Projeto de Algoritmos Baseados em Florestas de Posets para o Problema de Otimização U-curve

Instituto de Matemática e Estatística

Gustavo Estrela de Matos

5 de Novembro de 2017

Resumo

Conteúdo

Capítulo 1

Introdução

A seleção de características pode ser utilizada como um auxílio na construção de um modelo de aprendizado de máquina. Essa técnica consiste em, dado o conjunto de características observadas nas amostras, escolher um subconjunto que seja ótimo de acordo com alguma métrica. Devemos considerar a etapa de seleção de características na construção de um modelo de aprendizado quando a quantidade de características é muito grande, o que pode fazer o modelo ser muito caro computacionalmente; ou quando a quantidade de amostras é pequena comparada a complexidade do modelo original, em outras palavras, quando ocorre sobreajuste (do inglês, *overfitting*).

Mais formalmente, o problema de seleção de características consiste em um problema de otimização combinatória em que, dado um conjunto S de características, procuramos por um subconjunto $X \in \mathcal{P}(S)$ ótimo de acordo com uma função de custo $c : \mathcal{P}(S) \to \mathbb{R}_+$. É comum nas abordagens do problema explorar o fato de que o espaço de busca $\mathcal{P}(S)$ junto a relação $\subseteq$ define um reticulado booleano. No geral, a função de custo c deve ser capaz de medir quão informativas as características X são em respeito ao rótulo Y do problema de aprendizado, portanto essa função costuma depender da estimação da distribuição de probabilidade de (X, Y).

Quando ocorre a estimação da distribuição de probabilidade conjunta de (X, Y), o custo das cadeias do reticulado booleano reproduzem um fenômeno conhecido em aprendizado de máquina, "curvas em U". Para entender intuitivamente esse fenômeno, devemos observar que conforme subimos uma cadeia do reticulado estamos aumentando o número de características sendo consideradas, portanto existem mais possíveis valores de X, permitindo descrever melhor os valores de Y; por outro lado, também precisaríamos de mais amostras para estimar bem $\mathbb{P}(X, Y)$, e, quando isso não é possível, erros de estimação fazem com que o custo de X aumente.

Podemos então considerar um caso particular do problema de seleção de características em que a função de custo descreve "curvas em U" em todas as cadeias do reticulado booleano. Esse caso particular é conhecido como problema U-curve e existem na literatura algoritmos ótimos para esse problema como o `U-Curve Branch and Bound (UBB)`, `U-Curve-Search (UCS)` e `Poset Forest Search (PFS)`. A solução do problema U-Curve tem aplicações em problemas de aprendizado como projeto de W-operadores [JCJB04] e preditores na estimação de Redes Gênicas Probabilísticas [Bar+07].

O problema U-Curve é NP-difícil [Rei12], e os algoritmos apresentados na literatura tem limitações, tanto do ponto de vista de tempo de computação quanto do uso de memória. Dentre estes algoritmos, destacamos o PFS, que foi criado como um melhoramento do algoritmo UBB. O algoritmo PFS organiza o reticulado booleano em florestas, e essa organização em árvores, que são disjuntas, indica que a paralelização desse algoritmo deve trazer ganhos do ponto de vista de consumo de tempo. Além disso, a escolha de árvores para etapa de ramificação no

algoritmo também pode ser explorada e pode trazer ganhos em respeito ao consumo de tempo e de memória.

1.1 Objetivos do Trabalho

Podemos dividir os objetivos deste trabalho em objetivos gerais e específicos. **Objetivos gerais**:

- criar algoritmos para o problema U-Curve que sejam mais eficientes em consumo de tempo e/ou de memória do que as presentes soluções;
- verificar a qualidade das soluções econtradas no desenvolvimento de modelos de aprendizado computacional.

Objetivos específicos:

- estudar o algoritmo `Poset Forest Search (PFS)`;
- modificar a etapa ramificação do algoritmo `PFS` e avaliar as mudanças na dinâmica do algoritmo;
- paralelizar o algoritmo `PFS`, com as modificações feitas na etapa de ramificação (se houver melhorias com tal mudança);
- criar um novo algoritmo, de natureza paralela, para o roblema U-Curve, o `PUCS`;
- avaliar o consumo de recursos computacionais dos algoritmos criados, comparando com os algoritmos já presentes na literatura como o `UBB`;
- avaliar os conjuntos de características selecionados por cada algoritmo na seleção de modelos de aprendizado computacional, usando como exemplo conjuntos de dados do repositório UCI Machine Learning Repository.

Capítulo 2

Conceitos Fundamentais

2.1 O problema de seleção de características

A seleção de características é um problema de otimização combinatória em que procuramos o melhor subconjunto de um conjunto de características S. O espaço de busca desse problema é o conjunto potência de S, $\mathcal{P}(S)$, que é a coleção de todos os subconjuntos possíveis de S. A função de custo desse problema é uma função $c : \mathcal{P}(S) \to \mathbb{R}_+$.

Definição 2.1.1 (Problema de seleção de características)**.** *Seja S um conjunto de características, finito e não vazio, e c uma função de custo. Encontrar $X \in \mathcal{P}(S)$ tal que $c(X) \leq c(Y)$, $\forall Y \in \mathcal{P}(S)$.*

O espaço de busca do problema de seleção de características possui uma relação de ordem parcial definida pela relação $\subseteq$, portanto este conjunto é **parcialmente ordenado (poset)**.

Definição 2.1.2. *Uma **cadeia** do reticulado booleano é uma sequência X_1, X_2, ..., X_l tal que $X_1 \subseteq X_2 \subseteq \cdots \subseteq X_l$.*

2.2 Funções de custo

Nesta seção apresentaremos as duas funções de custo mais utilizadas durante este trabalho: a entropia condicional média (MCE) e a soma de subconjuntos. A primeira foi utilizada na seleção de modelos de aprendizado, enquanto a segunda foi utilizada para criação e solução de instâncias artificiais.

2.2.1 Custo de modelos de aprendizado computacional

A função de custo utilizada na solução do problema deve, de alguma forma, refletir a qualidade do conjunto de características avaliado. Por isso, diferentes aplicações de seleção de características podem ter diferentes funções de custo. No contexto de aprendizado de máquina, uma possível função de custo é a entropia condicional média (MCE), que já foi utilizada por exemplo na construção de W-operadores [DMJ06].

Definição 2.2.1. *Dado um problema de aprendizado em que Y é o conjunto de possíveis rótulos e $W = (w_1, ..., w_n)$, com $w_i \in A_i$, é o conjunto de variáveis. Seja $W' = (w_{I(1)}, w_{I(2)}, ..., w_{I(k)})$ um conjunto de variáveis (características) escolhidas, $\mathbf{X}$ uma vetor aleatório de tamanho k com $X_j \in A_{I(j)}$, e $log0 = 0$. Então, a **entropia condicional** de Y dado $\mathbf{X} = \mathbf{x}$ é:*

$$H(Y|\mathbf{X}=\mathbf{x}) = -\sum_{y\in Y} \mathbb{P}(Y=y|\mathbf{X}=\mathbf{x}) log \mathbb{P}(Y=y|\mathbf{X}=\mathbf{x})$$

Definição 2.2.2. *Sob o mesmo contexto definido em 2.2.1, definimos a* ***entropia condicional média*** *como:*

$$\mathbb{E}[H(Y|\mathbf{X})] = \sum_{\mathbf{x}\in\mathbf{X}} H(Y|\mathbf{X}=\mathbf{x})\mathbb{P}(\mathbf{X}=\mathbf{x})$$

A função H, em teoria da informação, mede o inverso da quantidade média de informação que uma variável tem. Esta função atinge valor máximo quando a distribuição de probabilidade da variável aleatória em questão é uniforme (todos valores que ela pode assumir são equiprováveis), e tem valores baixos quando essa distribuição é concentrada.

Problemas de aprendizado em que os rótulos tem uma distribuição concentrada são mais fáceis do que os problemas em que essa distribuição é menos concentrada. Tome como exemplo o problema de classificar o lançamento de uma moeda $\mathbf{x}$ em y (cara ou coroa); se toda moeda $\mathbf{x}$ é não viciada, então a distribuição de $\mathbb{P}(y|\mathbf{x})$ é pouco concentrada, por outro lado, quando a moeda é viciada, a distribuição de $\mathbb{P}(y|\mathbf{x})$ é concentrada e é mais fácil classificar este problema. Em termos mais formais, o erro do melhor classificador do problema mais fácil é menor do que o erro do melhor classificador do problema mais difícil.

Portanto, como a função H é capaz de medir a concentração da distribuição de Y dado $\mathbf{X}=\mathbf{x}$, e quanto maior esta concentração mais fácil é o modelo de aprendizado, podemos dizer que a função de custo $\mathbb{E}[H(Y|\mathbf{X})]$ pode representar a qualidade do modelo de classificação que usa o conjunto de características de $\mathbf{X}$.

Agora, como já entendemos o funcionamento da função de custo MCE e como ela se relaciona com a qualidade do conjunto de características avaliado, vamos entender o que acontece no modelo de aprendizado e na função de custo que usamos como exemplo quando percorremos uma cadeia do reticulado.

Uma cadeia do poset pode ser vista como uma sequência de possíveis escolhas de conjuntos de características ao qual a cada passo adicionamos uma característica. Isso significa que a cada passo dado a variável $\mathbf{x}$ ganha uma componente a mais. Quando estamos no início da cadeia, poucas variáveis do problema são consideradas, portanto há uma grande abstração dos dados dos objetos sendo classificados, e conforme subimos uma cadeia, diminuímos a abstração dos dados e isso faz com que a distribuição de Y dado $\mathbf{x}$ se concentre.

Essa concentração da distribuição da probabilidade indica que o custo dos subconjuntos deve diminuir conforme subimos por uma cadeia do reticulado, ou seja, este raciocínio nos leva a pensar que adicionar características sempre melhora a classificação; de fato, o valor de $\mathbb{E}[H(Y|\mathbf{X})]$ deve diminuir (até algum ponto de saturação) conforme aumentamos o número de variáveis do problema. Mas se isso é verdade, por que fazemos seleção de características? A inconsistência entre esse raciocínio e a motivação para seleção de característica é que essa linha de raciocínio negligenciou o fato de que problemas de classificação (supervisada) dependem de uma amostra da distribuição de Y dado $\mathbf{X}=\mathbf{x}$, ou seja, não sabemos nem ao menos calcular $H(Y|\mathbf{X}=\mathbf{x})$, podemos apenas estimar o seu valor a partir da amostra.

A amostra da distribuição de Y dado $\mathbf{X}=\mathbf{x}$ é obtida do conjunto de treinamento do problema de aprendizado e quando o número de amostras não é grande o suficiente a qualidade do classificador é comprometida. Além disso, o número de amostras necessárias deve crescer conforme aumentamos a complexidade do modelo de aprendizado utilizado. Considerando que quando subimos uma cadeia do reticulado booleano estamos aumentando a complexidade do modelo, temos que, a partir de um certo ponto, a qualidade do classificador que utiliza tal conjunto de características deve piorar.

Portanto, é esperado que a função de custo descreva um formato de U nas cadeias do reticulado. No começo da cadeia, o custo deve diminuir por conta da maior granularidade dos dados de entrada, até algum ponto onde a limitação no número de amostras combinada com o aumento da complexidade do modelo causem erros de estimação que aumentam o erro do classificador criado em tal modelo.

No cálculo da entropia condicional média, o efeito do aumento da complexidade de $\mathbf{X}$ é a estimação ruim de $\mathbb{P}(Y = y|\mathbf{X} = \mathbf{x})$. Contorna-se este problema modificando a entropia condicional média para penalizar a entropia de Y quando $\mathbf{x}$ foi observado poucas vezes. A função de custo utilizada é, então:

$$\hat{\mathbb{E}}[H(Y|\mathbf{X})] = \frac{N}{t}\sum_{\mathbf{x}\in\mathbf{X}} H(Y|\mathbf{X} = \mathbf{x})\mathbb{P}(\mathbf{X} = \mathbf{x})$$

2.2.2 Soma de subconjuntos

Para se avaliar o desempenho dos algoritmos criados neste trabalho, utilizamos instâncias artificiais que são reduções do problema da soma de subconjuntos. Este problema consiste em, dado um conjunto finito de inteiros não-negativos S e um inteiro não-negativo t, descobrir se há um subconjunto de S que soma t. Podemos resolver este problema com a solução de uma instância do problema de seleção de características onde o conjunto de características é S' uma cópia de S e a função de custo é c:

$$c(X) = |t - \sum_{x\in X} x|, \text{ para todo } X \in \mathcal{P}(S').$$

Assim como a função de custo MCE, a função de custo de somas de subconjuntos também apresenta um formato interessante nas cadeias do reticulado booleano. Para toda cadeia com elementos $A \subseteq B \subseteq C$ vale que $c(B) \leq max\{c(A), c(B)\}$. Vamos provar esta propriedade para dois casos disjuntos, quando $|t - \sum_{b\in B} b| > 0$ e quando $|t - \sum_{b\in B} b| \leq 0$. Começamos a demonstração definindo $D = B \setminus A$ e $E = C \setminus B$.

- se $|t - \sum_{b\in B} b| > 0$, então:

$$\begin{aligned} c(B) &= |t - \sum_{b\in B} b| \\ &\leq |t - \sum_{b\in B} b + \sum_{d\in D} d| \quad \text{(pois } S \text{ contém apenas números positivos e } t - \sum_{b\in B} b > 0) \\ &= |t - \sum_{a\in B\setminus D} a| \\ &= |t - \sum_{a\in A} a| \\ &= c(A) \end{aligned}$$

  portanto, $c(B) \leq c(A)$, logo $c(B) \leq max\{c(A), c(C)\}$.

- se $|t - \sum_{b \in B} b| \leq 0$, então:

$$\begin{aligned} c(B) &= |t - \sum_{b \in B} b| \\ &\leq |t - \sum_{b \in B} b - \sum_{e \in E} e| \quad \text{(pois } S \text{ contém apenas números positivos e } t - \sum_{b \in B} b \leq 0) \\ &= |t - \sum_{c \in B \cup E} c| \\ &= |t - \sum_{c \in C} c| \\ &= c(C) \end{aligned}$$

  portanto, $c(B) \leq c(C)$, logo $c(B) \leq max\{c(A), c(C)\}$.

Como acabamos de provar para os dois casos possíveis, temos que $c(B) \leq max\{c(A), c(C)\}$. □

2.3 O problema U-Curve

As duas funções de custo apresentadas na seção 2.2.1 descrevem curvas que tem um formato em U (a menos de oscilações) nas cadeias do reticulado booleano, vamos definir esta propriedade agora.

Definição 2.3.1. *Uma cadeia é dita* ***maximal*** *se não existe outra cadeia no reticulado que contenha propriamente esta cadeia.*

Definição 2.3.2. *Uma função de custo c é dita* ***decomponível em curvas U*** *se para toda cadeia maximal* $X_1, ..., X_l$, $c(X_j) \leq max\{c(X_i), c(X_k)\}$ *sempre que* $X_i \subseteq X_j \subseteq X_k$, $i, j, k \in \{1, ..., l\}$.

Vamos considerar então o problema de seleção de características em que a função de custo utilizada é decomponível em curvas U. Este é o problema central deste trabalho.

Definição 2.3.3 (Problema U-Curve)**.** *Dados um conjunto finito e não-vazio S e uma função de custo c decomponível em curvas em U, encontrar um subconjunto* $X \in \mathcal{P}(S)$ *tal que* $c(X) \leq c(Y)$, $\forall Y \in \mathcal{P}(S)$.

O problema U-Curve é um caso particular do problema de seleção de características com uma propriedade que nos permite achar o mínimo global sem a necessidade de avaliar cada ponto do reticulado booleano. Isso é possível porque a propriedade U-Curve (da decomponibilidade da função de custo em curvas U) nos garante que o custo dos elementos de uma cadeia não podem cair uma vez que aumentaram. Sejam por exemplo dois elementos $A \subseteq B$ de $\mathcal{P}(S)$, então:

- se $c(B) > c(A)$, então $c(X) > c(A)$ para todo X do intervalo $[B, \mathcal{P}(S)]$;
- se $c(A) > c(B)$, então $c(X) > c(B)$ para todo X do intervalo $[\emptyset, A]$;

Desta maneira, quando um problema de seleção de características tem uma função de custo decomponível em curvas U a menos de algumas oscilações, é vantajoso aproximar a solução deste problema pela solução encontrada por um algoritmo de busca do problema U-Curve. Tal abordagem não é ótima, porém, como existem poucas oscilações da função de custo, é provável que a solução encontrada ainda seja próxima da melhor solução.

Capítulo 3

O algoritmo Parallel U-Curve Search

O algoritmo `Parallel U-Curve Search (PUCS)` foi desenvolvido para resolver o problema U-Curve particionando o espaço de busca em partes que podem ser resolvidas independentemente e de forma paralela. Além disso, a dinâmica desse algoritmo depende de parâmetros que determinam o tempo de execução e qualidade da solução obtida, permitindo ao usuário adequar o algoritmo aos recursos computacionais disponíveis.

3.1 Princípios do algoritmo

3.1.1 Partição do espaço de busca

Seja S o conjunto de características do problema em questão. O primeiro passo do particionamento é escolher arbitrariamente S' um subconjunto de S; de maneira complementar, definimos $\overline{S'} = S \setminus S'$. Agora, sejam $X, Y \in \mathcal{P}(S)$ e $\sim$ a relação:

$$X \sim Y \iff (X \cap S') = (Y \cap S')$$

Esta relação é de equivalência, pois nela valem:

- reflexividade

$$X \sim X, \text{ pois } (X \cap S') = (X \cap S')$$

- simetria

$$\begin{aligned} X \sim Y &\iff \\ (X \cap S') = (Y \cap S') &\iff \\ (Y \cap S') = (X \cap S') &\iff \\ Y \sim X & \end{aligned}$$

- transitividade,

$$\begin{aligned} X \sim Y, Y \sim Z &\Rightarrow \\ (X \cap S') = (Y \cap S') = (Z \cap S') &\Rightarrow \\ (X \cap S') = (Z \cap S') &\Rightarrow \\ X \sim Z & \end{aligned}$$

Portanto, o conjunto das classes de equivalência definidas por $\sim$ é uma partição do espaço de busca original. Tome como exemplo o conjunto $S = \{a, b, c\}$; se $S' = a$, então existem duas classes de equivalência no particionamento do espaço de busca que definimos, formados pelos conjuntos $\{\emptyset, b, c, bc\}$ e $\{a, ab, ac, abc\}$.

Pela definição da relação $\sim$ temos que a presença de cada característica de S' em uma dada parte do reticulado não muda, isto é, ou ela está presente em todos subconjuntos da parte ou não está presente em nenhum, portanto, dizemos que estas variáveis são **fixas**. De modo análogo, as variáveis de $\overline{S'}$ são **livres**. Tanto variáveis fixas quanto livres podem definir reticulados booleanos junto a relação de ordem parcial $\subseteq$.

O conjunto $\mathcal{P}(S')$ induz um reticulado booleano em que cada elemento representa uma classe de equivalência do espaço de soluções do problema original, chamamos este de **reticulado externo**. Para cada classe de equivalência (nó do reticulado externo), o conjunto $\mathcal{P}(\overline{S'})$ induz um outro reticulado booleano (**reticulado interno**) em que cada elemento representa um subconjunto de problema original. Seja $A \in \mathcal{P}(S')$ um elemento do reticulado externo, então cada $B \in \mathcal{P}(\overline{S'})$ do reticulado interno em A representa o conjunto $X = B \cup A$ do espaço de busca do problema original.

Os reticulados internos e externo elucidam a estrutura recursiva do problema de seleção de características e sugerem que podemos construir uma solução ao problema original a partir de soluções de outros problemas, sobre os reticulados externo e internos, abordagem conhecida em computação como divisão e conquista. Seja $\langle S, c\rangle$ uma instância do problema de seleção de características, S' o conjunto de variáveis fixas, $\overline{S'}$ o conjunto de variáveis livres, e $A \in \mathcal{P}(S')$ um subconjunto que é nó do reticulado externo, então podemos definir um outro problema de seleção de características $\langle \overline{S'}, c_A\rangle$ em que

$$c_A(X) = c(X \cup A).$$

Resolver a instância $\langle \overline{S'}, c_A\rangle$ é essencialmente achar o mínimo do problema inicial restrito a classe de equivalência de A, dizemos também que estamos resolvendo a parte A. Se soubermos em qual classe o mínimo global reside, podemos resolver apenas tal parte e garantir que a solução encontrada é a solução do problema original.

O algoritmo `PUCS` percorre o reticulado externo recolhendo partes candidatas a conter o mínimo e resolve cada uma destas, escolhendo o mínimo das soluções destas partes como o mínimo global do problema.

3.1.2 Dinâmica do algoritmo

Para escolher partes candidatas a conter o mínimo global do problema, o algoritmo `PUCS` faz um passeio aleatório no reticulado externo. O algoritmo se inicia escolhendo arbitrariamente um nó inicial que pertence ao espaço de busca, então a cada passo as condições de podas são verificadas e escolhe-se aleatoriamente um vizinho do nó corrente que ainda pertence ao espaço de busca para se tornar o novo nó corrente. O algoritmo repete estes passos até que todos os nós do reticulado externo tenham sido ou podados ou visitados.

As podas eliminam do reticulado externo intervalos da forma $[X, \mathcal{P}(S')]$ ou $[\emptyset, X]$ e são realizadas sempre que a hipótese de curva em u implica que todas as partes contidas nestes intervalos não contém o mínimo global. Para entender o critério de poda, vamos definir que a **ponta superior** de um reticulado booleano $\mathcal{P}(A)$ é o próprio conjunto A e a **ponta inferior** deste reticulado é o conjunto vazio.

Definição 3.1.1 (Critério de poda para o reticulado externo do algoritmo PUCS)**.**

Capítulo 4

Conclusão

Bibliografia

[Bar+07] J. Barrera, R. M. Cesar-Jr, D.C. Martins-Jr, R.Z.N Vencio, E. F. Merino, M. M. Yamamoto, F. G. Leonardi, C. A. B. Pereira e H. A. Portillo. «Constructing probabilistic genetic networks of *Plasmidium falciparum* from dynamical expression signals of the intraerythrocytic development cycle.» Em: *Methods of Microarray Data Analysis* V (2007), pp. 11–26.

[DMJ06] R.M. Cesar-Jr an J. Barrera D.C. Martins-Jr. «W-operator window design by minimization of mean conditional entropy». Em: *Patter Analysis & Applications* (2006), pp. 139–153.

[JCJB04] D. C. Martins Jr, R. M. Cesar-Jr e J. Barrera. «W-operator window design by maximization of training data information». Em: *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium* (2004), pp. 162–169.

[Rei12] M. S. Reis. «Minimization of decomposable in U-shaped curves functions defined on poset chains – algorithms and applications». Tese de doutoramento. University of Sao Paulo, 2012.