_{ij} \hat{y}_{\text{ybuser}}) \\ \hat{b}_{\text{item}} & \hat{y}_{\text{item}} + \hat{y}(e_{ij} \hat{y}_{\text{ybitem}}) \\ \hat{b}_{\text{branco}} & \hat{y}_{\text{witem}} + \hat{y}_2 \frac{e_{ij} \cdot (r_{il} \hat{y}_{\text{Bil}})}{|Q_j(i)|} - \hat{y}_2 \cdot \hat{b}_{\text{branco}} \hat{y}_{\text{Qj(i)}} \\ c_{lj} & \hat{y}_{\text{clj}} + \hat{y}_2 \frac{e_{ij}}{|Q_j(i)|} - \hat{y}_2 \cdot c_{lj} \hat{y}_{\text{Qj(i)}} \end{aligned}$$

O trabalho em [309] assume uma estrutura mais geral, na qual a matriz de feedback implícita não é necessariamente derivado apenas da matriz de classificação. Por exemplo, um varejista pode criar a matriz de classificações implícitas com base em usuários que navegaram, classificaram ou compraram um item. Esta generalização é relativamente simples de incorporar em nossos modelos, alterando o termo final da Equação 3.36 para  $\frac{\hat{y}_{Q_{j(i)}}(i) c_{lj}}{|\hat{y}_{Q_{j(i)}}(i)|}$ . Aqui,  $Q_{j(i)}$  é o conjunto de vizinhos mais próximos de usuário  $i$  (com base em avaliações explícitas), que também forneceu algum tipo de feedback implícito para o item  $j$ . Esta modificação também pode ser aplicada à parte do fator latente do modelo, embora trabalhemos consistentemente com a suposição simplificada de que o feedback implícito a matriz é derivada da matriz de classificações.

### 3.7.3 Porção do fator latente do modelo

A previsão acima mencionada é feita com base no modelo de vizinhança. Um modelo de fator latente correspondente é introduzido na seção 3.6.4.6, no qual o feedback implícito é

integrado com informações de classificação para fazer previsões. Replicamos a Equação 3.21 dessa seção aqui:

$$\hat{r}_{ij}^k = \frac{\text{interface do usuário} + y_{hs}}{h_{yi}} \cdot v_{js} \quad (3.37)$$

Como na seção 3.6.4.6,  $I_i$  representa o conjunto de itens avaliados pelo usuário  $i$ . A matriz  $m \times (k + 2)$   $Y = [y_{hs}]$  contém as variáveis de feedback implícitas, e sua construção é descrita na seção 3.6.4.6. Além disso, a coluna  $(k + 2)$  de  $U$  contém apenas 1s, a coluna  $(k + 1)$  de  $V$  contém apenas 1s, e as duas últimas colunas de  $Y$  são 0s.

Observe que o lado direito da Equação 3.37 já considera os vieses do usuário e do item. Como as duas últimas colunas das matrizes fatoriais contêm as variáveis de viés, o componente  $s = 1$  uisvjs da Equação 3.37 inclui os termos de viés.

### 3.7.4 Integrando as porções de vizinhança e fator latente

Agora é possível integrar os dois modelos nas Equações 3.36 e 3.37 para criar um único valor previsto da seguinte forma:

Observe que os termos de viés iniciais buser da Equação [\(1\)](#) estão ausentes aqui porque estão incluídos no termo final correspondente ao modelo de fator latente. Os mesmos vieses de usuário e item agora são compartilhados por ambos os componentes do modelo.

O problema de otimização correspondente, que minimiza o erro quadrático agregado =  $(r_{ij} - \hat{y}_{ij})^2$  sobre as entradas no (conjunto observado) S é o seguinte:

$$\text{Minimize } J = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{m_i} \|x_i - \hat{x}_j\|^2 + \frac{\lambda}{2} \sum_{i=1}^n \|x_i\|^2$$

sujeito a: (k

+ 2)<sup>a</sup> coluna de U contém apenas 1s ( $k + 1$ )<sup>a</sup>

coluna de V contém apenas 1s As duas

últimas colunas de Y contêm apenas 0s O

valor de  $\hat{r}_{ij}$  na função objetivo mencionada pode ser materializado com a ajuda da Equação 3.38. Como em todos os modelos de fatores latentes, a soma dos quadrados das variáveis de otimização é incluída para regularização. Observe que os diferentes parâmetros  $\hat{\gamma}$  e  $\hat{\gamma}^2$  são usados para regularizar os conjuntos de variáveis do modelo de fatores latentes e do modelo de vizinhança, respectivamente, para maior flexibilidade no processo de otimização.

### 3.7.5 Resolvendo o Modelo de Otimização

Como em todos os outros modelos de otimização discutidos neste capítulo, uma abordagem de gradiente descendente é usada para resolver o problema de otimização. Neste caso, a otimização

O modelo é bastante complexo porque contém um número relativamente grande de termos e um grande número de variáveis. No entanto, a abordagem para resolver o modelo de otimização é exatamente a mesma que no caso do modelo de fator latente da seção 3.6.4.6. Uma derivada parcial em relação a cada variável de otimização é usada para derivar a etapa de atualização. Omitimos a derivação das etapas de descida do gradiente e simplesmente as declaramos aqui em termos dos valores de erro  $e_{ij} = r_{ij} - \hat{r}_{ij}$ . As seguintes regras podem ser usadas para cada entrada observada ( $i, j$ ) na matriz de classificações:

$$\begin{aligned}
 & u_{iq} \leftarrow u_{iq} + \frac{\partial e_{ij}}{\partial u_{iq}} = u_{iq} + \hat{y}_{iq} - \hat{y}_{ij} \quad \{1 \dots k+2\} \\
 & v_{jq} \leftarrow v_{jq} + \frac{\partial e_{ij}}{\partial v_{jq}} = v_{jq} + \frac{y_{hq}}{\|v_j\|} - \hat{y}_{ij} \quad \{1 \dots k+2\} \\
 & y_{hq} \leftarrow y_{hq} + \frac{\partial e_{ij}}{\partial y_{hq}} = y_{hq} + \frac{e_{ij} \cdot v_{jq}}{\|v_j\|} - \hat{y}_{ij} \quad \{1 \dots k+2\}, \hat{y}_h \leftarrow \hat{y}_h + \hat{y}_{ij} \\
 & \text{branco} \leftarrow \text{branco} + \frac{\partial e_{ij}}{\partial \text{branco}} = \text{branco} + \frac{e_{ij} \cdot (r_{il} - \hat{y}_{il})}{\|Q_j(i)\|} \\
 & c_{lj} \leftarrow c_{lj} + \frac{\partial e_{ij}}{\partial c_{lj}} = c_{lj} + \frac{e_{ij}}{\|Q_j(i)\|} - \hat{y}_{ij} \quad \hat{y}_l \leftarrow \hat{y}_l + \hat{y}_{ij}
 \end{aligned}$$

Redefinir entradas perturbadas em colunas fixas de U, V, e Y

As três primeiras atualizações também podem ser escritas na forma vetorializada ( $k+2$ )-dimensional. Consulte a seção sobre SVD++ para uma nota de rodapé contendo essas atualizações. Repetimos o loop sobre todas as classificações observadas em  $S$  com um método de descida de gradiente estocástico. A estrutura algorítmica básica para a descida de gradiente estocástico é descrita na Figura 3.9. O valor de  $\hat{y}$  regula o tamanho do passo para variáveis associadas à porção do fator latente do modelo, enquanto  $\hat{y}^2$  regula o tamanho do passo para variáveis associadas à porção da vizinhança do modelo. As colunas fixas de  $U$ ,  $V$  e  $Y$  não devem ser atualizadas por essas regras, de acordo com as restrições do modelo de otimização. Isso é obtido na prática sempre redefinindo-as para seus valores fixos ao final de uma iteração. Além disso, essas colunas são sempre inicializadas para seus valores fixos, conforme exigido pelas restrições do modelo de otimização. Os parâmetros de regularização podem ser selecionados mantendo uma fração das entradas observadas durante o treinamento e ajustando a precisão das entradas mantidas.

Uma abordagem mais eficaz é usar o método de validação cruzada discutido no Capítulo 7. É particularmente importante usar diferentes tamanhos de passo e parâmetros de regularização para as porções de vizinhança e fator latente do modelo para evitar desempenho ruim.

### 3.7.6 Observações sobre Precisão

Foi demonstrado em [309] que o modelo combinado forneceu resultados superiores aos de cada um dos modelos individuais. Isso é resultado da capacidade do modelo combinado de se adaptar a características variáveis de diferentes porções do conjunto de dados. A ideia básica é semelhante àquela usada frequentemente em sistemas de recomendação híbridos (cf. Capítulo 6) para combinar diferentes tipos de modelos. Pode-se tentar aproximar os resultados do modelo integrado usando uma média ponderada das previsões dos dois modelos de componentes diferentes. Os pesos relativos podem ser aprendidos usando as técnicas de validação cruzada ou de retenção mencionadas anteriormente. No entanto, em comparação com o modelo médio, o modelo integrado desta seção é mais poderoso. Uma razão é que as variáveis de viés são compartilhadas pelos dois componentes, o que impede

sobreajuste das variáveis de viés às nuances específicas de cada modelo. Além disso, o uso da função de predição da Equação 3.38 regula implicitamente a importância de cada parte do modelo escolhendo automaticamente valores apropriados para cada um dos variáveis no processo de otimização. Como resultado, esse tipo de integração geralmente fornece precisão superior. No entanto, o modelo oferece apenas um desempenho ligeiramente superior ao que fornecido por SVD++, e os resultados dependem do conjunto de dados. Uma questão a ter em mente é que o modelo de vizinhança tem mais parâmetros a serem otimizados do que SVD++. Significativa vantagens não serão obtidas pelo componente de vizinhança a menos que o conjunto de dados seja suficientemente grande. Para conjuntos de dados menores, aumentar o número de parâmetros geralmente leva a sobreajuste. Nesse sentido, a escolha adequada entre modelos de fatores assimétricos, SVD puro com vieses, SVD++ e fatoração integrada à vizinhança, muitas vezes depende do tamanho do conjunto de dados em questão. Modelos mais complexos requerem conjuntos de dados maiores para evitar overfitting. Para conjuntos de dados muito pequenos, seria melhor usar modelos de fatores assimétricos. Para conjuntos de dados muito grandes conjuntos de dados, o modelo de fatoração integrada à vizinhança é o melhor. SVD++ geralmente faz melhor que SVD puro (com vieses) na maioria das configurações.

### 3.7.7 Integrando Modelos de Fatores Latentes com Modelos Arbitrários

A integração de modelos de fatores latentes com modelos baseados em vizinhança fornece informações úteis dicas sobre a integração do primeiro com outros tipos de modelos, como métodos baseados em conteúdo. Tal integração naturalmente leva à criação de sistemas de recomendação híbridos.

Em geral, os perfis de itens podem estar disponíveis na forma de descrições de produtos. Da mesma forma, Os usuários podem ter criado perfis explicitamente descrevendo seus interesses. Suponha que o perfil do usuário i seja denotado pelo vetor de palavras-chave  $C_{user}^i$  e o perfil do item j é denotado pelo vetor de palavras-chave  $C_{item}^j$ . Além disso, suponha que as avaliações observadas do usuário i sejam denotado por  $R_{user}^i$ , e as classificações observadas do item j são denotadas por  $R_{item}^j$ . Então, pode-se escrever a seguinte forma geral da função de previsão:

Aqui,  $\bar{y}$  é um fator de equilíbrio que controla a importância relativa dos dois modelos.

segundo termo, que é  $F(C_{user}^i)$ , é uma função parametrizada do usuário perfil, perfil do item, avaliações do usuário e avaliações do item. É possível otimizar os parâmetros deste funcionar em conjunto com os fatores latentes para minimizar o erro de previsão na Equação 3.39.

A integração de modelos de vizinhança e de fatores latentes pode ser vista como uma caso deste método em que a função  $F()$  é uma função de regressão linear que usa apenas R<sub>item</sub><sub>eu</sub> e ignora todos os outros argumentos. No entanto, é possível projetar um quase número infinito de variantes dessa abordagem mais ampla, variando a escolha da função  $F()$ . Também é possível ampliar o escopo de  $F()$  usando outras fontes de dados, como redes sociais dados, localização ou tempo. Na verdade, praticamente qualquer modelo de filtragem colaborativa, que é proposto na forma de uma função de predição parametrizada, pode ser integrada com o fator latente modelo. De fato, muitos métodos foram propostos na literatura de pesquisa que integram vários tipos de regressão baseada em recursos, modelagem de tópicos ou outras novas fontes de dados com modelos de fatores latentes. Por exemplo, um método de regularização social (cf. seção 11.3.8 de O Capítulo 11 integra o modelo de fator latente com informações de confiança social para melhorar previsões. Há um escopo significativo para melhorar o estado da arte em recomendador sistemas através da identificação de novas fontes de dados, cujo poder preditivo pode ser integrado com modelos de fatores latentes usando a estrutura mencionada.

## 3.8 Resumo

---

Este capítulo discute uma série de modelos para filtragem colaborativa. O problema da filtragem colaborativa pode ser visto como uma generalização do problema da classificação. Portanto, muitos dos modelos que se aplicam à classificação também se aplicam à filtragem colaborativa com alguma generalização. Uma exceção notável são os modelos de fatores latentes, que são altamente adaptados ao problema da filtragem colaborativa. Os modelos de fatores latentes usam diferentes tipos de fatoração para prever classificações. Esses diferentes tipos de fatoração diferem na natureza de suas funções objetivo e nas restrições em suas matrizes base. Além disso, eles podem ter diferentes compensações em termos de precisão, sobreajuste e interpretabilidade.

Modelos de fatores latentes representam o que há de mais moderno em filtragem colaborativa. Uma ampla variedade de modelos de fatores latentes foi proposta, com base nas escolhas da função objetivo e nas restrições de otimização. Modelos de fatores latentes também podem ser combinados com métodos de vizinhança para criar modelos integrados, que podem se beneficiar do poder tanto dos modelos de fatores latentes quanto dos métodos de vizinhança.

## 3.9 Notas Bibliográficas

---

O problema da filtragem colaborativa está intimamente relacionado ao da classificação. Inúmeros sistemas de recomendação foram propostos na literatura; estes modificam os vários modelos de classificação para realizar recomendações. A relação entre filtragem colaborativa e classificação é discutida em [82]. Os primeiros métodos baseados em associação são descritos em [524]. Vários aprimoramentos do método, que usam níveis de suporte específicos para o item em questão, são discutidos em [358, 359, 365]. Os dois primeiros métodos alavancam associações de usuários em vez de associações de itens [358, 359]. Sistemas baseados em regras de associação encontraram usos significativos em sistemas de personalização e recomendação baseados na Web [441, 552]. Métodos de regras de associação podem ser combinados com métodos de vizinhança para extrair associações localizadas [25] entre itens ou entre usuários. Associações localizadas geralmente fornecem recomendações mais refinadas do que é possível com métodos baseados em regras globais. Um método para realizar filtragem colaborativa com o uso do método de Bayes é discutido em [437].

O uso de modelos relacionais probabilísticos para filtragem colaborativa é proposto em [219].

Máquinas de vetores de suporte para sistemas de recomendação são discutidas em [638].

Redes neurais também têm sido utilizadas recentemente para filtragem colaborativa [519, 679]. A máquina de Boltzmann restrita (RBM) é uma rede neural com uma camada de entrada e uma camada oculta. Esse tipo de rede tem sido utilizado para filtragem colaborativa [519], na qual as unidades visíveis correspondem a itens, e o treinamento é realizado sobre todos os usuários em cada época.

A classificação dos itens pelos usuários resulta na ativação das unidades visíveis. Como os RBMs podem utilizar a não linearidade dentro das unidades, eles podem, às vezes, alcançar desempenho superior aos modelos de fatores latentes. Os RBMs utilizam representações fatoradas do amplo espaço de parâmetros para reduzir o sobreajuste e demonstraram ser muito precisos no concurso Netflix Prize. A ideia básica das representações fatoradas de parâmetros também tem sido utilizada em outros métodos recentes, como as máquinas de fatoração [493].

Uma discussão detalhada de vários métodos de redução de dimensionalidade pode ser encontrada em [22]. O uso de métodos de redução de dimensionalidade para filtragem baseada em vizinhança foi proposto em [525]. Os trabalhos em [24, 525], propostos independentemente, também discutem os primeiros usos de modelos de fatores latentes como métodos autônomos para recomendação e imputação de dados ausentes. O trabalho em [24] combina um algoritmo EM com modelos de fatores latentes para imputar entradas ausentes. Métodos de fatores latentes autônomos são particularmente eficazes para aplicações colaborativas.

filtragem e representam o estado da arte na literatura. Métodos para regularizar métodos de fatores latentes são discutidos por Paterek em [473]. O mesmo trabalho também introduz a noção de viés de usuário e item em modelos de fatores latentes. Um modelo de fator assimétrico é discutido neste trabalho, no qual os usuários não são explicitamente representados por fatores latentes. Neste caso, um fator do usuário é representado como uma combinação linear dos fatores implícitos dos itens que ele avaliou.

Como resultado, o número de parâmetros a serem aprendidos é reduzido. De fato, o trabalho de Paterek (relativamente subestimado) [473] introduziu quase todas as inovações básicas que foram posteriormente combinadas e refinadas de diversas maneiras [309, 311, 313] para criar métodos de última geração, como o SVD++.

Os trabalhos iniciais [133, 252, 300, 500, 569, 666] mostraram como diferentes formas de fatoração de matrizes poderiam ser usadas para recomendações. A diferença entre as várias formas de fatoração de matrizes está na natureza das funções objetivo (perda) e nas restrições sobre as matrizes fatoriais. O método em [371] propõe a noção de filtragem colaborativa de kernel, que descobre hiperplanos não lineares nos quais as classificações são distribuídas. Essa abordagem é capaz de modelar distribuições de classificações mais complexas. Esses diferentes tipos de fatoração levam a diferentes compensações em qualidade, sobreajuste e interpretabilidade. Métodos incrementais para filtragem colaborativa para fatoração de matrizes são discutidos em [96].

Muitas variações da função objetivo básica e das restrições são utilizadas em diferentes formas de fatoração matricial. Os trabalhos em [180, 500, 569, 624] exploram a fatoração de margem máxima, que está intimamente relacionada à fatoração de matriz irrestrita. A principal diferença é que um regularizador de margem máxima é utilizado com perda de dobradiça na função objetivo, em vez de utilizar a norma de Frobenius da matriz de erro para quantificar a perda. Os trabalhos em [252, 666] são formas não negativas de fatoração matricial. Uma discussão detalhada dos métodos de fatoração de matriz não negativa para dados completos pode ser encontrada em [22, 537].

O trabalho em [666] explora o método convencional de fatoração não negativa com a norma de Frobenius, enquanto o trabalho em [252, 517] explora formas probabilísticas de fatoração de matrizes. Algumas das versões probabilísticas também minimizam a norma de Frobenius, mas também otimizam a regularização simultaneamente. Métodos para combinar métodos bayesianos com métodos de fatoração de matrizes (a fim de determinar criteriosamente os parâmetros de regularização) são discutidos em [518]. A amostragem de Gibbs é usada para atingir esse objetivo. Técnicas de inicialização para métodos de fatoração de matrizes não negativas são discutidas em [331]. Após a popularização dos modelos de fatores latentes pelo concurso Netflix Prize [73], outros métodos baseados em fatoração também foram propostos para filtragem colaborativa [309, 312, 313]. Um dos primeiros modelos de fatores latentes, que funciona com dados de feedback implícito, é apresentado em [260]. A descrição do SVD++ neste livro foi emprestada de [309]. Um trabalho recente [184] impõe uma penalidade proporcional à norma de Frobenius da UV para forçar valores não observados a terem classificações mais baixas. A ideia é penalizar classificações mais altas. Essa abordagem impõe vieses mais fortes do que [309], pois assume explicitamente que as classificações não observadas têm valores mais baixos.

Além disso, as classificações em [184] precisam ser quantidades não negativas, de modo que a norma de Frobenius penalize classificações mais altas em maior grau. Alguns dos métodos de fatores latentes [309] mostram como técnicas como SVD++ podem ser combinadas com métodos de vizinhança baseados em regressão (cf. seção 3.7). Portanto, esses métodos combinam regressão linear com modelos de fatoração. Um método de fatoração de matrizes que utiliza decomposição de valores singulares é discutido em [127]. O uso de métodos de complementação de matrizes indutivas em matrizes de filtragem colaborativa com informações secundárias é discutido em [267].

Vários modelos baseados em regressão são discutidos em [72, 309, 342, 434, 620, 669]. Uma análise geral de classificadores lineares, como regressão de mínimos quadrados e máquinas de vetores de suporte (SVMs), é fornecida em [669]. Este trabalho foi uma das primeiras avaliações de métodos lineares, embora tenha sido projetado apenas para conjuntos de dados de feedback implícito, como Web

Dados de cliques ou dados de vendas, nos quais apenas preferências positivas estão disponíveis. Observou-se que a filtragem colaborativa, nesses casos, é semelhante em forma à categorização de texto. No entanto, devido ao ruído nos dados e à natureza desequilibrada da distribuição de classes, o uso direto de métodos SVM às vezes não é eficaz. Alterações na função de perda são sugeridas em [669] para fornecer resultados mais precisos. A abordagem mostra que, ao usar uma função de perda quadrática na otimização SVM, obtém-se uma forma mais semelhante à abordagem dos mínimos quadrados. O SVM modificado tem um desempenho competitivo ou superior à abordagem dos mínimos quadrados. Os métodos em [72, 309] estão intimamente associados a métodos baseados em vizinhança e são discutidos na seção 2.6 do Capítulo 2. O trabalho em [620] usa coleções de modelos lineares, que são modelados como problemas de mínimos quadrados ordinários. O uso de modelos baseados em regressão, como preditores de declive um, é discutido em [342].

Conforme discutido na seção 2.6 do Capítulo 2, os modelos de regressão podem ser usados para mostrar a conexão formal entre métodos baseados em modelos e métodos baseados em vizinhança [72, 309].

Outros métodos para combinar regressão com modelos de fatores latentes são discutidos em [13]. Os trabalhos em [321, 455] desenvolvem vários tipos de modelos lineares esparsos (SLIM) que combinam a abordagem de vizinhança com regressão e fatoração de matrizes. A abordagem SLIM é projetada principalmente para conjuntos de dados de feedback implícito.

Uma quantidade significativa de trabalho tem sido dedicada à escolha da metodologia para determinar a solução dos problemas de otimização subjacentes. Por exemplo, uma discussão sobre as compensações entre gradiente descendente e gradiente descendente estocástico é apresentada em [351], e minilotes são propostos para preencher a lacuna entre os dois. Métodos de mínimos quadrados alternados são discutidos em [268, 677]. A ideia original de mínimos quadrados alternados é proposta na fatoração de matrizes positivas de matrizes completas [460]. Métodos para gradiente descendente estocástico distribuído e em larga escala em modelos de fatores latentes são propostos em [217].

O principal trade-off entre a descida estocástica e os mínimos quadrados alternados é o trade-off entre estabilidade e eficiência. O primeiro método é mais eficiente, enquanto o último é mais estável. Foi sugerido que métodos de descida coordenada [650] podem ser eficientes, mantendo a estabilidade. Também foi demonstrado [651] que métodos não paramétricos apresentam diversas vantagens para filtragem colaborativa em larga escala com modelos de fatores latentes. Métodos para abordar problemas de inicialização a frio em modelos de fatores latentes são discutidos em [676]. A competição do Prêmio Netflix foi particularmente notável na história dos modelos de fatores latentes porque resultou em diversas lições úteis [73] sobre a implementação adequada de tais modelos.

Recentemente, modelos de fatores latentes têm sido utilizados para modelar preferências mais ricas do usuário. Por exemplo, o trabalho em [322] mostra como se pode combinar preferências globais com preferências específicas de interesse para fazer recomendações.

## 3.10 Exercícios

---

1. Implemente um preditor de classificações baseado em árvore de decisão para conjuntos de dados incompletos. Use a abordagem de redução de dimensionalidade descrita no capítulo.
2. Como você usaria um sistema de filtragem colaborativa baseado em regras no caso em que as classificações são números reais em  $[y_1, 1]$ .
3. Projete um algoritmo que combine métodos de regras de associação com agrupamento para recomendações, a fim de descobrir associações localizadas em dados unários. Qual é a vantagem dessa abordagem em relação a um método baseado em regras padrão?
4. O modelo Bayesiano ingênuo discutido neste capítulo prevê as classificações de cada item usando as outras classificações do usuário como uma condição. Projete um modelo Bayesiano que use as

Outras classificações do item como condição. Discuta as vantagens e desvantagens de cada modelo. Identifique um caso em que cada abordagem funcionaria melhor. Como você combinaria as previsões dos dois modelos?

5. Suponha que um comerciante tenha uma matriz unária contendo o comportamento de compra de vários clientes. Cada entrada na matriz contém informações sobre se um cliente comprou ou não um determinado item. Entre os usuários que ainda não compraram um item, o comerciante deseja classificar todos os usuários em ordem de propensão à compra. Mostre como usar o modelo de Bayes para atingir esse objetivo.
6. Use o modelo de Bayes da Tabela 3.1 para determinar a probabilidade de João comprar Pão no futuro. Trate os 0s na tabela como valores realmente especificados para as classificações, em vez de valores ausentes (exceto para as classificações de João para Pão e Carne). Determine a probabilidade de ele comprar carne bovina no futuro. É mais provável que João compre pão ou carne bovina no futuro?
7. Implementar o modelo Bayes ingênuo para filtragem colaborativa.
8. Execute uma SVD de classificação 2 simples da matriz da Tabela 3.2 , tratando os valores ausentes como 0. Com base no uso da SVD, quais são as classificações previstas para os valores ausentes do usuário 3? Como isso se compara aos resultados mostrados no exemplo da seção 3.6.5.4, que usa uma inicialização diferente? Como os resultados se comparam aos obtidos usando o modelo de Bayes descrito no capítulo?
9. Suponha que você receba uma matriz R que pode ser fatorada como  $R = UV^T$ , onde as colunas de U são mutuamente ortogonais e as colunas de V são mutuamente ortogonais. Mostre como fatorar R em três matrizes na forma  $Q\tilde{y}P$  de P e Q são ortonormais  $^T$ , e  $\tilde{y}$  é uma matriz diagonal não negativa.
10. Implementar o método de fatoração de matriz irrestrita com gradiente estocástico descida e atualizações em lote.
11. Discuta as mudanças necessárias no método dos mínimos quadrados alternados para fatoração de matrizes irrestrita, quando se restringe a última coluna da matriz fator-usuário a conter apenas 1s, e a penúltima coluna da matriz fator-item a conter apenas 1s. Este método é útil para incorporar vieses de usuário e item na fatoração de matrizes irrestrita.
12. Discuta como você pode aplicar o método dos mínimos quadrados alternados para projetar modelos de fatores latentes com feedback implícito.
13. Seja a matriz  $m \times k$  F, matriz  $n \times k$  V, e a matriz  $n \times k$  Y seja definida conforme discutido na parte sobre o modelo de fator assimétrico da seção 3.6.4.6. Assuma uma configuração simplificada de modelos de fator assimétrico, na qual não precisamos levar em conta os vieses do usuário e do item.
  - (a) Mostre que as atualizações estocásticas de gradiente descendente para cada entrada observada ( $i, j$ ) na matriz de classificações R são as seguintes:

$$\frac{v_{jq} \tilde{y} v_{jq} + \tilde{y} e_{ij} \cdot}{h_{li}} - \frac{\tilde{y} \tilde{y} \cdot v_{jq} \tilde{y} q \tilde{y} \{1 \dots k\}}{\|i\|}$$

$$\frac{e_{ij} \cdot v_{jq} \tilde{y}}{h_{li} \cdot v_{jq} \tilde{y}} - \frac{y_{hq} \tilde{y} h_{qj} + \tilde{y}}{\|i\|} \cdot y_{hq} \tilde{y} q \tilde{y} \{1 \dots k\}, \tilde{y} h \tilde{y} l \|i\|$$

Aqui,  $e_{ij} = r_{ij} - \hat{r}_{ij}$  é o erro de entrada observado ( $i, j$ ) e  $I_i$  é o conjunto de itens para os quais o usuário  $i$  especificou classificações. (b) Quais mudanças precisariam ser feitas nas definições de várias matrizes e nas atualizações para levar em conta os vieses do usuário e do item?

---

## Capítulo 4

# Sistemas de recomendação baseados em conteúdo

---

"A forma deve ter um conteúdo, e esse conteúdo deve estar ligado à natureza." –  
Alvar Aalto

### 4.1 Introdução

---

Os sistemas colaborativos discutidos nos capítulos anteriores utilizam as correlações nos padrões de avaliação entre os usuários para fazer recomendações. Por outro lado, esses métodos não utilizam atributos dos itens para calcular previsões. Isso pareceria um desperdício; afinal, se John gosta do filme futurista de ficção científica *O Exterminador do Futuro*, há uma grande chance de que ele goste de um filme de gênero semelhante, como *Aliens*. Nesses casos, as avaliações de outros usuários podem não ser necessárias para fazer recomendações significativas.

Sistemas baseados em conteúdo são projetados para explorar cenários em que itens podem ser descritos com conjuntos descritivos de atributos. Nesses casos, as avaliações e ações do próprio usuário em outros filmes são suficientes para descobrir recomendações significativas. Essa abordagem é particularmente útil quando o item é novo e há poucas avaliações disponíveis para ele.

Os sistemas de recomendação baseados em conteúdo tentam associar os usuários a itens semelhantes aos que eles gostaram no passado. Essa similaridade não se baseia necessariamente em correlações de classificação entre os usuários, mas sim nos atributos dos objetos curtidos pelo usuário. Ao contrário dos sistemas colaborativos, que alavancam explicitamente as classificações de outros usuários, além daquelas do usuário-alvo, os sistemas baseados em conteúdo focam amplamente nas próprias classificações do usuário-alvo e nos atributos dos itens curtidos pelo usuário. Portanto, os outros usuários têm pouco ou nenhum papel a desempenhar nos sistemas baseados em conteúdo. Em outras palavras, a metodologia baseada em conteúdo alavanca uma fonte diferente de dados para o processo de recomendação. Como veremos no Capítulo 6, muitos sistemas de recomendação alavancam o poder de ambas as fontes. Esses sistemas de recomendação são chamados de sistemas de recomendação híbridos.

No nível mais básico, os sistemas baseados em conteúdo dependem de duas fontes de dados:

1. A primeira fonte de dados é uma descrição de vários itens em termos de atributos centrados no conteúdo. Um exemplo dessa representação poderia ser a descrição textual de um item pelo fabricante.
2. A segunda fonte de dados é um perfil de usuário, gerado a partir do feedback do usuário sobre diversos itens. O feedback do usuário pode ser explícito ou implícito. O feedback explícito pode corresponder a avaliações, enquanto o feedback implícito pode corresponder a ações do usuário. As avaliações são coletadas de forma semelhante aos sistemas colaborativos.

O perfil do usuário relaciona os atributos dos vários itens aos interesses do usuário (classificações). Um exemplo básico de perfil de usuário pode ser simplesmente um conjunto de documentos de treinamento rotulados com descrições de itens, as avaliações do usuário como rótulos e um modelo de classificação ou regressão relacionando os atributos dos itens às avaliações do usuário. O perfil específico do usuário depende muito da metodologia em questão. Por exemplo, avaliações explícitas podem ser usadas em um contexto e feedback implícito em outro. Também é possível que o usuário especifique seu próprio perfil em termos de palavras-chave de interesse, e essa abordagem compartilha algumas características com sistemas de recomendação baseados em conhecimento.

Vale ressaltar que as avaliações dos outros usuários geralmente não desempenham nenhum papel em um algoritmo de recomendação baseado em conteúdo. Isso é tanto uma vantagem quanto uma desvantagem, dependendo do cenário em questão. Por um lado, em cenários de inicialização a frio, onde há pouca informação disponível sobre as avaliações de outros usuários, essa abordagem ainda pode ser usada, desde que haja informações suficientes sobre os interesses do próprio usuário. Isso, pelo menos parcialmente, alivia o problema da inicialização a frio quando o número de outros usuários no sistema de recomendação é Além disso, quando um item é novo, não é possível obter as avaliações de outros usuários para esse item. Métodos baseados em conteúdo permitem recomendações em tais cenários porque podem extrair os atributos do novo item e usá-los para fazer previsões. Por outro lado, o problema do início a frio para novos usuários não pode ser resolvido com sistemas de recomendação baseados em conteúdo. Além disso, ao não usar as avaliações de outros usuários, reduz-se a diversidade e a novidade dos itens recomendados. Em muitos casos, os itens recomendados podem ser itens óbvios para o usuário ou podem ser outros itens que o usuário já consumiu antes. Isso ocorre porque os atributos de conteúdo sempre recomendarão itens com atributos semelhantes aos que o usuário viu no passado. Um item recomendado com atributos semelhantes geralmente apresenta pouca surpresa para o usuário. Essas vantagens e desvantagens serão discutidas em uma seção posterior deste capítulo.

Sistemas baseados em conteúdo são amplamente utilizados em cenários nos quais uma quantidade significativa de informações sobre atributos está disponível. Em muitos casos, esses atributos são palavras-chave, extraídas das descrições dos produtos. De fato, a grande maioria dos sistemas baseados em conteúdo extrai atributos de texto dos objetos subjacentes. Sistemas baseados em conteúdo são, portanto, particularmente adequados para fornecer recomendações em domínios ricos em texto e não estruturados. Um exemplo clássico do uso de tais sistemas é a recomendação de páginas da Web. Por exemplo, o comportamento de navegação anterior de um usuário pode ser utilizado para criar um sistema de recomendação baseado em conteúdo. No entanto, o uso de tais sistemas não se restringe apenas ao domínio da Web. Palavras-chave de descrições de produtos são usadas para criar perfis de itens e usuários para fins de recomendações em outros ambientes de comércio eletrônico. Em outros ambientes, atributos relacionais, como fabricante, gênero e preço, podem ser usados além das palavras-chave. Esses atributos podem ser usados para criar representações estruturadas, que podem ser armazenadas em um banco de dados relacional. Nesses casos, é necessário combinar os atributos estruturados e não estruturados em uma única representação estruturada. Os princípios básicos de

No entanto, sistemas baseados em conteúdo permanecem invariáveis quanto à utilização de uma representação estruturada ou não estruturada. Isso ocorre porque a maioria dos métodos de aprendizagem no domínio estruturado possui análogos diretos no domínio não estruturado, e vice-versa. Para preservar a uniformidade na exposição, nossa discussão neste capítulo se concentrará em contextos não estruturados. No entanto, a maioria desses métodos pode ser facilmente adaptada a contextos estruturados.

Os sistemas baseados em conteúdo estão intimamente relacionados aos sistemas de recomendação baseados em conhecimento. Um resumo da relação entre os vários tipos de sistemas é fornecido na Tabela 1.2 do Capítulo 1. Assim como os sistemas baseados em conteúdo, os sistemas de recomendação baseados em conhecimento utilizam os atributos de conteúdo dos itens para fazer recomendações. A principal diferença é que os sistemas baseados em conhecimento suportam a especificação explícita dos requisitos do usuário em conjunto com interfaces interativas entre o usuário e os sistemas de recomendação. Bases de conhecimento são utilizadas em conjunto com essa interatividade para relacionar os requisitos do usuário aos itens.

Por outro lado, os sistemas baseados em conteúdo geralmente utilizam uma abordagem baseada em aprendizagem, baseada em classificações históricas. Portanto, os sistemas baseados em conhecimento proporcionam melhor controle ao usuário no processo de recomendação, enquanto os sistemas baseados em conteúdo alavancam o comportamento passado de forma mais eficaz. No entanto, essas diferenças não são tão significativas, e alguns métodos baseados em conteúdo também permitem que os usuários especifiquem explicitamente seus perfis de interesse. Vários sistemas alavancam tanto os aspectos de aprendizagem quanto os interativos dentro de uma estrutura unificada. Tais sistemas são chamados de sistemas de recomendação híbridos. Os sistemas de recomendação baseados em conhecimento são discutidos no Capítulo 5, enquanto os sistemas de recomendação híbridos são discutidos no Capítulo 6.

Este capítulo está organizado da seguinte forma. A próxima seção fornece uma visão geral dos componentes básicos de um sistema de recomendação baseado em conteúdo. Os métodos de extração e seleção de recursos são discutidos na seção 4.3. O processo de aprendizado de perfis de usuários e sua utilização para recomendações é discutido na seção 4.4. Uma comparação das principais propriedades de sistemas colaborativos e baseados em conteúdo é apresentada na seção 4.5. As conexões entre filtragem colaborativa e métodos baseados em conteúdo são exploradas na seção 4.6. Um resumo é apresentado na seção 4.7.

## 4.2 Componentes básicos de sistemas baseados em conteúdo

---

Sistemas baseados em conteúdo possuem certos componentes básicos, que permanecem invariantes em diferentes instâncias de tais sistemas. Como sistemas baseados em conteúdo trabalham com uma ampla variedade de descrições de itens e conhecimento sobre usuários, é necessário converter esses diferentes tipos de dados não estruturados em descrições padronizadas. Na maioria dos casos, é preferível converter as descrições dos itens em palavras-chave. Portanto, sistemas baseados em conteúdo operam em grande parte, mas não exclusivamente, no domínio do texto. Muitas aplicações naturais de sistemas baseados em conteúdo também são centradas em texto. Por exemplo, sistemas de recomendação de notícias são frequentemente sistemas baseados em conteúdo e também são sistemas centrados em texto. Em geral, métodos de classificação de texto e modelagem de regressão continuam sendo as ferramentas mais amplamente utilizadas para a criação de sistemas de recomendação baseados em conteúdo.

Os principais componentes dos sistemas baseados em conteúdo incluem a parte de pré-processamento (offline), a parte de aprendizagem (offline) e a parte de previsão online. As partes offline são usadas para criar um modelo resumido, que geralmente é um modelo de classificação ou regressão.

Este modelo é então utilizado para a geração online de recomendações para os usuários. Os vários componentes dos sistemas baseados em conteúdo são os seguintes:

1. Pré-processamento e extração de recursos: sistemas baseados em conteúdo são usados em uma ampla variedade de domínios, como páginas da Web, descrições de produtos, notícias, recursos musicais e assim por diante. Na maioria dos casos, os recursos são extraídos dessas várias fontes para convertê-los em

uma representação de espaço vetorial baseada em palavras-chave. Este é o primeiro passo de qualquer sistema de recomendação baseado em conteúdo e é altamente específico de domínio. No entanto, a extração adequada dos recursos mais informativos é essencial para o funcionamento eficaz de qualquer sistema de recomendação baseado em conteúdo.

2. Aprendizagem baseada em conteúdo de perfis de usuários: Como discutido anteriormente, um modelo baseado em conteúdo é específico para um determinado usuário. Portanto, um modelo específico para cada usuário é construído para prever os interesses do usuário em itens, com base em seu histórico de compra ou avaliação de itens. Para atingir esse objetivo, o feedback do usuário é aproveitado, o que pode se manifestar na forma de classificações previamente especificadas (feedback explícito) ou atividade do usuário (feedback implícito). Tais feedbacks são usados em conjunto com os atributos dos itens para construir os dados de treinamento. Um modelo de aprendizagem é construído com base nesses dados de treinamento. Essa etapa geralmente não difere muito da modelagem de classificação ou regressão, dependendo se o feedback é categórico (por exemplo, o ato binário de selecionar um item) ou numérico (por exemplo, classificações ou frequência de compra). O modelo resultante é chamado de perfil do usuário porque relaciona conceitualmente os interesses do usuário (classificações) aos atributos do item.
3. Filtragem e recomendação: Nesta etapa, o modelo aprendido na etapa anterior é usado para fazer recomendações sobre itens para usuários específicos. É importante que esta etapa seja muito eficiente, pois as previsões precisam ser realizadas em tempo real.

Nas seções seguintes, descreveremos cada uma dessas fases em detalhes. A segunda fase do aprendizado frequentemente utiliza modelos de classificação prontos para uso. O campo da classificação de dados é uma área vasta por si só, e não é objetivo deste livro discutir modelos de classificação em detalhes. Portanto, ao longo deste capítulo, assumiremos uma familiaridade prática com modelos de classificação. O objetivo será mostrar como um modelo de classificação específico pode ser usado como uma caixa-preta no sistema de recomendação e os tipos de modelos de classificação que são especialmente adequados para sistemas de recomendação baseados em conteúdo. Uma breve descrição de dois dos modelos mais comumente usados está incluída, mas esta não é de forma alguma uma descrição exaustiva. Para o leitor que não está familiarizado com modelos de classificação, as notas bibliográficas incluem indicações para vários recursos úteis.

## 4.3 Pré-processamento e Extração de Características

---

A primeira fase em todos os modelos baseados em conteúdo é extrair características discriminativas para representar os itens. Características discriminativas são aquelas que são altamente preditivas dos interesses do usuário. Esta fase depende muito da aplicação específica em questão. Por exemplo, um sistema de recomendação de páginas da web será muito diferente de um sistema de recomendação de produtos.

### 4.3.1 Extração de Recursos

Na fase de extração de características, as descrições de vários itens são extraídas. Embora seja possível usar qualquer tipo de representação, como uma representação de dados multidimensional, a abordagem mais comum é extrair palavras-chave dos dados subjacentes. Essa escolha se deve ao fato de descrições de texto não estruturadas estarem frequentemente amplamente disponíveis em diversos domínios e continuarem sendo as representações mais naturais para descrever itens. Em muitos casos, os itens podem ter vários campos descrevendo vários aspectos do item. Por exemplo, um comerciante que vende livros pode ter descrições dos livros e palavras-chave que descrevem o

conteúdo, título e autor. Em alguns casos, essas descrições podem ser convertidas em um conjunto de palavras-chave. Em outros casos, pode-se trabalhar diretamente com uma representação multidimensional (estruturada). Esta última é necessária quando os atributos contêm quantidades numéricas (por exemplo, preço) ou campos extraídos de um pequeno universo de possibilidades (por exemplo, cor).

Os vários campos precisam ser ponderados adequadamente para facilitar seu uso no processo de classificação. A ponderação de características está intimamente relacionada à seleção de características, sendo a primeira uma versão flexível desta última. Neste último caso, os atributos são incluídos ou não, dependendo de sua relevância, enquanto no primeiro caso, as características recebem pesos diferenciados, dependendo de sua importância. A questão da seleção de características será discutida em detalhes na seção 4.3.4. Como a fase de extração de características é altamente específica para cada aplicação, fornecemos ao leitor uma ideia dos tipos de características que podem precisar ser extraídas no contexto de diversas aplicações.

#### 4.3.1.1 Exemplo de recomendação de produto

Considere um site de recomendação de filmes<sup>1</sup> como o IMDb [699], que fornece recomendações personalizadas de filmes. Cada filme geralmente é associado a uma descrição, como sinopse, diretor, atores, gênero e assim por diante. Uma breve descrição de Shrek no site do IMDb é a seguinte:

"Depois que seu pântano fica cheio de criaturas mágicas, um ogro concorda em resgatar uma princesa para um senhor vilão a fim de recuperar suas terras."

Muitos outros atributos, como tags de usuário, também estão disponíveis e podem ser tratados como palavras-chave centradas no conteúdo.

No caso de Shrek, pode-se simplesmente concatenar todas as palavras-chave nos vários campos para criar uma descrição textual. O principal problema é que as diferentes palavras-chave podem não ter a mesma importância no processo de recomendação. Por exemplo, um ator específico pode ter maior importância na recomendação do que uma palavra da sinopse. Isso pode ser alcançado de duas maneiras:

1. O conhecimento específico do domínio pode ser usado para decidir a importância relativa das palavras-chave.

Por exemplo, o título do filme e o ator principal podem ter mais peso do que as palavras na descrição. Em muitos casos, esse processo é feito de forma heurística, com tentativa e erro.

2. Em muitos casos, pode ser possível aprender a importância relativa de vários recursos de forma automatizada.

Esse processo é chamado de ponderação de recursos, que está intimamente relacionado à seleção de recursos. Tanto a ponderação quanto a seleção de recursos serão descritas em uma seção posterior.

#### 4.3.1.2 Exemplo de recomendação de página da Web

Documentos web requerem técnicas especializadas de pré-processamento devido a algumas propriedades comuns de sua estrutura e à riqueza dos links dentro deles. Dois aspectos principais do pré-processamento de documentos web incluem a remoção de partes específicas dos documentos (por exemplo, tags) que não são úteis e o aproveitamento da estrutura real do documento.

Nem todos os campos em um documento da Web são igualmente importantes. Documentos HTML têm vários campos, como o título, os metadados e o corpo do documento.

---

<sup>1</sup>O método exato de recomendação usado pelo IMDb é proprietário e desconhecido do autor. A descrição aqui é apenas para fins ilustrativos.

Normalmente, algoritmos analíticos tratam esses campos com diferentes níveis de importância e, portanto, os ponderam de forma diferente. Por exemplo, o título de um documento é considerado mais importante do que o corpo e recebe um peso maior. Outro exemplo de uma parte especialmente processada de um documento da Web é o texto âncora. O texto âncora contém uma descrição da página da Web apontada por um link. Devido à sua natureza descritiva, é considerado importante, mas às vezes não é relevante para o tópico da página em si. Portanto, ele é frequentemente removido do texto do documento. Em alguns casos, quando possível, o texto âncora pode até ser adicionado ao texto do documento para o qual ele aponta. Isso ocorre porque o texto âncora geralmente é uma descrição resumida do documento para o qual ele aponta. O aprendizado da importância desses vários recursos pode ser feito por meio de métodos automatizados, conforme discutido na seção 4.3.4.

Uma página da Web pode frequentemente ser organizada em blocos de conteúdo que não estão relacionados ao assunto principal da página. Uma página da Web típica terá muitos blocos irrelevantes, como anúncios, isenções de responsabilidade ou avisos, que não são muito úteis para mineração. Foi demonstrado que a qualidade dos resultados da mineração melhora quando apenas o texto do bloco principal é usado. No entanto, a determinação (automatizada) de blocos principais a partir de coleções em escala da Web é, em si, um problema de mineração de dados interessante. Embora seja relativamente fácil decompor a página da Web em blocos, às vezes é difícil identificar o bloco principal. A maioria dos métodos automatizados para determinar os blocos principais baseia-se no fato de que um determinado site normalmente utilizará um layout semelhante para todos os seus documentos. Portanto, a estrutura do layout é aprendida a partir dos documentos do site, extraíndo árvores de tags do site. Outros blocos principais são então extraídos por meio do algoritmo de correspondência de árvores [364, 662]. Métodos de aprendizado de máquina também podem ser usados para essa tarefa. Por exemplo, o problema de rotular o bloco principal em uma página pode ser tratado como um problema de classificação. As notas bibliográficas contêm indicações de métodos para extrair o bloco principal de um documento da Web.

#### 4.3.1.3 Exemplo de recomendação musical

O Pandora Internet Radio [693] é um conhecido mecanismo de recomendação musical que associa faixas a recursos extraídos do Music Genome Project [703]. Exemplos de tais recursos de faixas podem ser "raízes de trance", "riffs de sintetizador", "harmonias tonais", "batidas de bateria diretas" e assim por diante. Os usuários podem inicialmente especificar um único exemplo de uma faixa de seu interesse para criar uma "estaçao". A partir desse único exemplo de treinamento, músicas semelhantes são tocadas para o usuário. Os usuários podem expressar suas preferências ou desgostos em relação a essas músicas.

O feedback do usuário é usado para construir um modelo mais refinado para recomendação musical. Vale ressaltar que, embora as características subjacentes sejam bastante diferentes neste caso, elas ainda podem ser tratadas como palavras-chave, e o "documento" para uma determinada música corresponde ao conjunto de palavras-chave associadas a ela. Alternativamente, é possível associar atributos específicos a essas diferentes palavras-chave, o que leva a uma representação estrutural multidimensional.

A especificação inicial de uma trilha de interesse é mais semelhante a um sistema de recomendação baseado em conhecimento do que a um sistema de recomendação baseado em conteúdo. Esses tipos de sistemas de recomendação baseados em conhecimento são chamados de sistemas de recomendação baseados em casos. No entanto, quando classificações são utilizadas para fazer recomendações, a abordagem se torna mais semelhante à de um sistema de recomendação baseado em conteúdo. Em muitos casos, o Pandora também fornece uma explicação para as recomendações em termos dos atributos dos itens.

#### 4.3.2 Representação e Limpeza de Características

Este processo é

particularmente importante quando o formato não estruturado é usado para representação. A fase de extração de características é capaz de determinar grupos de palavras a partir de descrições não estruturadas de produtos ou páginas da web. No entanto, essas representações precisam ser limpas e representadas em um formato adequado para processamento. Existem várias etapas na limpeza.

processo:

1. Remoção de stopwords: Grande parte do texto extraído de descrições livres de itens conterá muitas palavras que não são específicas do item, mas que são comuns no vocabulário inglês. Essas palavras geralmente são palavras de alta frequência.

Por exemplo, palavras como "um", "uma" e "o" não serão particularmente específicas para o item em questão. Em aplicativos de recomendação de filmes, é comum encontrar tais palavras na sinopse. Em geral, artigos, preposições, conjunções e pronomes são tratados como stopwords. Na maioria dos casos, listas padronizadas de stopwords estão disponíveis em vários idiomas.

2. Rastreamento: No rastreamento, variações da mesma palavra são consolidadas. Por exemplo, formas singulares e plurais de uma palavra ou diferentes tempos verbais da mesma palavra são consolidados. Em alguns casos, raízes comuns são extraídas de várias palavras. Por exemplo, palavras como "hoping" e "hope" são consolidadas na raiz comum "hop". É claro que a derivação pode, às vezes, ter um efeito prejudicial, pois uma palavra como "hop" tem um significado próprio diferente. Muitas ferramentas prontas [710–712] estão disponíveis para derivação.

3. Extração de frases: A ideia é detectar palavras que ocorrem juntas em documentos com frequência. Por exemplo, uma frase como "cachorro-quente" significa algo diferente de suas palavras constituintes. Dicionários definidos manualmente estão disponíveis para extração de frases, embora métodos automatizados também possam ser usados [144, 364, 400].

Após executar essas etapas, as palavras-chave são convertidas em uma representação de espaço vetorial. Cada palavra também é chamada de termo. Na representação em espaço vetorial, os documentos são representados como conjuntos de palavras, juntamente com suas frequências. Embora possa ser tentador usar a frequência bruta de ocorrência das palavras, isso geralmente não é desejável. Isso ocorre porque palavras comuns costumam ser estatisticamente menos discriminativas. Portanto, tais palavras são frequentemente desconsideradas por meio de ponderação reduzida. Isso é semelhante ao princípio das stopwords, exceto que é feito de forma suave, desconsiderando a palavra, em vez de removê-la completamente.

Como as palavras são descontadas? Isso é feito usando a noção de frequência inversa de documentos. A frequência inversa de documentos ido do i-ésimo termo é uma função decrescente do número de documentos n em que ocorre.

$$idi = \log(n/ni) \quad (4.1)$$

Aqui, o número de documentos na coleção é denotado por n.

Além disso, é preciso ter cuidado para que a ocorrência excessiva de uma única palavra na coleção não receba muita importância. Por exemplo, quando descrições de itens são coletadas de fontes não confiáveis ou plataformas abertas, como a web, elas podem conter uma quantidade significativa de spam. Para atingir esse objetivo, uma função de amortecimento f(-), como a raiz quadrada ou o logaritmo, é opcionalmente aplicada às frequências antes do cálculo de similaridade.

$$\begin{aligned} f(xi) &= \sqrt{xi} f(xi) \\ &= \log(xi) \end{aligned}$$

O amortecimento de frequência é opcional e frequentemente omitido. Omitir o processo de amortecimento equivale a definir  $f(x_i)$  como  $x_i$ . A frequência normalizada  $h(x_i)$  para a  $i$ -ésima palavra é definida pela combinação da frequência inversa do documento com a função de amortecimento:

$$h(x_i) = f(x_i) \cdot idf \quad (4.2)$$

Este modelo é popularmente conhecido como modelo tf-idf, onde tf representa o termo frequência e idf representa a frequência inversa do documento.

### 4.3.3 Coletando curtidas e descurtidas do usuário

Além do conteúdo sobre os itens, também é necessário coletar dados sobre os gostos e desgostos do usuário para o processo de recomendação. A coleta de dados é feita durante a fase offline, enquanto as recomendações são determinadas durante a fase online, quando um usuário específico está interagindo com o sistema. O usuário para quem a previsão é realizada em um determinado momento é chamado de usuário ativo. Durante a fase online, as preferências do usuário são combinadas com o conteúdo para criar as previsões. Os dados sobre os gostos e desgostos do usuário podem assumir qualquer uma das seguintes formas:

1. Classificações: neste caso, os usuários especificam classificações que indicam sua preferência pelo item. As classificações podem ser binárias, baseadas em intervalos ou ordinais. Em casos raros, as classificações podem até ter valores reais. A natureza da classificação tem um impacto significativo no modelo usado para aprender os perfis dos usuários.
2. Feedback implícito: O feedback implícito refere-se às ações do usuário, como comprar ou navegar por um item. Na maioria dos casos, apenas as preferências positivas do usuário são capturadas com o feedback implícito, mas não as preferências negativas.
3. Opiniões em texto: Em muitos casos, os usuários podem expressar suas opiniões na forma de descrições em texto. Nesses casos, classificações implícitas podem ser extraídas dessas opiniões. Esta forma de extração de classificação está relacionada ao campo de mineração de opinião e análise de sentimento. Esta área está além do escopo deste livro. Leitores interessados podem consultar [364].
4. Casos: Os usuários podem especificar exemplos (ou casos) de itens nos quais estão interessados. Tais casos podem ser usados como feedback implícito com classificadores de vizinho mais próximo ou Rocchio. No entanto, quando a recuperação de similaridade é usada em conjunto com funções de utilidade cuidadosamente projetadas, esses métodos estão mais intimamente relacionados aos sistemas de recomendação baseados em casos. Sistemas baseados em casos são uma subclasse de sistemas de recomendação baseados em conhecimento, nos quais o conhecimento de domínio é usado para descobrir itens correspondentes, em vez de algoritmos de aprendizagem (cf. seção 5.3.1 do Capítulo 5). Muitas vezes, é difícil delinear onde os sistemas de recomendação baseados em conteúdo terminam e os sistemas de recomendação baseados em conhecimento começam. Por exemplo, a Pandora Internet Radio frequentemente usa um caso inicial de um álbum de música interessante para configurar "estações de rádio" para usuários com itens musicais semelhantes. Em um estágio posterior, o feedback do usuário sobre gostos e desgostos é utilizado para refinar as recomendações. Portanto, a primeira parte da abordagem pode ser vista como um sistema baseado em conhecimento, e a segunda parte da abordagem pode ser vista como um sistema baseado em conteúdo (ou colaborativo).

Em todos os casos mencionados, as preferências ou desgostos de um usuário por um item são finalmente convertidos em uma classificação unária, binária, intervalar ou real. Essa classificação também pode ser vista como a extração de um rótulo de classe ou variável dependente, que é eventualmente aproveitada para fins de aprendizagem. propósitos.

#### 4.3.4 Seleção e Ponderação Supervisionadas de Características O objetivo da seleção e

ponderação de características é garantir que apenas as palavras mais informativas sejam retidas na representação do espaço vetorial. De fato, muitos sistemas de recomendação conhecidos [60, 476] defendem explicitamente que um limite de tamanho deve ser usado para o número de palavras-chave. Os resultados experimentais em [476], realizados em diversos domínios, sugeriram que o número de palavras extraídas deveria estar entre 50 e 300.

A ideia básica é que as palavras ruidosas frequentemente resultam em overfitting e, portanto, devem ser removidas a priori. Isso é particularmente importante, considerando o fato de que o número de documentos disponíveis para aprender um perfil de usuário específico geralmente não é muito grande. Quando o número de documentos disponíveis para aprender é pequeno, a tendência do modelo ao overfitting será maior. Portanto, é crucial reduzir o tamanho do espaço de recursos.

Há dois aspectos distintos na incorporação da informatividade de características na representação do documento. Um deles é a seleção de características, que corresponde à remoção de palavras.

A segunda é a ponderação de características, que envolve dar maior importância às palavras. Observe que a remoção de stopwords e o uso da frequência inversa de documentos são exemplos de seleção e ponderação de características, respectivamente. No entanto, essas são formas não supervisionadas de seleção e ponderação de características, nas quais o feedback do usuário não recebe importância. Nesta seção, estudaremos métodos supervisionados para seleção de características, que levam em consideração as avaliações do usuário para avaliar a informatividade das características. A maioria desses métodos avalia a sensibilidade da variável dependente a uma característica para avaliar sua informatividade.

As medidas para calcular a informatividade de características podem ser usadas para realizar uma seleção rigorosa de características ou para ponderar heuristicamente as características com uma função da quantificação computada da informatividade. As medidas usadas para a informatividade de características também são diferentes, dependendo se a avaliação do usuário é tratada como um valor numérico ou categórico.

Por exemplo, no contexto de classificações binárias (ou classificações com um pequeno número de valores discretos), faz sentido usar representações categóricas em vez de numéricas. Também descreveremos alguns métodos comumente usados para ponderação de características. Na maioria das descrições a seguir, assumiremos uma representação não estruturada (textual), embora os métodos possam ser facilmente generalizados para representações estruturadas (multidimensionais).

Isso ocorre porque a representação vetorial do texto pode ser vista como um caso especial da representação multidimensional. As notas bibliográficas contêm indicações para mais detalhes sobre métodos de seleção de recursos.

##### 4.3.4.1 Índice de Gini

O índice de Gini é uma das medidas mais comumente usadas para seleção de características. É uma medida simples e intuitiva, de fácil compreensão. O índice de Gini é inherentemente adequado para classificações binárias, classificações ordinais ou classificações distribuídas em um pequeno número de intervalos. Este último caso pode, às vezes, ser obtido discretizando as classificações. A ordenação entre as classificações é ignorada e cada valor possível da classificação é tratado como uma instância de um valor de atributo categórico. Isso pode parecer uma desvantagem, pois perde informações sobre a ordenação relativa das classificações. No entanto, na prática, o número de classificações possíveis geralmente é pequeno e, portanto, não se perde precisão significativa.

Seja  $t$  o número total de valores possíveis da classificação. Entre documentos que contêm uma determinada palavra  $w$ , seja  $p_1(w) \dots p_t(w)$  a fração de itens classificados em cada um desses  $t$  valores possíveis. Então, o índice de Gini da palavra  $w$  é definido da seguinte forma:

$$\text{Gini}(w)=1 - \sum_{i=1}^t p_i(w)^2 \quad (4.3)$$

O valor de Gini( $w$ ) sempre se encontra no intervalo  $(0, 1-1/t)$ , com valores menores indicando maior poder discriminativo. Por exemplo, quando a presença da palavra  $w$  sempre resulta na classificação do documento com o  $j$ -ésimo valor de classificação possível (ou seja,  $p_j(w) = 1$ ), essa palavra é muito discriminativa para previsões de classificação. Da mesma forma, o valor do índice de Gini nesse caso é  $1-12 = 0$ . Quando cada valor de  $p_j(w)$  assume o mesmo valor de  $1/t$ ,  $i = 1(1/t) = 1 - 1/t$ .

O índice de Gini atinge seu valor máximo de  $1 - \frac{t}{t}$

#### 4.3.4.2 Entropia

A entropia é muito semelhante, em princípio, ao índice de Gini, exceto pelo fato de que os princípios da teoria da informação são utilizados para projetar a medida. Como no caso anterior, seja  $t$  o número total de valores possíveis da classificação e  $p_1(w) \dots p_t(w)$  a fração de documentos contendo uma determinada palavra  $w$ , que são classificados em cada um desses  $t$  valores possíveis. Então, a entropia da palavra  $w$  é definida da seguinte forma:

$$\text{Entropia}(w) = -\sum_{i=1}^t p_i(w) \log(p_i(w)) \quad (4.4)$$

O valor de Entropy( $w$ ) sempre se encontra no intervalo  $(0, 1)$ , sendo valores menores mais indicativos de poder discriminativo. É fácil perceber que a entropia tem características semelhantes ao índice de Gini. De fato, essas duas medidas frequentemente produzem resultados muito semelhantes, embora tenham interpretações probabilísticas diferentes. O índice de Gini é mais fácil de entender, enquanto as medidas de entropia são mais firmemente fundamentadas em princípios matemáticos da teoria da informação.

#### 4.3.4.3 Estatística $\hat{\gamma}^2$

A estatística  $\hat{\gamma}^2$  pode ser calculada tratando a coocorrência entre a palavra e a classe como uma tabela de contingência. Por exemplo, considere um cenário em que estamos tentando determinar se uma palavra específica é relevante para os interesses de compra de um usuário. Suponha que o usuário tenha comprado cerca de 10% dos itens da coleção e que a palavra "w" ocorra em cerca de 20% das descrições. Suponha que o número total de itens (e documentos correspondentes) na coleção seja 1.000. Então, o número esperado de ocorrências de cada combinação possível de ocorrência de palavra e contingência de classe é o seguinte:

	O termo ocorre na descrição	O termo não ocorre
Item comprado pelo usuário	$1000 \cdot 0,1 \cdot 0,2 = 20$	$1000 \cdot 0,1 \cdot 0,8 = 80$
O usuário não comprou o item	$1000 \cdot 0,9 \cdot 0,2 = 180$	$1000 \cdot 0,9 \cdot 0,8 = 720$

Os valores esperados mencionados acima são calculados sob a suposição de que a ocorrência do termo na descrição e o interesse do usuário no item correspondente são independentes. Se essas duas grandezas forem independentes, então claramente o termo será irrelevante para o processo de aprendizagem. No entanto, na prática, o item pode estar altamente relacionado ao item em questão. Por exemplo, considere um cenário em que a tabela de contingência se desvia dos valores esperados e é muito provável que o usuário compre o item que contém o termo. Nesse caso, a tabela de contingência pode ter a seguinte aparência:

	O termo ocorre na descrição	O termo não ocorre
Item comprado pelo usuário	O1 = 60	O2 = 40
O usuário não comprou o item	O3 = 140	O4 = 760

A estatística  $\chi^2$  mede o desvio normalizado entre os valores observados e esperados nas várias células da tabela de contingência. Neste caso, a tabela de contingência contém  $p = 2 \times 2 = 4$  células. Seja  $O_i$  o valor observado da  $i$ -ésima célula e  $E_i$  o valor esperado valor da  $i$ -ésima célula. Então, a estatística  $\chi^2$  é calculada da seguinte forma:

$$\chi^2 = \sum_{i=1}^p \frac{(O_i - E_i)^2}{E_i} \quad (4.5)$$

Portanto, no exemplo particular desta tabela, a estatística  $\chi^2$  é avaliada como o seguinte:

$$\begin{aligned} \chi^2 &= \frac{(60 - 20)^2}{20} + \frac{(40 - 80)^2}{80} + \frac{(140 - 180)^2}{180} + \frac{(760 - 720)^2}{720} \\ &= 80 + 20 + 8,89 + 2,22 \\ &= 111,11 \end{aligned}$$

Também é possível calcular a estatística  $\chi^2$  como uma função dos valores observados no tabela de contingência sem calcular explicitamente os valores esperados. Isso é possível porque os valores esperados são funções dos valores observados agregados em linhas e colunas. Uma fórmula aritmética simples para calcular a estatística  $\chi^2$  em uma tabela de contingência  $2 \times 2$  é como segue (ver Exercício 8):

$$\chi^2 = \frac{(O_1 + O_2 + O_3 + O_4) \cdot (O_1 O_4 - O_2 O_3)^2}{(O_1 + O_2) \cdot (O_3 + O_4) \cdot (O_1 + O_3) \cdot (O_2 + O_4)} \quad (4.6)$$

Aqui,  $O_1 \dots O_4$  são as frequências observadas de acordo com a tabela acima. É fácil verifique se esta fórmula produz a mesma estatística  $\chi^2$  de 111,11. Observe que o teste  $\chi^2$  pode também pode ser interpretado em termos do nível probabilístico de significância com o uso de um  $\chi^2$  distribuição. No entanto, para fins práticos, é suficiente saber que valores maiores de a estatística  $\chi^2$  indica que um termo e item específico estão relacionados em maior grau. Note que se os valores observados forem exatamente iguais aos valores esperados, isso implica que o termo correspondente é irrelevante para o item em questão. Nesse caso, a estatística  $\chi^2$  será avaliado para seu menor valor possível de 0. Portanto, os principais recursos k com o maior As estatísticas  $\chi^2$  são mantidas.

#### 4.3.4.4 Desvio Normalizado

O problema com a maioria das medidas acima mencionadas é que elas perdem informações sobre a ordenação relativa das classificações. Para os casos em que as classificações têm alta granularidade, o desvio normalizado é uma medida apropriada.

Seja  $\bar{\chi}^2$  a variância das classificações em todos os documentos. Além disso, seja  $\bar{\chi}^2(w)$  a classificação média de todos os documentos que contêm a palavra  $w$ , e  $\bar{\chi}^2(w)$  seja a classificação média de todos os documentos que não contêm a palavra  $w$ . Então, o desvio normalizado do A palavra  $w$  é definida da seguinte forma:

$$\text{Desvio}(w) = \frac{|\bar{\chi}^2(w) - \bar{\chi}^2|}{\bar{\chi}^2} \quad (4.7)$$

Valores maiores de  $\text{Dev}(w)$  são indicativos de palavras mais discriminatórias.

A quantificação mencionada baseia-se na distribuição relativa das classificações de documentos que contêm uma palavra específica em relação à distribuição de classificações de todos os documentos. Tal abordagem é particularmente adequada quando as classificações são tratadas como quantidades numéricas. Uma medida relacionada é o índice de discriminação de Fisher, que calcula a razão entre a separação interclasse e a separação intraclasse no espaço de características (em vez de na dimensão das classificações). Essa medida é descrita em detalhes em [22]. O índice discriminante de Fisher, no entanto, é mais adequado para variáveis dependentes categóricas do que para variáveis dependentes numéricas, como classificações.

#### 4.3.4.5 Ponderação de recursos

A ponderação de características pode ser vista como uma versão flexível da seleção de características. Na seção anterior sobre representação de características neste capítulo, já foi discutido como medidas como a frequência inversa de documentos podem ser usadas para ponderar documentos. No entanto, a frequência inversa de documentos é uma medida não supervisionada que não depende das preferências ou aversões do usuário. Uma medida supervisionada também pode ser usada para ponderar ainda mais a representação do espaço vetorial, a fim de atribuir importância diferencial a diferentes palavras. Por exemplo, em um aplicativo de recomendação de filmes, palavras-chave que descrevem um gênero de filme ou o nome de um ator são mais importantes do que palavras selecionadas da sinopse do filme. Por outro lado, as palavras na sinopse também são, de certa forma, indicativas de gostos. Portanto, elas também não podem ser excluídas.

A ponderação de características é uma abordagem mais refinada para discriminar entre várias palavras, utilizando um peso em vez de uma decisão binária rígida. A abordagem mais simples para a ponderação de características é tomar qualquer uma das medidas de seleção de características e usá-las para derivar os pesos. Por exemplo, o inverso do índice de Gini ou da entropia pode ser usado. Em muitos casos, uma função heurística pode ser aplicada adicionalmente à medida de seleção para controlar a sensibilidade do processo de ponderação. Por exemplo, considere a seguinte função de ponderação  $g(w)$  para a palavra  $w$ , onde  $a$  é um parâmetro maior que 1.

$$g(w) = a \cdot \bar{G}ini(w) \quad (4.8)$$

O peso resultante  $g(w)$  sempre estará no intervalo  $(a \cdot \bar{G}ini(w), a)$ . Variando o valor de  $a$ , a sensibilidade do processo de ponderação pode ser controlada. Valores menores de  $a$  resultarão em maior sensibilidade. O peso de cada palavra  $w$  na representação do espaço vetorial é então multiplicado por  $g(w)$ . Funções de ponderação semelhantes podem ser definidas em relação à entropia e ao desvio normalizado. O processo de seleção de uma ponderação de características apropriada é um processo altamente heurístico que varia significativamente de acordo com a aplicação em questão.

O valor de  $a$  pode ser visto como um parâmetro da função de ponderação. Também é possível aprender os parâmetros ótimos dessa função usando técnicas de validação cruzada. Essas técnicas são discutidas no Capítulo 7.

## 4.4 Aprendizagem de perfis de usuários e filtragem

A aprendizagem de perfis de usuários está intimamente relacionada ao problema de classificação e modelagem de regressão. Quando as avaliações são tratadas como valores discretos (por exemplo, "polegar para cima" ou "polegar para baixo"), o problema é semelhante ao da classificação de texto. Por outro lado, quando as avaliações são tratadas como um conjunto de entidades numéricas, o problema é semelhante ao da modelagem de regressão. Além disso, o problema de aprendizagem pode ser proposto em domínios estruturados e não estruturados. Para homogeneidade na apresentação, assumiremos que as descrições de

Os itens estão na forma de documentos. No entanto, a abordagem pode ser facilmente generalizada para qualquer tipo de dado multidimensional, pois o texto é um tipo especial de dado multidimensional.

Em cada caso, assumimos que temos um conjunto de documentos de treinamento, que são rotulados por um usuário específico. Esse usuário também é chamado de usuário ativo quando obtém uma recomendação do sistema. Os documentos de treinamento correspondem às descrições dos itens, que são extraídos nas fases de pré-processamento e seleção de recursos. Além disso, os dados de treinamento contêm as classificações atribuídas pelo usuário ativo a esses documentos. Esses documentos são usados para construir um modelo de treinamento. Observe que os rótulos atribuídos por outros usuários (além do usuário ativo) não são usados no processo de treinamento. Portanto, os modelos de treinamento são específicos para usuários específicos e não podem ser usados para usuários escolhidos arbitrariamente. Isso é diferente da filtragem colaborativa tradicional, na qual métodos como a fatoração de matrizes constroem um único modelo para todos os usuários. O modelo de treinamento para um usuário específico representa o perfil do usuário.

Os rótulos nos documentos correspondem às classificações numéricas, binárias ou unárias. Suponha que o  $i$ -ésimo documento em DL tenha uma classificação denotada por  $c_i$ . Também temos um conjunto DU de documentos de teste, que não são rotulados. Observe que tanto DL quanto DU são específicos para um usuário (ativo) específico. Os documentos de teste podem corresponder a descrições de itens que podem ser potencialmente recomendados ao usuário, mas que ainda não foram comprados ou avaliados por ele. Em domínios como recomendação de notícias, os documentos em DU podem corresponder a documentos da Web candidatos para recomendação ao usuário ativo. A definição precisa de DU depende do domínio em questão, mas os documentos individuais em DU são extraídos de maneira semelhante aos em DL. O modelo de treinamento em DL é usado para fazer recomendações de DU para o usuário ativo. Como no caso da filtragem colaborativa, o modelo pode ser usado para fornecer um valor previsto da classificação ou uma lista classificada das principais recomendações.

É imediatamente evidente que este problema é semelhante ao da modelagem de classificação e regressão no domínio textual. O leitor pode consultar uma pesquisa recente [21] para uma discussão detalhada de muitas dessas técnicas. A seguir, discutiremos alguns dos métodos de aprendizagem comuns.

#### 4.4.1 Classificação do vizinho mais próximo

O classificador vizinho mais próximo é uma das técnicas de classificação mais simples e pode ser implementado de forma relativamente direta. O primeiro passo é definir uma função de similaridade, que é usada no classificador vizinho mais próximo. A função de similaridade mais comumente usada é a função cosseno. Sejam  $X = (x_1 \dots x_d)$  e  $Y = (y_1 \dots y_d)$  um par de documentos, em que as frequências normalizadas da  $i$ -ésima palavra são dadas por  $x_i$  e  $y_i$ , respectivamente, nos dois documentos. Observe que essas frequências são normalizadas ou ponderadas com o uso da ponderação tf-idf não supervisionada ou dos métodos supervisionados discutidos na seção anterior. Em seguida, a medida do cosseno é definida usando essas frequências normalizadas da seguinte forma:

$$\text{Cosseno}(X, Y) = \frac{\sum_{i=1}^d x_i y_i}{\sqrt{\sum_{i=1}^d x_i^2} \sqrt{\sum_{i=1}^d y_i^2}} \quad (4.9)$$

A similaridade de cosseno é frequentemente utilizada no domínio textual devido à sua capacidade de se ajustar aos diferentes comprimentos dos documentos subjacentes. Quando essa abordagem é utilizada para outros tipos de dados estruturados e multidimensionais, outras funções de similaridade/distância, como a distância euclidiana e a distância de Manhattan, são utilizadas. Para dados relacionais com atributos categóricos, diversas medidas de similaridade baseadas em correspondência estão disponíveis [22].

Esta função de similaridade é útil para fazer previsões para itens (documentos) nos quais a preferência do usuário é desconhecida. Para cada documento em DU, seus  $k$  vizinhos mais próximos em DL são determinados usando a função de similaridade de coseno. O valor médio da classificação para os  $k$  vizinhos de cada item em DU é determinado. Este valor médio é a classificação prevista para o item correspondente em DU. Um aprimoramento heurístico adicional é que se pode ponderar cada classificação com o valor de similaridade. Nos casos em que as classificações são tratadas como valores categóricos, o número de votos para cada valor da classificação é determinado e o valor da classificação com a maior frequência é previsto. Os documentos em DU são então classificados com base no valor previsto da classificação e os principais itens são recomendados ao usuário.

O principal desafio com o uso dessa abordagem é sua alta complexidade computacional. Observe que o vizinho mais próximo de cada documento em DU precisa ser determinado, e o tempo necessário para cada determinação de vizinho mais próximo é linear ao tamanho de DL. Portanto, a complexidade computacional é igual a  $|DL| \times |DU|$ . Uma maneira de tornar a abordagem mais rápida é usar agrupamento para reduzir o número de documentos de treinamento em DL. Para cada valor distinto da classificação, o subconjunto correspondente de documentos em DL é agrupado em grupos  $|DL|$ . Portanto, se houver  $s$  valores distintos das classificações, o número total de grupos é  $p \cdot s$ .

Normalmente, um agrupamento rápido baseado em centróide (ou seja, k-means) é usado para criar cada grupo de  $p$  clusters. Observe que o número de grupos  $p \cdot s$  é significativamente menor que o número de documentos de treinamento. Nesses casos, cada grupo é convertido em um documento maior correspondente à concatenação<sup>2</sup> dos documentos naquele grupo. A representação em espaço vetorial desse documento maior pode ser extraída somando as frequências de palavras de seus constituintes. O rótulo de classificação correspondente associado ao documento é igual à classificação dos documentos constituintes. Para cada documento de destino T, os documentos  $k < p$  mais próximos são encontrados a partir deste conjunto recém-criado de documentos  $p$ . A classificação média deste conjunto de documentos  $k$  é retornada como o rótulo para o destino. Como no caso anterior, a classificação é prevista para cada item em DU e os itens com melhor classificação são retornados ao usuário ativo. Essa abordagem acelera o processo de classificação, pois é necessário calcular a similaridade entre o documento de destino e um número relativamente pequeno de documentos agregados. Embora essa abordagem incorra em uma sobrecarga adicional de pré-processamento de agrupamento, essa sobrecarga é geralmente pequena em comparação com a economia no momento da recomendação, quando os tamanhos de DL e DU são grandes.

Um caso especial dessa abordagem baseada em agrupamento é aquele em que todos os documentos pertencentes a um determinado valor de classificação são agregados em um único grupo. Assim, o valor de  $p$  é definido como 1. A representação em espaço vetorial do vetor resultante de cada grupo também é chamada de vetor protótipo. Para um documento de teste, a classificação do documento mais próximo é relatada como relevante para o alvo. Essa abordagem está intimamente relacionada à classificação de Rocchio, que também permite a noção de feedback de relevância do usuário ativo.

O método Rocchio foi originalmente desenvolvido para classes binárias, que, no nosso caso, se traduzem em classificações binárias. As notas bibliográficas contêm referências ao método Rocchio.

#### 4.4.2 Conexões com sistemas de recomendação baseados em casos

Os métodos de vizinho mais próximo estão conectados a sistemas de recomendação baseados em conhecimento em geral e a sistemas de recomendação baseados em casos em particular. Sistemas de recomendação baseados em conhecimento são discutidos em detalhes no Capítulo 5. A principal diferença é que, nos sistemas de recomendação baseados em casos, o usuário especifica interativamente um único exemplo de interesse, e os vizinhos mais próximos desse exemplo são recuperados como possíveis itens de interesse para o

---

<sup>2</sup>Para dados estruturados, o centróide do grupo pode ser usado.

Além disso, uma quantidade significativa de conhecimento de domínio é utilizada no projeto da função de similaridade, pois apenas um único exemplo está disponível. Este único exemplo pode ser mais apropriadamente visto como um requisito do usuário em vez de uma classificação histórica, porque é especificado interativamente. Em sistemas baseados em conhecimento, há menos ênfase no uso de dados históricos ou classificações. Assim como o método Rocchio, tais métodos também são interativos, embora a interatividade é muito mais sofisticada em sistemas baseados em casos.

#### 4.4.3 Classificador de Bayes

O classificador de Bayes é discutido na seção 3.4 do Capítulo 3, em filtragem colaborativa. No entanto, a discussão no Capítulo 3 é um uso não padronizado do modelo de Bayes, no qual o As entradas ausentes são previstas a partir das especificadas. No contexto de sistemas de recomendação baseados em conteúdo, o problema se traduz em um uso mais convencional do modelo de Bayes. para classificação de texto. Portanto, revisitaremos o modelo de Bayes no contexto da classificação de texto classificação.

Neste caso, temos um conjunto DL contendo os documentos de treinamento e um conjunto DU contendo os documentos de teste. Para facilitar a discussão, assumiremos que os rótulos são binários, nos quais os usuários especificam uma classificação de gosto ou desgosto como +1 ou -1, respectivamente, para cada treinamento documento em DL. No entanto, é relativamente fácil generalizar este classificador para o caso em que as classificações assumem mais de dois valores.

Como antes, suponha que a classificação do i-ésimo documento em DL seja denotada por  $c_i \in \{y_1, 1\}$ . Portanto, este conjunto rotulado representa o perfil do usuário. Existem dois modelos comumente utilizados em dados de texto, que correspondem aos modelos de Bernoulli e multinomial. respectivamente. A seguir, discutiremos apenas o modelo de Bernoulli. O modelo multinomial o modelo é discutido em detalhes em [22].

No modelo de Bernoulli, as frequências das palavras são ignoradas, e apenas a presença ou ausência da palavra no documento é considerada. Portanto, cada documento é tratado como um vetor binário de d palavras contendo apenas valores de 0 e 1. Considere um documento de destino  $X \in DU$  que pode corresponder à descrição de um item. Suponha que d características binárias em  $X$  são denotadas por  $(x_1 \dots x_d)$ . Informalmente, gostaríamos de determinar  $P(\text{Usuário ativo gosta de } X | x_1 \dots x_d)$ . Aqui, cada  $x_i$  é um valor de 0-1, correspondendo a se ou não a i-ésima palavra está presente no documento  $X$ . Então, se a classe (classificação binária) de  $X$  for denotado por  $c(X)$ , isso é equivalente a determinar o valor de  $P(c(X)=1 | x_1 \dots x_d)$ . Por determinando  $P(c(X)=1 | x_1 \dots x_d)$  e  $P(c(X) = 0 | x_1 \dots x_d)$  e selecionando o maior dos dois, pode-se determinar se o usuário ativo gosta ou não de  $X$ . Essas expressões podem ser avaliada usando a regra de Bayes e então aplicando uma suposição ingênua como a seguir:

$$\begin{aligned} P(c(X)=1 | x_1 \dots x_d) &= \frac{P(c(X)=1) \cdot P(x_1 \dots x_d | c(X)=1)}{P(x_1 \dots x_d)} \\ &\hat{=} \frac{P(c(X)=1) \cdot \prod_{i=1}^d P(x_i | c(X)=1)}{P(x_1 \dots x_d)} \quad [\text{Suposição ingênua}] \end{aligned}$$

A suposição ingênua afirma que as ocorrências de palavras em documentos são condicionalmente eventos independentes (em uma classe específica) e, portanto, pode-se substituir  $P(x_1 \dots x_d | c(X)=1)$  com  $\prod_{i=1}^d P(x_i | c(X)=1)$ . Além disso, a constante de proporcionalidade é usada no primeiro relacionamento porque o denominador é independente da classe. Portanto, o denominador não desempenha nenhum papel na decisão entre a ordem relativa das classes.

O denominador, no entanto, desempenha um papel na classificação da propensão de diferentes itens (documentos) a serem apreciados pelo usuário. Isso é relevante para o problema de classificação de itens para um usuário específico, na ordem de  $P(c(X)=1|x_1 \dots x_d)$ .

Nos casos em que tal classificação dos itens é necessária, a constante de proporcionalidade deixa de ser irrelevante. Isso é particularmente comum em aplicações de recomendação, nas quais não basta determinar as probabilidades relativas de itens pertencentes a diferentes valores de classificação, mas sim classificá-los uns em relação aos outros. Nesses casos, a constante de proporcionalidade precisa ser determinada. Suponha que a constante de proporcionalidade na relação acima seja denotada por  $K$ . A constante de proporcionalidade  $K$  pode ser obtida considerando o fato de que a soma das probabilidades de todas as instanciações possíveis de  $c(X)$  deve ser sempre 1. Portanto, temos:

$$K \cdot P(\underline{c(X)} = 1) \cdot \sum_{i=1}^d P(x_i | \underline{c(X)} = 1) + P(\underline{c(X)} = \bar{y}) \cdot \sum_{i=1}^d P(x_i | \underline{c(X)} = \bar{y}) = 1$$

Portanto, podemos derivar o seguinte valor para  $K$ :

$$K = \frac{1}{P(\underline{c(X)} = 1) \cdot \sum_{i=1}^d P(x_i | \underline{c(X)} = 1) + P(\underline{c(X)} = \bar{y}) \cdot \sum_{i=1}^d P(x_i | \underline{c(X)} = \bar{y})}$$

Essa abordagem é usada para determinar a probabilidade de um usuário gostar de cada item possível em DU. Os itens em DU são então classificados de acordo com essa probabilidade e apresentados ao usuário. Esses métodos são particularmente adequados para classificações binárias. Existem outras maneiras de usar a probabilidade para estimar o valor previsto das classificações e classificar os itens ao lidar com classificações que não são necessariamente binárias. Esses métodos são discutidos em detalhes na seção 3.4 do Capítulo 3.

#### 4.4.3.1 Estimativa de probabilidades intermediárias

O método de Bayes requer o cálculo de probabilidades intermediárias, como  $P(x_i | \underline{c(X)} = 1)$ . Até o momento, ainda não discutimos como essas probabilidades podem ser estimadas de forma orientada por dados. A principal utilidade da regra de Bayes mencionada é que ela expressa as probabilidades de previsão em termos de outras probabilidades [por exemplo,  $P(x_i | \underline{c(X)} = 1)$ ] que podem ser estimadas mais facilmente de forma orientada por dados. Reproduzimos a condição de Bayes acima:

$$\begin{aligned} P(\underline{c(X)} = 1 | x_1 \dots x_d) &= P(\underline{c(X)} = 1) \cdot \frac{\prod_{i=1}^d P(x_i | \underline{c(X)} = 1)}{\prod_{i=1}^d P(x_i | \underline{c(X)} = \bar{y})} \\ P(\underline{c(X)} = \bar{y} | x_1 \dots x_d) &= P(\underline{c(X)} = \bar{y}) \cdot \frac{\prod_{i=1}^d P(x_i | \underline{c(X)} = \bar{y})}{\prod_{i=1}^d P(x_i | \underline{c(X)} = 1)} \end{aligned}$$

Para calcular as probabilidades de Bayes, precisamos estimar as probabilidades no lado direito das equações acima. Estas incluem as probabilidades de classe anteriores  $P(\underline{c(X)} = 1)$  e  $P(\underline{c(X)} = \bar{y})$ . Além disso, as probabilidades condicionais por características, como  $P(x_i | \underline{c(X)} = 1)$  e  $P(x_i | \underline{c(X)} = \bar{y})$ , precisam ser estimadas. A probabilidade  $P(\underline{c(X)} = 1)$  pode ser estimada como a fração de exemplos de treinamento positivos  $D_+$  nos dados rotulados DL. Para reduzir o sobreajuste, a suavização laplaciana é realizada adicionando valores proporcionais a um pequeno parâmetro  $\hat{\gamma} > 0$  ao numerador e denominador.

$$P(\underline{c(X)} = 1) = \frac{|D_+| + \hat{\gamma}}{|DL| + 2 \cdot \hat{\gamma}} \quad (4.10)$$

Tabela 4.1: Ilustração do método de Bayes para um sistema baseado em conteúdo

Palavra-chave	Bateria	Guitarra	Batida	Orquestra	Sinfônica	Clássica	Curtir ou		
Song-Id	Não gosto	gostei							
1	1	1	1	0	0	0	0	Não gosto	
2	1	1	0	0	0	0	1	Não gosto	
	0		1	0	0	0	0	Não gosto	
3 4	0	1 0	0	1	1	1	1	Como	
5	0	1	0	1	0	0	1	Como	
6	0	0	0	1	1	1	0	Como	
Teste-1	0	0	0	1	0	0	0	?	
Teste-2	1	0	1	0	0	0	0	?	

O valor de  $P(c(X) = \hat{y}1)$  é estimado de forma exatamente semelhante. Além disso, a probabilidade de característica condicional  $P(x_i|c(X) = 1)$  é estimada como a fração das instâncias em a classe positiva para a qual a  $i$ -ésima característica assume o valor de  $x_i$ . Seja  $q+(x_i)$  representado o número de instâncias na classe positiva que assumem o valor de  $x_i \in \{0, 1\}$  para o com recurso. Então, podemos usar um parâmetro de suavização Laplaciano  $\hat{y} > 0$  para estimar o probabilidade da seguinte forma:

$$P(x_i|c(X) = 1) = \frac{q+(x_i) + \hat{y}}{|D_{treino}| + 2 \cdot \hat{y}} \quad (4.11)$$

Uma abordagem semelhante pode ser usada para estimar  $P(x_i|c(X) = \hat{y}1)$ . Observe que o Laplaciano A suavização é útil em casos onde há poucos dados de treinamento disponíveis. Em casos extremos, onde  $D_{treino}$  estiver vazio, a probabilidade  $P(x_i|c(X) = 1)$  seria (apropriadamente) estimada como 0,5 como uma espécie de crença prévia. Sem suavização, tal estimativa seria indeterminada, porque tanto o numerador quanto o denominador da razão seriam 0. Suavização laplaciana, como muitos métodos de regularização, pode ser interpretado em termos da maior importância de crenças anteriores quando a quantidade de dados de treinamento é limitada. Embora tenhamos apresentado a estimativa acima mencionada para o caso de classificações binárias, é relativamente fácil generalizar a estimativa quando há  $k$  valores distintos das classificações. Um tipo semelhante de estimativa é discutido no contexto da filtragem colaborativa na seção 3.4 do Capítulo 3.

#### 4.4.3.2 Exemplo de Modelo Bayesiano

Fornecemos um exemplo do uso do modelo de Bayes para um conjunto de 6 exemplos de treinamento e dois exemplos de teste. Na Tabela 4.1, as colunas correspondem às características que representam propriedades de várias músicas. O gosto ou desgosto do usuário é ilustrado na última coluna da tabela.

Portanto, a coluna final pode ser vista como a classificação. As primeiras 6 linhas correspondem à exemplos de treinamento, que correspondem ao perfil do usuário. O par final de linhas corresponde para duas faixas musicais candidatas que precisam ser classificadas para o usuário específico em questão. No jargão do aprendizado de máquina, essas linhas também são chamadas de instâncias de teste. Observe que a coluna final (variável dependente) é especificada apenas para as linhas de treinamento porque o usuário As classificações de "curtir" ou "não curtir" não são conhecidas para as linhas de teste. Esses valores precisam ser previstos.

Ao examinar as características da Tabela 4.1, torna-se imediatamente evidente que o primeiro três características (colunas) podem ocorrer frequentemente em muitos gêneros musicais populares, como rock música, enquanto as três últimas características ocorrem tipicamente na música clássica. O perfil do usuário, representado pela Tabela 4.1 parece sugerir claramente uma preferência pela música clássica em detrimento do rock

música. Da mesma forma, entre os exemplos de teste, apenas o primeiro dos dois parece corresponder aos interesses do usuário. Vamos examinar como a abordagem de Bayes consegue derivar esse fato de forma orientada por dados. Para facilitar a computação, assumiremos que a suavização laplaciana não é utilizada, embora seja importante usar tais métodos de suavização em aplicações reais.

Usando o modelo de Bayes, podemos derivar as probabilidades condicionais para curtidas e desgostos com base nas características observadas nos exemplos de teste:

$$\begin{aligned}
 & P(\text{Curtir}|\text{Teste-1}) \approx 0,5 \quad P(\text{Como}|x_i) \\
 & \qquad \qquad \qquad i=1 \\
 & 3 = (0,5) \cdot \frac{1}{4} - \frac{2}{2} \cdot \frac{3}{4} \cdot \frac{3}{3} \cdot \frac{1}{4} \cdot \frac{1}{3} \\
 & = \frac{3}{128} \\
 & \qquad \qquad \qquad 6 \\
 & P(\text{Não gostei}|\text{Teste-1}) \approx 0,5 \quad P(\text{Não gosto}|x_i) \\
 & \qquad \qquad \qquad i=1 \\
 & 1 = (0,5) \cdot \frac{1}{4} - \frac{0}{2} \cdot \frac{1}{4} \cdot \frac{0}{3} \cdot \frac{3}{4} \cdot \frac{2}{3} \\
 & = 0
 \end{aligned}$$

Ao normalizar as duas probabilidades para que a soma seja 1, obtemos o resultado de que  $P(\text{Gostei}|\text{Teste-1})$  é 1 e  $P(\text{Não Gostei}|\text{Teste-1})$  é 0. No caso do Teste-2, o resultado é exatamente o oposto, onde  $P(\text{Gostei}|\text{Teste-2})$  é 0. Portanto, o Teste-1 deve ser recomendado ao usuário ativo em vez do Teste-2. Este é o mesmo resultado que obtivemos na inspeção visual deste exemplo.

Quando a suavização laplaciana é utilizada, não obteremos tais valores de probabilidade binária para as diversas classes, embora uma das classes obtenha uma probabilidade muito maior do que a outra. Nesses casos, todos os exemplos de teste podem ser classificados em ordem de probabilidade prevista de um "Curtir" e recomendados ao usuário. A suavização laplaciana é aconselhável porque um único valor 0 na forma produktívabio da expressão do lado direito da regra de Bayes pode resultar em um valor de probabilidade condicional de 0.

#### 4.4.4 Classificadores baseados em regras

Classificadores baseados em regras podem ser projetados de várias maneiras, incluindo métodos leave-one-out, bem como métodos associativos. Uma descrição detalhada dos vários tipos de classificadores baseados em regras é fornecida em [18, 22]. A seguir, discutiremos apenas classificadores associativos porque eles são baseados nos princípios simples de regras de associação. Uma discussão sobre métodos baseados em regras é fornecida na seção 3.3 do Capítulo 3. Consulte essa seção para as definições básicas de regras de associação e suas medidas, como suporte e confiança. O suporte de uma regra define a fração de linhas que satisfazem tanto o antecedente quanto o consequente de uma regra. A confiança de uma regra é a fração de linhas que satisfazem o consequente, das linhas já conhecidas por satisfazer o antecedente. O conceito de uma linha "satisfazendo" o antecedente ou consequente é descrito em mais detalhes abaixo.

Classificadores baseados em regras em sistemas baseados em conteúdo são semelhantes aos classificadores baseados em regras em filtragem colaborativa. Nas regras item-item da filtragem colaborativa, tanto os antecedentes quanto os consequentes das regras correspondem às classificações dos itens. A principal diferença é que os antecedentes das regras na filtragem colaborativa correspondem3 às classificações de vários itens.

---

<sup>3</sup>Uma abordagem diferente na filtragem colaborativa é aproveitar as regras de usuário-usuário. Para regras de usuário-usuário, os antecedentes e consequentes podem conter as classificações de usuários específicos. Consulte a seção 3.3 do Capítulo 3.

enquanto os antecedentes das regras nos métodos baseados em conteúdo correspondem à presença de palavras-chave específicas nas descrições dos itens. Portanto, as regras têm o seguinte formato:

O item contém o conjunto de palavras-chave A  $\Rightarrow$  Classificação = Curtir

O item contém o conjunto de palavras-chave B  $\Rightarrow$  Classificação=Não gostei

Portanto, diz-se que um antecedente de uma regra "satisfaz" uma linha específica (representação de palavra-chave do item) se todas as palavras-chave do antecedente estiverem contidas nessa linha. Os consequentes correspondem às diversas avaliações, que assumimos como curtidas ou descurtidas binárias para simplificar. Diz-se que uma linha satisfaz o consequente dessa regra se o valor da avaliação no consequente corresponder à variável dependente (avaliação) dessa linha.

O primeiro passo é aproveitar o perfil do usuário ativo (ou seja, documentos de treinamento) para explorar todas as regras com o nível desejado de suporte e confiança. Como em todos os métodos baseados em conteúdo, as regras são específicas para o usuário ativo em questão. Por exemplo, no caso da Tabela 4.1, o usuário ativo parece se interessar por música clássica. Nesse caso, um exemplo de regra relevante, que tem 33% de suporte e 100% de confiança, é o seguinte:

{Clássica, Sinfônica}  $\Rightarrow$  Curtir

Portanto, a ideia básica é minerar todas essas regras para um determinado usuário ativo. Em seguida, para itens-alvo cujos interesses do usuário são desconhecidos, determina-se quais regras são acionadas. Uma regra é acionada por uma descrição de item-alvo se as palavras-chave antecedentes da primeira estiverem incluídas na última. Uma vez que todas essas regras acionadas tenham sido determinadas para o usuário ativo, a classificação média nos consequentes dessas regras é relatada como a classificação do item-alvo. Existem muitas heurísticas diferentes para combinar as classificações dos consequentes. Por exemplo, podemos optar por ponderar a classificação com a confiança da regra ao calcular a média. Caso nenhuma regra seja acionada, a heurística padrão precisa ser usada. Por exemplo, pode-se determinar a classificação média do usuário ativo sobre todos os itens e também determinar a classificação média do item-alvo por todos os usuários. A média dessas duas quantidades é relatada. Portanto, a abordagem geral para classificação baseada em regras pode ser descrita da seguinte forma:

1. (Fase de treinamento): Determine todas as regras relevantes do perfil do usuário no nível desejado de suporte mínimo e confiança do conjunto de dados de treinamento DL.
2. (Fase de teste) Para cada descrição de item na DU, classifique a , determinar as regras de disparo e um classificação média. Classifique os itens na DU com base nessa classificação média.

Uma vantagem dos sistemas baseados em regras é o alto nível de interpretabilidade que eles oferecem. Por exemplo, para um item recomendado, pode-se usar as palavras-chave no antecedente das regras disparadas para dar uma recomendação ao usuário-alvo sobre por que ele pode gostar de um item específico.

#### 4.4.4.1 Exemplo de métodos baseados em regras

Para ilustrar o uso de métodos baseados em regras, forneceremos um exemplo das regras geradas para o usuário ativo na Tabela 4.1. Com um nível de suporte de 33% e um nível de confiança de

75%, as seguintes regras são geradas junto com seus valores de suporte-confiança:

- Regra 1: {Clássico}  $\rightarrow$  Curtir (50%, 100%)
- Regra 2: {Sinfonia}  $\rightarrow$  Curtir (33%, 100%)
- Regra 3: {Clássica, Sinfônica}  $\rightarrow$  Curtir (33%, 100%)
- Regra 4: {Bateria, Guitarra}  $\rightarrow$  Não gosto (33%, 100%)
- Regra 5: {Tambores}  $\rightarrow$  Não gosto (33%, 100%)
- Regra 6: {Bater}  $\rightarrow$  Não gostar (33%, 100%)
- Regra 7: {Guitarra}  $\rightarrow$  Não gosto (50%, 75%)

As regras mencionadas são classificadas principalmente em ordem decrescente de confiança, com os empates desfeitos em ordem decrescente de apoio. É evidente que a regra 2 é acionada pelo Teste-1, enquanto as regras 5 e 6 são acionadas pelo Teste-2. Portanto, o Teste-1 deve ser preferido ao Teste-2 como recomendação ao usuário ativo. Observe que as regras acionadas pelo Teste-1 também fornecem uma compreensão de por que ele deve ser considerado a melhor recomendação para o usuário ativo.

Essas explicações costumam ser muito úteis em sistemas de recomendação, tanto da perspectiva do cliente quanto da perspectiva do comerciante.

#### 4.4.5 Modelos baseados em regressão

Modelos baseados em regressão têm a vantagem de poderem ser usados para vários tipos de classificações, como classificações binárias, classificações baseadas em intervalos ou classificações numéricas. Grandes classes de modelos de regressão, como modelos lineares, modelos de regressão logística e modelos probit ordenados, podem ser usadas para modelar vários tipos de classificações. Aqui, descreveremos o modelo mais simples, conhecido como regressão linear. As notas bibliográficas contêm indicações para métodos de regressão mais sofisticados.

Seja  $DL$  uma matriz  $n \times d$  representando os  $n$  documentos no conjunto de treinamento rotulado  $DL$  em um léxico de tamanho  $d$ . Da mesma forma, seja  $y$  um vetor coluna  $n$ -dimensional contendo as avaliações do usuário ativo para os  $n$  documentos no conjunto de treinamento. A ideia básica da regressão linear é assumir que as avaliações podem ser modeladas como uma função linear das frequências das palavras. Seja  $W$  um vetor linha  $d$ -dimensional representando os coeficientes de cada palavra na função linear que relaciona as frequências das palavras à avaliação. Então, o modelo de regressão linear assume que as frequências das palavras na matriz de treinamento  $DL$  estão relacionadas aos vetores de avaliação da seguinte forma:

$$\bar{y} \approx DL\bar{W} \quad (4.12)$$

Portanto, o vetor  $(DL\bar{W} - y)$  é um vetor  $n$ -dimensional de erros de predição. Para maximizar a qualidade da predição, deve-se minimizar a norma quadrada deste vetor. Além disso, um termo de regularização  $\gamma \|W\|^2$  pode ser adicionado à função objetivo para reduzir o sobreajuste. Essa forma de regularização também é chamada de regularização de Tikhonov. Aqui,  $\gamma > 0$  é o parâmetro de regularização. Portanto, a função objetivo  $O$  pode ser expressa da seguinte forma:

$$\text{Minimize } O = \|DL\bar{W} - y\|^2 + \gamma \|W\|^2 \quad (4.13)$$

O problema pode ser resolvido definindo o gradiente desta função objetivo em relação a  $W$  como 0. Isso resulta na seguinte condição:

$$DT_{eu}(DL\bar{W} - y) + \gamma W = 0$$

$$(LDL^T DT + \gamma I)W = DT^T L\bar{y}$$

Tabela 4.2: A família de modelos de regressão e aplicabilidade a vários tipos de classificações

Modelo de Regressão	Natureza da classificação (variável alvo)
Regressão Linear	Real
Regressão Polinomial	Real
Regressão Kernel	Real
Regressão Logística Binária	Unário, Binário
Regressão Logística Multidirecional	Categórico, Ordinal
Probit	Unário, Binário
Probit Multidirecional	Categórico, Ordinal
Probit Ordenado	Ordinal, baseado em intervalos

A matriz  $(DT LDL + \gamma I)$  pode ser demonstrada como sendo positiva-definida e, portanto, invertível (ver Exercício 7). Portanto, podemos resolver diretamente o vetor peso  $W$  da seguinte forma:

$$\bar{WT} = (DT LDL + \gamma I)^{-1} DT \bar{y} \quad (4.14)$$

Aqui,  $I$  é uma matriz identidade  $ad \times d$ . Portanto, uma solução de forma fechada sempre existe para  $\bar{WT}$ . Para qualquer vetor de documento dado (descrição do item)  $X$  do conjunto não rotulado  $D$  sua classificação, pode ser previsto como o produto escalar entre  $W$  e  $X$ . A regularização de Tikhonov usa o Termo de regularização  $L2 \gamma \cdot \|W\|^2$ . Também é possível usar a regularização  $L1$ , na qual este termo é substituído por  $\gamma \cdot \|W\|$ . O problema de otimização resultante não possui uma solução de forma fechada, e métodos de descida de gradiente devem ser utilizados. Esta forma de regularização, também conhecido como Lasso [242], pode ser usado na dupla função de seleção de recursos. Isso ocorre porque tais métodos têm a tendência de selecionar vetores de coeficientes esparsos para  $W$ , nos quais a maioria componentes de  $W$  assumem o valor 0. Tais características podem ser descartadas. Portanto, L1- Os métodos de regularização fornecem insights altamente interpretáveis sobre subconjuntos importantes de recursos para o processo de recomendação. Uma discussão detalhada desses modelos pode ser encontrada em [22].

O modelo linear é um exemplo de um modelo de regressão adequado para valores reais classificações. Na prática, as classificações podem ser unárias, binárias, baseadas em intervalos ou categóricas (pequenas número de valores ordinais). Vários modelos lineares foram projetados para diferentes tipos de variáveis de classe alvo. Alguns exemplos incluem regressão logística, regressão probit, ordenada regressão probit e regressão não linear. Classificações unárias são frequentemente tratadas como classificações binárias, nas quais os itens não rotulados são tratados como instâncias negativas. No entanto, Existem modelos positivos não rotulados (PU) para esses casos [364]. A regressão probit ordenada é especialmente útil para classificações baseadas em intervalos. Além disso, modelos de regressão não linear, como regressão polinomial e regressão kernel, podem ser usadas em casos onde a dependência entre as características e as variáveis alvo é não linear. Quando o número de características é grande e o número de amostras de treinamento é pequeno, os modelos lineares geralmente têm um desempenho muito bom e pode, de fato, superar os modelos não lineares. Isso ocorre porque os modelos lineares são menos propensos a sobreajuste. A Tabela 4.2 mostra o mapeamento entre os vários modelos de regressão e a natureza da variável alvo (classificação).

#### 4.4.6 Outros modelos de aprendizagem e visão geral comparativa

Como o problema da filtragem baseada em conteúdo é uma aplicação direta da classificação e da modelagem de regressão, muitas outras técnicas da literatura podem ser utilizadas. Uma discussão detalhada de vários modelos de classificação podem ser encontrados em [18, 86, 242, 436]. O modelo de árvore de decisão

discutido no Capítulo 3 também pode ser aplicado a métodos baseados em conteúdo. No entanto, para dados de alta dimensionalidade, como texto, as árvores de decisão geralmente não fornecem resultados muito eficazes. Resultados experimentais [477] demonstraram o baixo desempenho das árvores de decisão em comparação com outros métodos de classificação. Embora os classificadores baseados em regras estejam intimamente relacionados às árvores de decisão, eles frequentemente podem fornecer resultados superiores porque não pressupõem um particionamento estrito do espaço de características. Resultados bem-sucedidos foram obtidos com classificadores baseados em regras para classificação de e-mails [164, 165]. Entre os vários modelos, a abordagem Bayesiana tem a vantagem de poder lidar com todos os tipos de variáveis de características com o uso de um modelo apropriado. Os modelos baseados em regressão são muito robustos e podem lidar com todas as formas de variáveis-alvo. A regressão logística e a regressão probit ordenada são particularmente úteis para classificações binárias e baseadas em intervalos.

No caso de classificações binárias, máquinas de vetores de suporte [114] são uma escolha popular. Máquinas de vetores de suporte são muito semelhantes à regressão logística; a principal diferença é que a perda é quantificada como uma perda de dobradiça em vez de com o uso da função logit. Máquinas de vetores de suporte são altamente resistentes ao sobreajuste, e existem inúmeras implementações prontas para uso. Máquinas de vetores de suporte lineares e baseadas em kernel têm sido usadas na literatura. Para o caso de dados de alta dimensão, como texto, observou-se que máquinas de vetores de suporte lineares são suficientes. Para tais casos, métodos especializados com desempenho linear [283] foram projetados. Embora redes neurais [87] possam ser usadas para construir modelos arbitrariamente complexos, elas não são aconselháveis quando a quantidade de dados disponíveis é pequena. Isso ocorre porque as redes neurais são sensíveis ao ruído nos dados subjacentes e podem sobreajustar os dados de treinamento quando seu tamanho é pequeno.

#### 4.4.7 Explicações em Sistemas Baseados em Conteúdo

Como os sistemas baseados em conteúdo extraem modelos com base em características de conteúdo, eles frequentemente fornecem insights altamente interpretáveis para o processo de recomendação. Por exemplo, em um sistema de recomendação de filmes, muitas vezes é útil apresentar ao usuário uma razão pela qual ele pode gostar de um filme específico, como a presença de uma característica específica de gênero, característica de ator ou um conjunto informativo de palavras-chave. Como resultado, o usuário ativo poderá fazer uma escolha mais informada sobre se deve ou não assistir a esse filme. Da mesma forma, um conjunto descritivo de palavras-chave em um sistema de recomendação musical pode fornecer uma melhor compreensão de por que um usuário pode gostar de uma faixa específica. Como exemplo específico, a rádio Pandora Internet [693] fornece explicações para faixas recomendadas, como as seguintes:

"Estamos tocando essa faixa porque ela apresenta raízes de trance, batidas four-on-the-floor, influências de disco, um talento para ganchos cativantes, batidas feitas para dançar, batidas de bateria diretas, pronúncia clara, letras românticas, letras narrativas, construção/ruptura sutil, uma introdução rítmica, uso de harmonias modais, uso de padrões de acordes, preenchimentos leves de bateria, ênfase na performance instrumental, um riff de baixo de sintetizador, riffs de sintetizador, uso sutil de sintetizadores arpejados, sintetizadores com muitos efeitos e swoops de sintetizador."

Cada uma dessas características relatadas pode ser vista como uma característica importante, responsável pela classificação da instância de teste como "semelhante". Observe que tais explicações detalhadas frequentemente faltam em sistemas colaborativos, onde uma recomendação pode ser explicada apenas em termos de itens semelhantes, e não em termos de características detalhadas desses itens. A natureza e a extensão dos insights são, no entanto, altamente sensíveis ao modelo específico utilizado. Por exemplo, o modelo de Bayes e os sistemas baseados em regras são altamente

interpretável em termos da causalidade específica da classificação. Considere o exemplo da Tabela 4.1 , em que a seguinte regra é acionada para o Teste-1:

{Sinfonia}  $\rightarrow$  Curtir

É evidente que o item descrito pelo Teste-1 foi recomendado ao usuário por se tratar de uma sinfonia. Da mesma forma, no modelo de classificação de Bayes, é evidente que a contribuição de  $P(\text{Sinfonia}|\text{Semelhante})$  é maior na fórmula multiplicativa para classificação. Outros modelos, como os de regressão linear e não linear, são mais difíceis de interpretar. No entanto, certas instâncias desses modelos, como o Lasso, fornecem insights importantes sobre as características mais relevantes para o processo de classificação.

## 4.5 Baseado em conteúdo versus colaborativo

### Recomendações

---

É instrutivo comparar os métodos baseados em conteúdo com os métodos colaborativos discutidos nos Capítulos 2 e 3. Os métodos baseados em conteúdo apresentam diversas vantagens e desvantagens em comparação com os métodos colaborativos. As vantagens dos métodos baseados em conteúdo são as seguintes:

1. Quando um novo item é adicionado a uma matriz de classificação, ele não contém avaliações dos diversos usuários. Nenhum dos métodos de filtragem colaborativa baseados em memória e em modelos recomendaria tal item, pois não há avaliações suficientes disponíveis para fins de recomendação. Por outro lado, no caso de métodos baseados em conteúdo, os itens avaliados anteriormente por um determinado usuário são utilizados para fazer recomendações.

Portanto, desde que o usuário não seja novo, recomendações significativas sempre podem ser feitas de forma a tratar o novo item de forma justa em comparação a outros itens.

Os sistemas colaborativos têm problemas de inicialização a frio tanto para novos usuários quanto para novos itens, enquanto os sistemas baseados em conteúdo têm problemas de inicialização a frio apenas para novos usuários.

2. Conforme discutido na seção anterior, métodos baseados em conteúdo fornecem explicações em termos das características dos itens. Isso muitas vezes não é possível com recomendações colaborativas.

3. Métodos baseados em conteúdo geralmente podem ser usados com classificadores de texto prontos para uso.

Além disso, cada problema de classificação específico do usuário geralmente não é muito grande, como no caso de sistemas colaborativos. Portanto, eles são particularmente fáceis de usar, com relativamente pouco esforço de engenharia.

Por outro lado, os métodos baseados em conteúdo também apresentam diversas desvantagens que não estão presentes nos recomendadores colaborativos.

1. Sistemas baseados em conteúdo tendem a encontrar itens semelhantes aos que o usuário já viu. Esse problema é chamado de superespecialização. É sempre desejável ter um certo grau de novidade e serendipidade nas recomendações. Novidade se refere ao fato de o item ser diferente daquele que o usuário já viu. Da mesma forma, serendipidade implica que o usuário gostaria de descobrir itens surpreendentemente relevantes que, de outra forma, não teria encontrado. Esse é um problema para sistemas baseados em conteúdo, nos quais modelos de classificação baseados em atributos tendem a recomendar itens muito semelhantes.

Por exemplo, se um usuário nunca ouviu ou avaliou música clássica, um recurso baseado em conteúdo

O sistema normalmente não recomendará tal item a ela porque a música clássica será descrita por valores de atributos muito diferentes daqueles que o usuário avaliou até o momento. Por outro lado, um sistema colaborativo pode recomendar tais itens alavancando os interesses de seu grupo de pares. Por exemplo, um sistema colaborativo pode inferir automaticamente uma associação surpreendente entre certas músicas pop e músicas clássicas e recomendar as músicas clássicas correspondentes a um usuário que seja um amante da música pop. A superespecialização e a falta de serendipidade são os dois desafios mais significativos dos sistemas de recomendação baseados em conteúdo.

2. Embora sistemas baseados em conteúdo ajudem a resolver problemas de inicialização a frio para novos itens, eles não ajudam a resolver esses problemas para novos usuários. De fato, para novos usuários, o problema em sistemas baseados em conteúdo pode ser mais grave, pois um modelo de classificação de texto geralmente requer um número suficiente de documentos de treinamento para evitar overfitting.  
Pareceria um desperdício que os dados de treinamento de todos os outros usuários fossem descartados e apenas o (pequeno) conjunto de dados de treinamento específico para um único usuário fosse aproveitado.

Apesar dessas desvantagens, os sistemas baseados em conteúdo frequentemente complementam muito bem os sistemas colaborativos devido à sua capacidade de alavancar o conhecimento baseado em conteúdo no processo de recomendação. Esse comportamento complementar é frequentemente aproveitado em sistemas de recomendação híbridos (cf. Capítulo 6), nos quais o objetivo é combinar o melhor dos dois mundos para criar um sistema de recomendação ainda mais robusto. Em geral, os sistemas baseados em conteúdo raramente são usados isoladamente e, geralmente, são usados em combinação com outros tipos de sistemas de recomendação.

## 4.6 Usando modelos baseados em conteúdo para filtragem colaborativa

---

Existe uma conexão interessante entre modelos de filtragem colaborativa e métodos baseados em conteúdo. Acontece que métodos baseados em conteúdo podem ser usados diretamente para filtragem colaborativa. Embora a descrição do conteúdo de um item se refira às suas palavras-chave descritivas, é possível imaginar cenários em que as avaliações dos usuários são aproveitadas para definir descrições baseadas em conteúdo. Para cada item, pode-se concatenar o nome de usuário (ou identificador) de um usuário que avaliou o item com o valor dessa avaliação para criar uma nova "palavra-chave". Portanto, cada item seria descrito em termos de tantas palavras-chave quanto o número de avaliações desse item. Por exemplo, considere um cenário em que as descrições de vários filmes são as seguintes:

O Exterminador do Futuro: John#Curtir, Alice#Não Curtir, Tom#Curtir

Alienígenas: John#Gostei, Peter#Não gostei, Alice#Não gostei, Sayani#Gostei

Gladiador: Jack#Gostar, Mary#Gostar, Alice#Gostar

O símbolo “#” é usado para denotar a demarcação da concatenação e garantir uma palavra-chave única para cada combinação de usuário-avaliação. Essa abordagem geralmente é mais eficaz quando o número de avaliações possíveis é pequeno (por exemplo, avaliações unárias ou binárias). Após a construção de tal descrição baseada em conteúdo, ela pode ser usada em conjunto com um algoritmo baseado em conteúdo pronto para uso. Há um mapeamento quase um para um entre os métodos resultantes e os vários modelos de filtragem colaborativa, dependendo do método base usado para classificação. Embora cada uma dessas técnicas seja mapeada para um modelo de filtragem colaborativa,

o inverso não é verdadeiro porque muitos métodos de filtragem colaborativa não podem ser capturados por esta abordagem. No entanto, fornecemos alguns exemplos de mapeamento:

1. Um classificador de vizinho mais próximo nesta representação mapeia aproximadamente um modelo de vizinhança baseado em itens para filtragem colaborativa (cf. seção 2.3.2 do Capítulo 2).
2. Um modelo de regressão sobre o conteúdo mapeia aproximadamente para um modelo de regressão do usuário para filtragem colaborativa (cf. seção 2.6.1 do Capítulo 2).
3. Um classificador baseado em regras sobre o conteúdo mapeia aproximadamente para um classificador baseado em regras do usuário classificador para filtragem colaborativa (cf. seção 3.3.2 do Capítulo 3).
4. Um classificador Bayes no conteúdo mapeia aproximadamente para um modelo Bayes do usuário para filtragem colaborativa (cf. Exercício 4 do Capítulo 3).

Portanto, muitos métodos de filtragem colaborativa podem ser capturados definindo uma representação de conteúdo apropriada e utilizando diretamente métodos baseados em conteúdo prontos para uso. Estes

As observações são importantes porque abrem inúmeras oportunidades para hibridização.

Por exemplo, é possível combinar palavras-chave baseadas em classificações com palavras-chave descritivas reais para obter um modelo ainda mais robusto. De fato, essa abordagem é frequentemente usada em alguns modelos híbridos sistemas de recomendação. Essa abordagem não desperdiça mais os dados de classificação disponíveis outros usuários e combina o poder dos modelos colaborativos e baseados em conteúdo dentro de um estrutura unificada.

#### 4.6.1 Aproveitando os perfis de usuário

Outro caso em que modelos semelhantes à filtragem colaborativa podem ser criados com atributos de conteúdo é quando os perfis de usuário estão disponíveis na forma de palavras-chave especificadas. Por exemplo, Os usuários podem optar por especificar seus interesses particulares na forma de palavras-chave. Nesses casos, Em vez de criar um modelo de classificação local para cada usuário, pode-se criar um modelo de classificação global para todos os usuários, utilizando as características do usuário. Para cada combinação usuário-item, uma representação centrada no conteúdo pode ser criada usando o produto de Kronecker dos vetores de atributos do usuário e item correspondentes [50]. Um modelo de classificação ou regressão é construído sobre esta representação para mapear combinações de itens de usuário para classificações. Tal abordagem é descrita em detalhes na seção 8.5.3 do Capítulo 8.

## 4.7 Resumo

---

Este capítulo apresenta a metodologia dos sistemas de recomendação baseados em conteúdo, nos quais Modelos de treinamento específicos para o usuário são criados para o processo de recomendação. Os atributos de conteúdo nas descrições dos itens são combinados com as avaliações dos usuários para criar perfis de usuário. Modelos de classificação são criados com base nesses modelos. Esses modelos são então usados para classificar descrições de itens que ainda não foram avaliadas pelo usuário. Esses sistemas utilizam diversos modelos de classificação e regressão, como classificadores de vizinho mais próximo, métodos baseados em regras, o método de Bayes e modelos lineares. O método de Bayes foi usado com grande sucesso em uma variedade de cenários devido à sua capacidade de lidar com vários tipos de conteúdo. Os sistemas baseados em conteúdo têm a vantagem de poderem lidar com problemas de inicialização a frio em relação a novos itens, embora não consigam lidar com problemas de inicialização a frio. em relação a novos usuários. A serendipidade dos sistemas baseados em conteúdo é relativamente baixa porque recomendações baseadas em conteúdo são baseadas no conteúdo dos itens previamente avaliados por o usuário.

## 4.8 Notas Bibliográficas

---

Os primeiros sistemas baseados em conteúdo foram atribuídos ao trabalho em [60] e aos sistemas Syskill & Webert [82, 476–478]. Fab, no entanto, usa um projeto de hibridização parcial no qual o grupo de pares é determinado usando métodos baseados em conteúdo, mas as avaliações de outros usuários são alavancadas no processo de recomendação. Os trabalhos em [5, 376, 477] fornecem excelentes artigos de visão geral sobre sistemas de recomendação baseados em conteúdo. Este último trabalho foi projetado para encontrar sites interessantes e, portanto, vários classificadores de texto foram testados quanto à sua eficácia. Em particular, o trabalho em [82] fornece uma série de indicadores úteis sobre o desempenho relativo de vários sistemas baseados em conteúdo. Métodos probabilísticos para modelagem de usuários são discutidos em [83]. O trabalho em [163, 164] é notável por seu uso de sistemas baseados em regras na classificação de e-mail. O feedback de relevância de Rocchio [511] também foi utilizado durante os primeiros anos, embora o trabalho não tenha embasamento teórico e possa frequentemente apresentar desempenho insatisfatório em diversos cenários. Diversos métodos de classificação de texto, que podem ser utilizados para recomendações baseadas em conteúdo, são discutidos em [21, 22, 400]. Uma discussão sobre a noção de serendipidade no contexto da recuperação de informação é apresentada em [599]. Alguns sistemas baseados em conteúdo filtram explicitamente itens muito semelhantes para aprimorar a serendipidade. O trabalho em [418] discute como se pode ir além das métricas de precisão para medir a qualidade de um sistema de recomendação.

Métodos para extração, limpeza e seleção de características na classificação de texto são discutidos em [21, 364, 400]. A extração do bloco de conteúdo principal de uma página da Web contendo vários blocos é obtida com a ajuda do algoritmo de correspondência de árvore, que pode ser encontrado em [364, 662]. O uso de representações visuais para extrair a estrutura do conteúdo de páginas da Web é descrito em [126]. Uma discussão detalhada sobre medidas de seleção de características para classificação pode ser encontrada em [18]. Uma pesquisa recente sobre classificação de texto [21] discute algoritmos de seleção de características para o caso específico de dados de texto.

Vários sistemas do mundo real foram projetados com o uso de sistemas baseados em conteúdo. Alguns dos primeiros são Fab [60] e Syskill & Webert [477]. Um sistema antigo, conhecido como Personal WebWatcher [438, 439], faz recomendações aprendendo os interesses dos usuários a partir das páginas da Web que eles visitam. Além disso, as páginas da Web vinculadas pela página visitada são usadas no processo de recomendação. O sistema Letizia [356] usa uma extensão de navegador da Web para rastrear o comportamento de navegação do usuário e a utiliza para fazer recomendações. Um sistema conhecido como Dynamic-Profiler usa uma taxonomia predefinida de categorias para fazer recomendações de notícias aos usuários em tempo real [636]. Nesse caso, os registros da Web do usuário são usados para aprender as preferências e fazer recomendações personalizadas. O sistema IfWeb [55] representa os interesses do usuário na forma de uma rede semântica. O sistema WebMate [150] aprende perfis de usuário na forma de vetores de palavras-chave. Este sistema é projetado para monitorar os interesses positivos do usuário em vez dos negativos. Os princípios gerais nas recomendações da Web não são muito diferentes daqueles da filtragem de notícias. Métodos para realizar recomendações de notícias são discutidos em [41, 84, 85, 392, 543, 561]. Alguns desses métodos usam representações aprimoradas, como WordNet, para melhorar o processo de modelagem.

Os sistemas de recomendação da web são geralmente mais desafiadores do que os sistemas de recomendação de notícias, pois o texto subjacente costuma ser de qualidade inferior. O sistema Citeseer [91] é capaz de descobrir publicações interessantes em um banco de dados bibliográfico identificando as citações comuns entre os artigos. Assim, ele utiliza explicitamente as citações como um mecanismo de conteúdo para determinar similaridade.

Sistemas baseados em conteúdo também têm sido utilizados em outros domínios, como livros, música e filmes. Métodos baseados em conteúdo para recomendações de livros são discutidos em [448]. O principal desafio nas recomendações musicais é a lacuna semântica entre recursos facilmente disponíveis

e a probabilidade de um usuário apreciar a música. Esta é uma característica comum entre os domínios da música e da imagem. Algum progresso na redução da lacuna semântica foi alcançado.

foi feito em [138, 139]. Pandora [693] usa os recursos extraídos no Music Genome

Projeto para fazer recomendações. O sistema ITR discute como se pode usar descrições textuais [178] de itens (por exemplo, descrições de livros ou enredos de filmes) para fazer recomendações.

Trabalhos posteriores [179] mostram como se pode integrar tags em um recomendador baseado em conteúdo.

A abordagem utiliza ferramentas linguísticas como o WordNet para extrair conhecimento para o processo de recomendação.

Um sistema de recomendação de filmes que utiliza categorização de texto é o

Sistema INTIMATE [391]. Um método que combina sistemas de recomendação baseados em conteúdo e colaborativos é discutido em [520]. Uma visão geral mais ampla dos sistemas de recomendação híbridos é

fornecido em [117]. Uma possível direção de trabalho, mencionada em [376], é aprimorar os sistemas de recomendação

baseados em conteúdo com conhecimento enciclopédico [174, 210, 211], como aquele

obtidos na Wikipédia. Alguns métodos foram projetados para usar a Wikipédia para filmes

recomendação [341]. Curiosamente, esta abordagem não melhora a precisão da

sistema de recomendação. A aplicação de conhecimento semântico avançado em sistemas baseados em conteúdo

recomendações foram mencionadas como uma direção para trabalho futuro em [376].

## 4.9 Exercícios

---

1. Considere um cenário em que um usuário fornece classificações de gosto/desgosto de um conjunto de 20 itens, em que ela classifica 9 itens como "gostei" e os restantes como "não gostei". Suponha que 7 descrições de itens contêm a palavra "thriller" e o usuário não gosta de 5 delas itens. Calcule o índice de Gini em relação à distribuição original dos dados e com respeito ao subconjunto de itens que contêm a palavra "thriller". Deve apresentar seleção algoritmos retêm essa palavra nas descrições dos itens?

2. Implementar um classificador baseado em regras com o uso de mineração de padrões de associação.

3. Considere um sistema de recomendação de filmes em que os filmes pertencem a um ou mais dos seguintes os gêneros ilustrados na tabela, e um usuário específico fornece o seguinte conjunto de classificações para cada um dos filmes.

Gênero ѹ Comédia	Drama	Romance	Suspense	Ação	Terror	Curtir ou			Não gosto
ID do filme ѹ									
		0		0	0	0			Não gosto
1 2	1 1	1	1 1	0	1	0			Não gosto
	1	1	0	0	0	0			Não gosto
3 4	0	0	0	1	1	0			Como
5	0	1	0	1	1	1			Como
6	0	0	0	0					Como
Teste-1	0	0	0	1	1 0	1 1			?
Teste-2	0	1	1	0	0	0			?

Explore todas as regras com pelo menos 33% de apoio e 75% de confiança. Com base nessas regras, você recomendaria o item Teste-1 ou Teste-2 ao usuário?

4. Implemente um classificador Bayes com suavização laplaciana.

5. Repita o Exercício 3 usando um classificador de Bayes. Não use suavização laplaciana.

Explique por que a suavização laplaciana é importante neste caso.

6. Repita o Exercício 3 com o uso de um classificador de vizinho mais próximo.
7. Para uma matriz de dados de treinamento D, a regressão de mínimos quadrados regularizada requer a inversão da matriz  $(DT D + \gamma I)$ , onde  $\gamma > 0$ . Mostre que esta matriz é sempre invertível.
8. A distribuição  $\chi^2$  é definida pela seguinte fórmula, conforme discutido no capítulo:

$$\chi^2 = \sum_{i=1}^P \frac{(O_i - E_i)^2}{E_i}$$

Mostre que para uma tabela de contingência  $2 \times 2$ , a fórmula acima mencionada pode ser reescrita da seguinte forma:

$$\frac{(O_1 + O_2 + O_3 + O_4) \cdot (O_1 O_4 - O_2 O_3)^2}{(O_1 + O_2) \cdot (O_3 + O_4) \cdot (O_1 + O_3) \cdot (O_2 + O_4)} \chi^2 =$$

Aqui,  $O_1 \dots O_4$  são definidos da mesma forma que no exemplo tabular no texto.