

Aprendizado Descritivo

Aula 11 – Algoritmos de beam search para descoberta de subgrupos

Professor Renato Vimieiro

DCC/ICEx/UFMG

Introdução

- Na aula passada, vimos como o SD-Map se baseia no FP-Growth para encontrar de forma exaustiva subgrupos interessantes na base de dados
- Um dos grandes problemas à busca exaustiva em descoberta de subgrupos é o fato das medidas de qualidade não serem anti-monotônicas
- Apesar da estimativa de custo permitir a poda da árvore, o algoritmo ainda pode ser bastante lento, pois as podas não são aplicáveis nas primeiras iterações
- Assim, surge a necessidade de métodos heurísticos que não varram completamente o espaço de busca, mas que retornem boas soluções
- Hoje veremos uma abordagem gulosa para a solução do problema

Algoritmo SD

- O algoritmo SD foi proposto por Gamberger e Lavrac em 2002
- O SD se baseia na heurística de busca em feixe (beam search)
- A ideia do beam search é muito similar a best-first search, em que as soluções candidatas são exploradas conforme uma função de estimativa qualidade
 - Soluções mais promissoras são exploradas primeiro
- No caso de beam search, o conjunto de soluções a serem exploradas é limitado
 - O parâmetro beam width limita o algoritmo a explorar, no máximo, esse número de soluções candidatas
 - A cada iteração, os k melhores candidatos são mantidos para serem refinados na próxima iteração

Algoritmo SD

- Antes de aprofundarmos a discussão em relação ao algoritmo em si, precisamos definir a linguagem de descrição que determina o espaço de busca do SD
- Teoricamente, o algoritmo pode ser facilmente adaptado para outras linguagens. Porém, os autores definiram uma linguagem específica ao apresentarem o SD
- As descrições continuam sendo conjuntos (conjunções) de seletores
- Os seletores, no entanto, são definidos especificamente para o caso de atributos numéricos (reais), discretos (inteiros), e categóricos

Algoritmo SD

- Em relação ao atributo alvo, o SD foi desenvolvido especificamente para dados categóricos (binário).
 - Podendo também ser adaptado para outros tipos
- Os autores assumem que o usuário definirá um valor (classe) específico para o atributo alvo
- Assim, eles computam $dom_p(a_i) = \{v_1, v_2, \dots, v_{ip}\}$ e $dom_n(a_i) = \{w_1, w_2, \dots, w_{in}\}$,
 - O domínio de valores dos atributos nos objetos da classe positiva e da classe negativa
 - No caso de atributos numéricos, eles assumem ainda que $v_i \leq v_{i+1}$ e $w_i \leq w_{i+1}$ (os valores estão ordenados)

Algoritmo SD

- Seguindo a lógica de discretização por entropia de Fayyad-Irani, os seletores dos atributos contínuos são gerados da seguinte forma:
 - $a_i \leq (v_{ix} + w_{iy})/2$ para todo par (v_{ix}, w_{iy}) em que eles são vizinhos imediatos
 - $a_i > (v_{ix} + w_{iy})/2$ para todo par (w_{iy}, v_{ix}) vizinhos imediatos
- Para atributos inteiros (discretos), são gerados os mesmos seletores de atributos contínuos e mais:
 - $a_i = v_{ix}$ e $a_i \neq w_{iy}$
- Finalmente, os atributos categóricos são tratados da mesma forma que o segundo tipo de seletor de inteiros
 - Note que não são considerados conjuntos de valores como no caso do SD-Map, embora estejamos considerando negações

Algoritmo SD

- Uma vez definida a linguagem de descrição e, conseqüentemente, o espaço de busca, podemos detalhar o algoritmo
- Como foi dito, a busca é feita considerando-se os k (beam width) melhores candidatos segundo a função de qualidade
 - Os autores usam o Q_g
- Inicialmente, o beam é formado pelos k melhores seletores individuais
- Em seguida, o algoritmo entra em um loop em que os candidatos são refinados
 - A partir de cada candidato do beam, novas soluções candidatas são geradas considerando a inserção de novos seletores à descrição
 - O novo beam é formado com os k melhores candidatos considerando as descrições atuais e as novas candidatas que satisfizerem um limiar de suporte mínimo na classe positiva (similar ao SD-Map)
 - Soluções **irrelevantes** são descartadas do beam
- O loop é interrompido se não houver mudança de uma iteração para a outra, ou se uma profundidade máxima for alcançada (limite no número de seletores na descrição)
 - A solução final é o último beam obtido

Algoritmo SD

- Sejam $c^+(X)$ e $c^-(X)$ as coberturas de um subgrupo X na classe positiva e negativa, respectivamente.
- Um subgrupo X é dito irrelevante se existir um outro subgrupo X' tal que
 - $c^+(X) \subset c^+(X')$ e $c^-(X) \supset c^-(X')$
 - O teste de relevância tenta manter um conjunto diverso de subgrupos interessantes
- Como um subgrupo anteriormente relevante pode ser tornar irrelevante após a inclusão de uma nova descrição, a inclusão das soluções no novo beam deve ser sequencial e em ordem

Algoritmo SD

- Após encontrar o conjunto dos k (beam width) melhores subgrupos que atenderam às restrições de suporte e relevância, é sugerida a aplicação de uma etapa de pós-processamento para filtragem de descrições
- Essa etapa tem como objetivo diminuir a redundância dos subgrupos encontrados, tentando maximizar a cobertura dos exemplos (positivos) da base
- A ideia é implementar um esquema de pesos na cobertura, em que o peso de um objeto para a cobertura de um subgrupo é inversamente proporcional ao número de (outros) subgrupos que o cobriram anteriormente

Algoritmo SD

- Seja $s(o)$ o peso do objeto o para a cobertura de um subconjunto
 - Inicialmente, $s(o) = 1$ para todo objeto
- A cobertura de um subgrupo X pode ser redefinida por $c(X) = \sum 1/s(o)$
- Se um objeto ainda não foi coberto, ele contribui integralmente para a cobertura do subgrupo
 - Caso contrário, ele contribui proporcionalmente ao número de vezes que já foi coberto
- Assim, podemos escolher um subconjunto das descrições encontradas de forma gulosa
 - A cada iteração, escolhemos o subgrupo com maior cobertura positiva
 - Removemos o subgrupo do conjunto, e
 - Atualizamos o $s(o)$ dos objetos positivos cobertos pela descrição
- Repetimos o processo até alcançarmos o número de subgrupos desejados

Algoritmo CN2-SD

- Em 2004, Lavrac et al. investigaram como adaptar algoritmo para aprendizado de regras de classificação para o contexto de descoberta de subgrupos
- Eles mostraram como o algoritmo CN2 poderia ser adaptado caso a medida de qualidade fosse trocada para o contexto de descoberta de subgrupos
- O algoritmo CN2 é um algoritmo baseado em cobertura de exemplos
 - A cada iteração, o algoritmo encontra a regra com maior precisão, e remove os exemplos cobertos da base para a próxima iteração

Algoritmo CN2-SD

- Como o interesse no caso de SD não é obter um modelo global, a ideia da cobertura foi adaptada para incluir um esquema de pesos como foi feito no algoritmo de seleção do SD
- Eles discutiram dois tipos de funções de peso para a cobertura dos objetos
 - Função aditiva: $s(X, i) = 1/(i + 1)$, em que i denota o número de vezes que X foi coberto
 - Função multiplicativa: $s(X, i) = \gamma^i$ em que $\gamma \in [0,1]$ é um parâmetro definido pelo usuário
 - $\gamma = 1$ faz com que o algoritmo sempre retorne a mesma regra
 - $\gamma = 0$ faz com que o algoritmo se comporte como o original, descartando objetos já cobertos

Algoritmo CN2-SD

- A função de peso é diretamente incorporada na medida de qualidade dos subgrupos
- Especificamente, eles propuseram a seguinte modificação ao WRAcc

- $WRAcc(X) = \frac{c(X)}{N'} \left[\frac{c^+(X)}{c(X)} - \frac{P'}{N'} \right]$

- $c(X) = \sum_{o \in X} s(o)$

- $c^+(X) = \sum_{o \in X \wedge t(o)=1} s(o)$

- $N' = \sum_{o \in O} s(o)$

- $P' = \sum_{o \in O \wedge t(o)=1} s(o)$

Algoritmo CN2-SD

- O algoritmo então adota o seguinte procedimento
- A fase interna do algoritmo usa beam search para encontrar a melhor regra conforme o WRAcc adaptado
 - A largura do beam pode ser definida como o usuário desejar, mas, ao final, somente o melhor subgrupo é retornado
- A fase externa do algoritmo invoca o fase interna quantas vezes forem necessárias para alcançar o número de subgrupos desejados
 - A cada iteração, o peso dos objetos é ajustado conforme a cobertura do subgrupo.
 - Todos os objetos são considerados (não só os positivos)
 - Somente regras 'diferentes' são incluídas no resultado final. Eles consideram diferentes segundo a descrição, mas pode ser estendido para outras formas

Algoritmo CN2-SD

- O algoritmo original do CN2 possuía duas versões:
 - Ordenada: em que o resultado final era uma lista de decisão (as regras são ordenadas e a primeira aplicável determina a classe do exemplo)
 - Não-ordenada: o resultado final é um conjunto de regras de classificação (todas as regras aplicáveis são usadas para determinar a classe do exemplo)
- Essas duas versões também são consideradas na adaptação para o contexto de SD
 - Ordenada: as descrições são geradas e a classe majoritária é usada na computação da medida de qualidade
 - Não-ordenada: os subgrupos são encontrados tomando cada classe como positiva

Leitura

- Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. 2004. Subgroup Discovery with CN2-SD. J. Mach. Learn. Res. 5 (12/1/2004), 153–188.
<https://dl.acm.org/doi/10.5555/1005332.1005338>
- Dragan Gamberger and Nada Lavrac. 2002. Expert-guided subgroup discovery: methodology and application. J. Artif. Int. Res. 17, 1 (July 2002), 501–527. <https://dl.acm.org/doi/10.5555/1622810.1622825>

Aprendizado Descritivo

Aula 10 – Algoritmos exaustivos para descoberta de subgrupos

Professor Renato Vimieiro

DCC/ICEx/UFMG