4
Algoritmos para grandes conjuntos de dados

Tradicionalmente, o algoritmo para regressão PLS é utilizado em conjuntos pequenos contendo uma centena de amostras. Isto é comum na área de quimiometria pois muitas vezes a geração dos dados juntamente com a variável dependente de interesse é custosa [14].

Com o objetivo de propor ferramentas adequadas para o tratamento de grandes conjuntos de dados, são descritos neste capítulo dois algoritmos oferecendo melhor desempenho computacional quando comparados com o algoritmo clássico. O primeiro é PPLS[1] [37, 41, 40], uma versão paralela do algoritmo PLS1, e o segundo é DPLS[2] [37, 41, 36], uma versão aproximada PLS2 do algoritmo, para o caso de mais de uma variável dependente.

4.1
PPLS

Para o caso PLS1, de apenas uma variável dependente, o modelo de regressão pode ser calculado com o algoritmo paralelo PPLS [37, 41, 40]. Outras formulações da regressão PLS baseadas em aprendizagem Hebbiana [19], possuem uma arquitetura naturalmente paralelizável, mas dependem de uma regra de convergência para o cálculo dos fatores.

Aproveitando o fato de o cálculo do autovetor (equação 2-1) ser exato para cada iteração algoritmo PLS1 ser exato, PPLS segue a mesma abordagem não dependendo de critérios de aproximação ou convergência. Isto permitiu um bom desempenho com uma reduzida massa de dados.

4.1.1
Algoritmo

Dado que a fase de treinamento é responsável pelo consumo da maioria do tempo dentro do processo de treinamento e predição na regressão PLS,

[1] Parallel PLS.
[2] Direct PLS.

apenas sua paralelização é fornecida. Além disto, os conjuntos de dados normalmente usados na fase de predição são pequenos quando comparados com os de treinamento, e desta forma o algoritmo seqüencial pode ser usado sem prejuízo.

Dadas $\gamma+1$ máquinas, denominadas de nós 0 a γ, e n amostras (x_i, y_i) $1 \leq i \leq n$ com $\mathsf{x}_i \in \mathbb{R}^m$ e $y_i \in \mathbb{R}$, queremos calcular o modelo linear para a regressão de Y em X. O nó 0 possui um papel distinto, o de coordenar a comunicação e repassar os resultados parciais para os nós 1 a γ.
A matrizes X e o vetor Y são definidos da seguinte forma:

$$\mathsf{X} = \begin{bmatrix} \mathsf{x}_1^\top \\ \vdots \\ \mathsf{x}_n^\top \end{bmatrix} \text{ e } \mathsf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Em cada um dos γ nós, n/γ amostras são armazenadas após a partição de X e Y nos seguintes blocos:

$$\mathsf{X} = \begin{bmatrix} \mathsf{X}_1 \\ \mathsf{X}_2 \\ \vdots \\ \mathsf{X}_\gamma \end{bmatrix} \text{ e } \mathsf{Y} = \begin{bmatrix} \mathsf{Y}_1 \\ \mathsf{Y}_2 \\ \vdots \\ \mathsf{Y}_\gamma \end{bmatrix}$$

onde os blocos X_γ e Y_γ são armazenados no nó γ. O cálculo do conjunto $\{\mathsf{w}, b, \mathsf{p}\}$ para regressão, é conseguido combinando os dois algoritmos apresentados nas figuras 4.2 e 4.1. Eles descrevem o cálculo a ser realizado nos nós 1 a γ e o nó 0 respectivamente.

Após cada iteração nos códigos apresentados, o nó mestre 0 contém o conjunto $\{\mathsf{w}, b, \mathsf{p}\}$ correspondente à regressão para o primeiro fator. Para o cálculo do próximo, basta executar os mesmos passos novamente.

4.1.2 Desempenho

Para a análise do desempenho do PPLS, foram usadas duas medidas [5, 15]: *Speedup* e *Efficiency*. A primeira fornece uma estimativa do ganho realizado a cada máquina adicionada ao processamento, enquanto que a segunda indica o aproveitamento de cada processador.

Para os testes, foi usado um *cluster* formado por até 22 computadores IBM 400MHz com 32MB de RAM. A biblioteca MPI [35] com a implementação mpich v1.2.0 foi adotada para a comunicação entre as máquinas.

PLS Paralelo para o nó i

$\mathsf{R_i} = \mathsf{X_i^\top Y}$
envia $\mathsf{R_i}$ para o nó 0

recebe w do nó 0
$\mathsf{t_i} = \mathsf{X_i w}$
$\mathsf{b_i} = \mathsf{t_i^\top Y_i}$
$\mathsf{d_i} = \mathsf{t_i^\top t_i}$
envia $\mathsf{b_i}, \mathsf{d_i}$ para o nó 0

$\mathsf{p_i} = \mathsf{X_i^\top t_i}$
envia $\mathsf{p_i}$ para o nó 0

// Cálculo residual
recebe p do nó 0
recebe b do nó 0
$\mathsf{X_i} = \mathsf{X_i} - \mathsf{t_i p^\top}$
$\mathsf{Y_i} = \mathsf{Y_i} - \mathsf{b} \cdot \mathsf{t_i}$

Figura 4.1: Cálculo de$\{w, b, p\}$ para o nó i.

PLS Paralelo para o nó mestre 0

for $\mathtt{i} = 1$ to γ
  recebe $\mathsf{R_i}$ do nó $\mathtt{i}$
$\mathsf{R} = \sum_{\mathtt{i}=1}^{\gamma} \mathsf{R_i}$
$\mathsf{w} = \mathsf{R}/\|\mathsf{R}\|$

for $\mathtt{i} = 1$ to γ
  envia w para o nó $\mathtt{i}$
for $\mathtt{i} = 1$ to γ
  recebe $\mathsf{b_i}$ do nó $\mathtt{i}$
  recebe $\mathsf{d_i}$ do nó $\mathtt{i}$
$\mathsf{b} = \sum_{\mathtt{i}=1}^{\gamma} \mathsf{b_i} / \sum_{\mathtt{i}=1}^{\gamma} \mathsf{d_i}$

for $\mathtt{i} = 1$ to γ
  recebe $\mathsf{p_i}$ do nó $\mathtt{i}$
$\mathsf{p} = \sum_{\mathtt{i}=1}^{\gamma} \mathsf{p_i} / \sum_{\mathtt{i}=1}^{\gamma} \mathsf{d_i}$

// Cálculo residual
for $\mathtt{i} = 1$ to γ
  envia p para o nó $\mathtt{i}$
  envia b para o nó $\mathtt{i}$

Figura 4.2: Cálculo de$\{w, b, p\}$ para o nó 0.

Sendo que conjuntos com um número significativo de amostras é pouco comum em experimentos de quimiometria, foi empregada a serie D do conjunto Santa Fé para os experimentos. A cada 300 pontos da série foi gerada uma linha da matriz X e para a variável dependente Y foi usado o ponto seguinte. Desta forma tentamos descrever através da regressão PLS um ponto em função dos 300 últimos utilizando as matrizes X (5040×300) e Y (5040×1).

As figuras 4.3, 4.4 e 4.5 mostram o desempenho do PPLS para o cálculo dos 300 fatores.

4.1.3 Avaliação

Como podemos observar na tabela 4.1, conseguimos um *Speedup* de no mínimo 3 indo de 4 a 9 máquinas. Além disso, obtivemos um aproveitamento acima de 74% de cada processador para a mesma configuração, mostrando

Tabela 4.1: Desempenho do PPLS

Nós	Tempo em seg.	Speedup	Efficiency
1	111.778	1.000	1.000
2	60.545	1.846	0.923
3	44.646	2.503	0.834
4	37.312	2.995	0.748
5	33.410	3.345	0.669
6	31.038	3.601	0.600
7	29.773	3.754	0.536
8	29.395	3.802	0.475
9	32.852	3.402	0.378
10	38.129	2.931	0.293
12	43.800	2.552	0.232
14	51.025	2.190	0.182
15	55.143	2.027	0.155
18	67.158	1.664	0.118
20	75.224	1.485	0.099
21	79.316	1.409	0.088

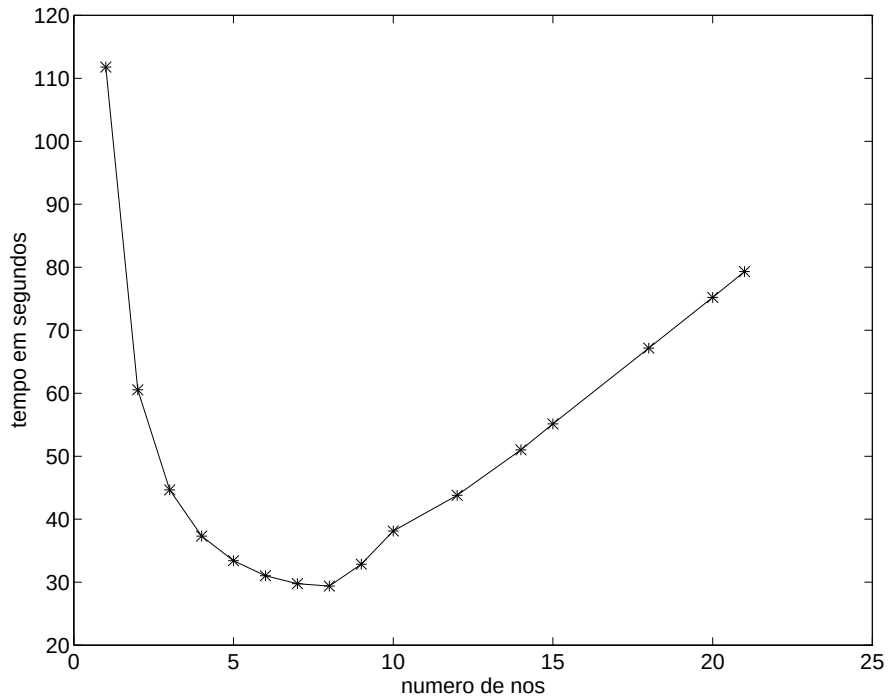

Figura 4.3: Tempo de execução do PPLS.

que a paralelização proposta é eficiente. Vale notar que estes resultados dependem do tamanho da massa de dados empregada. De fato, quanto maior for esta, melhor será o desempenho do algoritmo, pois a comunicação

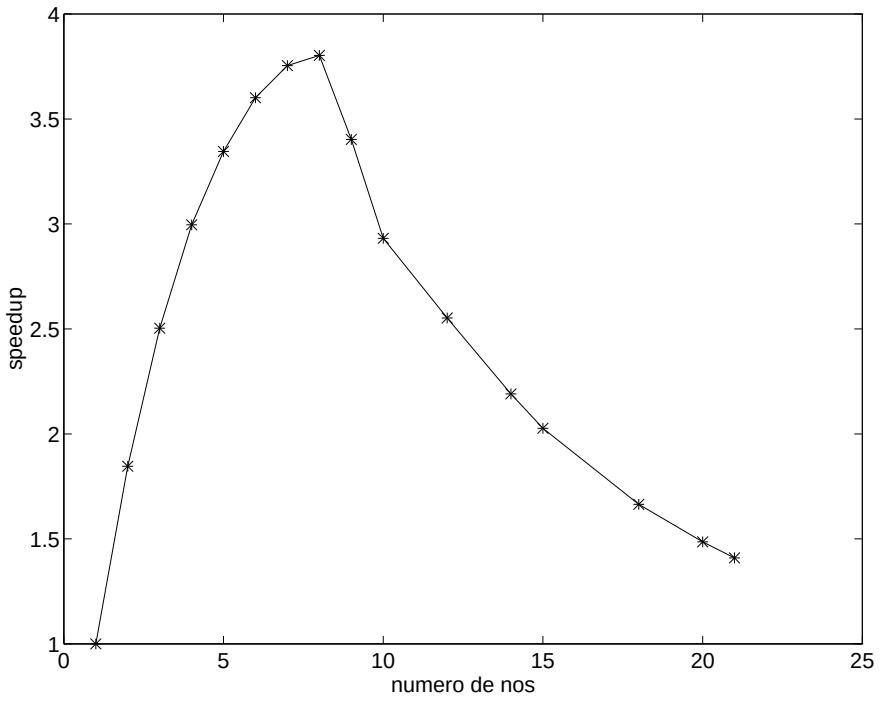

Figura 4.4: Speedup para o PPLS.

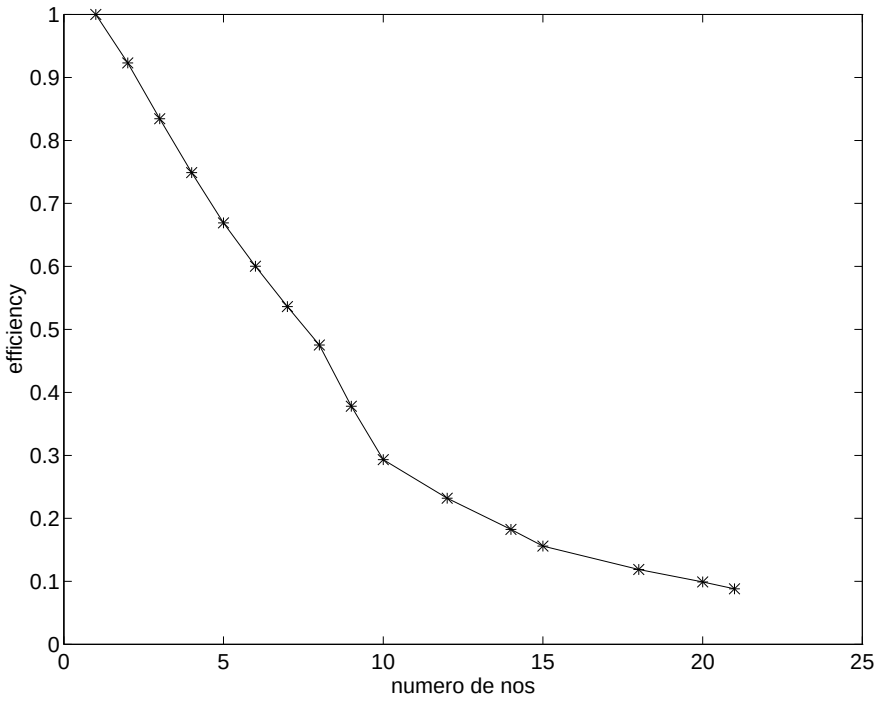

Figura 4.5: Efficiency para o PPLS.

entre os nós independe do número de amostras (número de linhas de X_i) em cada um. Dada a simplicidade e o baixo custo na implementação distribuída do PPLS, sua aplicação é indicada em ambientes com grandes volumes de dados.

A mesma abordagem paralela não é aplicada ao algoritmo PLS2 pois o cálculo do autovetor requer várias iterações até sua convergência. Isto acarreta num grande aumento na comunicação, tornando a paralelização computacionalmente ineficiente. Para o caso de mais de uma variável dependente, pode ser usado o algoritmo DPLS, apresentado na próxima seção.

4.2 DPLS

Quando a regressão PLS inclui mais de uma variável dependente, é necessário o cálculo dos autovetores da matriz composta pelas variáveis dependentes e independentes. Esta etapa é normalmente realizada com o algoritmo NIPALS [55, 52], mas podem ser usadas técnicas neurais não supervisionadas como aprendizagem Hebbiana, por exemplo [20, 9, 10]. O algoritmo DPLS[3] [37, 41, 36] fornece um método aproximado para este cálculo, possuindo uma modelagem competitiva e tempo de execução nitidamente superior como observado nas próximas seções.

4.2.1 Algoritmo

A cada iteração do algoritmo PLS2, o autovetor da matriz $\mathsf{X}^\top\mathsf{Y}\mathsf{Y}^\top\mathsf{X}$ associado ao maior autovalor deve ser calculado de forma iterativa. A diferença de DPLS com PLS2 está na substituição deste passo por um aproximado, porém não iterativo.

Seja G a matriz $\mathsf{X}^\top\mathsf{Y}\mathsf{Y}^\top\mathsf{X}$. Para cada conjunto de fatores, deve ser calculado o autovetor w de G associado ao maior autovalor.
Quando Y possui apenas uma variável, G possui posto igual a 1 e o autovetor não normalizado é simplesmente $\mathsf{X}^\top\mathsf{Y}$. Este autovetor corresponde ao único autovalor não nulo $(\mathsf{X}^\top\mathsf{Y})^\top(\mathsf{X}^\top\mathsf{Y})$. Por outro lado, isto deixa de ser válido para $l \geq 2$. Porém, a matriz G pode ser decomposta em blocos [49] da

[3] Direct PLS.

seguinte forma:

$$\mathsf{G} = \mathsf{X}^\top(\mathsf{Y}_1\mathsf{Y}_1^\top + \mathsf{Y}_2\mathsf{Y}_2^\top + \cdots + \mathsf{Y}_l\mathsf{Y}_l^\top)X$$

ou seja,

$$\mathsf{G} = \mathsf{X}^\top\mathsf{Y}_1\mathsf{Y}_1^\top\mathsf{X} + \cdots + \mathsf{X}^\top\mathsf{Y}_l\mathsf{Y}_l^\top\mathsf{X} \tag{4-1}$$

onde Y_i $(1 \leq i \leq l)$ corresponde a $i$ésima coluna de Y. Para uma notação mais simples escrevemos (4-1) como

$$\mathsf{G} = \mathsf{G}_1 + \mathsf{G}_2 + \cdots + \mathsf{G}_l$$

onde $\mathsf{G}_i = \mathsf{X}^\top\mathsf{Y}_i\mathsf{Y}_i^\top X$ para $i = 1, \ldots, l$.

Como podemos observar, o autovetor w de G associado ao maior autovalor é também autovetor das l matrizes conhecidas G_i de posto 1. Para o cálculo de w, métodos de potência baseiam-se na multiplicação de um vetor inicial aleatório por G, seguido de sua normalização até convergência. É razoável que dentre as matrizes G_i, o autovetor $\mathsf{w}_{(1)}$ daquela com maior autovalor $\lambda_{(1)}$, terá maior influência em w. Por outro lado, autovetores w_i possuindo pouca correlação com $\mathsf{w}_{(1)}$, vão contribuir pouco na direção de w. Com isto, uma simples aproximação para w pode ser obtida da seguinte forma:

1. Para cada $i = 1, \ldots, l$, calcular o autovetor w_i e o autovalor correspondente λ_i de G_i dados pelas relações

$$\mathsf{w}_i = \mathsf{X}^\top\mathsf{Y}_i \tag{4-2}$$

$$\lambda_i = \mathsf{w}_i^\top\mathsf{w}_i \tag{4-3}$$

2. Encontrar o autovetor $\mathsf{w}_{(1)}$ correspondente ao maior autovalor $\lambda_{(1)}$, definido por

$$\lambda_{(1)} = \max_{1 \leq i \leq l}\{\lambda_i\}$$

3. Calcular w dado por

$$\mathsf{w} = \sum_{i=1}^{l} \lambda_i\mathsf{w}_i(\mathsf{w}_i^\top\mathsf{w}_{(1)}) \tag{4-4}$$

e normalizar $\mathsf{w} \leftarrow \mathsf{w}/\|\mathsf{w}\|$.

A equação (4-4) oferece uma forma explícita para o cálculo de w, sem

que seja necessário nenhum critério de convergência como apresentado pelo algoritmo NIPALS.

Além disto, o algoritmo DPLS mostra uma qualidade competitiva na predição e uma nítida melhora na eficiência computacional, como relatado mais adiante na seção de experimentos.

4.2.2 Modelagem adaptativa

Dado que o algoritmo aproximado é baseado numa formulação explícita, é possível fornecer uma formulação adaptativa. De fato, para o passo 1, ao introduzir uma nova amostra juntamente com suas variáveis dependentes, ($\mathsf{x} \in \mathbb{R}^m, \mathsf{y} \in \mathbb{R}^l$), obtemos com a equação (4-2) o autovetor atualizado w'_i, dado por

$$\begin{aligned} \mathsf{w}'_i &= [\mathsf{X}^\top \; \mathsf{x}] \begin{bmatrix} \mathsf{Y}_i \\ y_i \end{bmatrix} \\ &= \mathsf{X}^\top \mathsf{Y}_i + \mathsf{x} y_i \\ &= \mathsf{w}_i + y_i \mathsf{x} \end{aligned} \tag{4-5}$$

e para o autovalor correspondente atualizado λ'_i, usando a equação (4-3) obtemos

$$\begin{aligned} \lambda'_i &= \mathsf{w}'^\top_i \mathsf{w}'_i \\ &= (\mathsf{w}_i^\top + y_i \mathsf{x}^\top)(\mathsf{w}_i + y_i \mathsf{x}) \\ &= \lambda_i + 2 y_i \mathsf{x}^\top \mathsf{w}_i + y_i^2 \mathsf{x}^\top \mathsf{x} \end{aligned} \tag{4-6}$$

Ambas as equações (4-5) e (4-6) requerem complexidade de tempo $O(m)$, fazendo com que seu uso seja indicado considerando o espaço necessário para armazenar todos os w_i e λ_i do modelo anterior.

Uma formulação adaptativa também pode ser fornecida para os passos 2 e 3, mas seria necessário um espaço considerável para armazenar os dados do modelo anterior, tornando o uso pouco prático.

4.2.3 Análise de complexidade

Sejam X e Y matrizes com dimensão $(n \times m)$ e $(n \times l)$ respectivamente. Para ambos PLS e DPLS, considerando o algoritmo clássico PLS2 [16, 52]

como apresentado na figura 2.1, o cálculo de cada conjunto de fatores a cada iteração pode ser decomposto nos dois passos:

1. calcular o autovetor w associado ao maior autovalor;

2. calcular o vetor de coeficientes de regressão b, as cargas p e as matrizes residuais.

Desta forma, a diferença de complexidade computacional entre PLS e DPLS reside apenas no primeiro passo.

Para a comparação é usado o algoritmo PLS2 convencional [52, 16, 21] com estrutura igual ao da figura 2.1. O cálculo do passo 1 é realizado com o algoritmo NIPALS, ou seja, um método de potência para extração do autovetor associado ao maior autovalor [17, 11]. Cada iteração deste método para a convergência, linhas 4 a 10 da figura 2.1, requer complexidade de tempo $O(nm+nl)$. Por outro lado, DPLS requer complexidade $O(nm+ml)$ para sua única iteração que fornece o autovetor aproximado. Na prática, l é pequeno ($l \leq 5$), fazendo com que $nm \gg ml$. Podemos concluir que no caso do NIPALS requerer apenas uma iteração, a complexidade de tempo de ambos os algoritmos PLS2 e DPLS é praticamente a mesma. Porém, esta situação não ocorre na prática, possuindo a mesma probabilidade de que o vetor aleatório inicial seja igual ao autovetor procurado. Normalmente são necessárias mais do que 10 iterações até a convergência, fazendo com que o algoritmo DPLS mostre um desempenho computacional bem superior.

De fato, como mostrado em [17], o erro obtido na k-ésima iteração do NIPALS para o autovetor estimado $\mathsf{w}^{(k)}$ é limitado por

$$|\sin(\theta_k)| \leq \tan(\theta_0) \left| \frac{\lambda_{(2)}}{\lambda_{(1)}} \right|^k$$

onde

$$cos(\theta_k) = \left| \mathsf{w}_{(1)}^{\top} \mathsf{w}^{(k)} \right|$$

e $\lambda_{(1)}$ e $\lambda_{(2)}$ correspondem ao maior e segundo maior autovalor. Como pode ser observado, dependendo da razão $|\lambda_{(2)}|/|\lambda_{(1)}|$, a taxa de convergência do NIPALS pode ser muito lenta [13], levando a um DPLS relativamente rápido.

4.2.4 Experimentos

Os nove conjuntos que compõem o conjunto NIR foram usados para avaliar o DPLS. Para cada um, foi comparado o PRESS obtido usando o PLS2 e o DPLS em várias rodadas. Para cada caso foi escolhido o número de fatores que fornecesse simultaneamente uma melhor predição e síntese do modelo. Todos os conjuntos foram centrados usando a média obtida com as amostras para treinamento.

Resultados

Ao comparar a curva PRESS dos modelos produzidos pela regressão PLS2 e DPLS, obtivemos resultados semelhantes. Para ilustrar o desem-

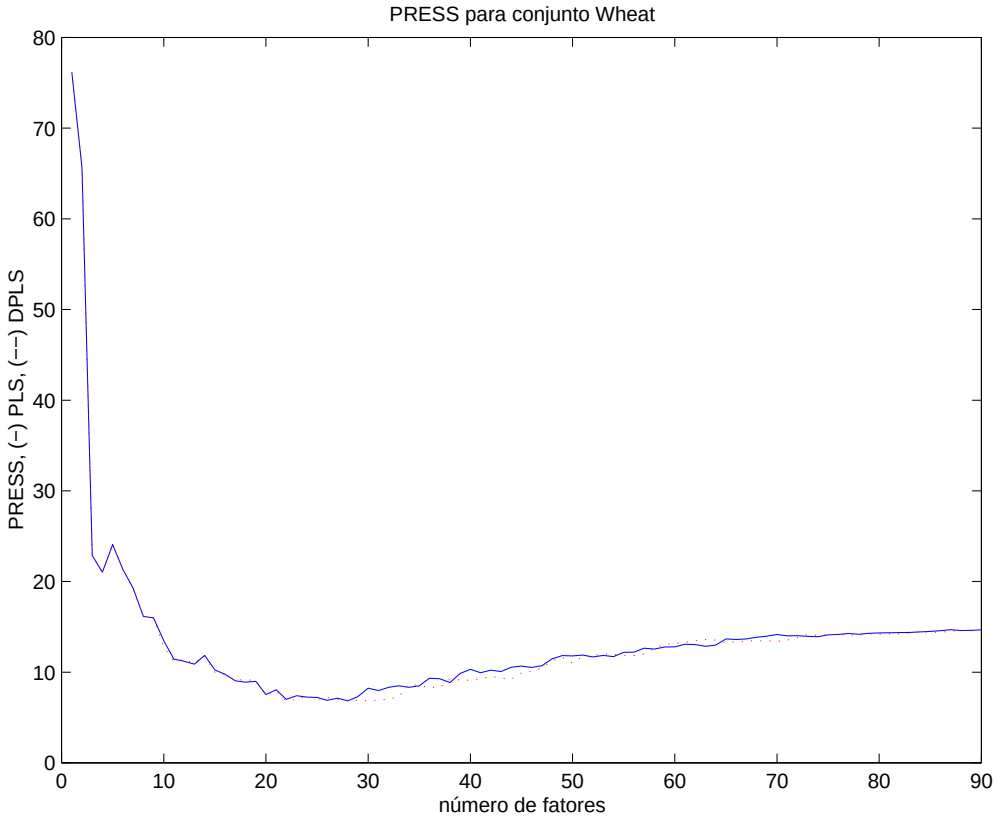

Figura 4.6: PRESS dos modelos PLS e DPLS para o conjunto Wheat.

penho geral do DPLS, a curva PRESS é mostrada para o primeiro conjunto de dados na figura 4.6. Para uma melhor análise, é mostrada na figura 4.7 a região crítica contendo o número de fatores escolhidos. Mesmo que para esta rodada o desempenho do DPLS tenha sido ligeiramente melhor, o interessante é a semelhança no comportamento das curvas. Com outros

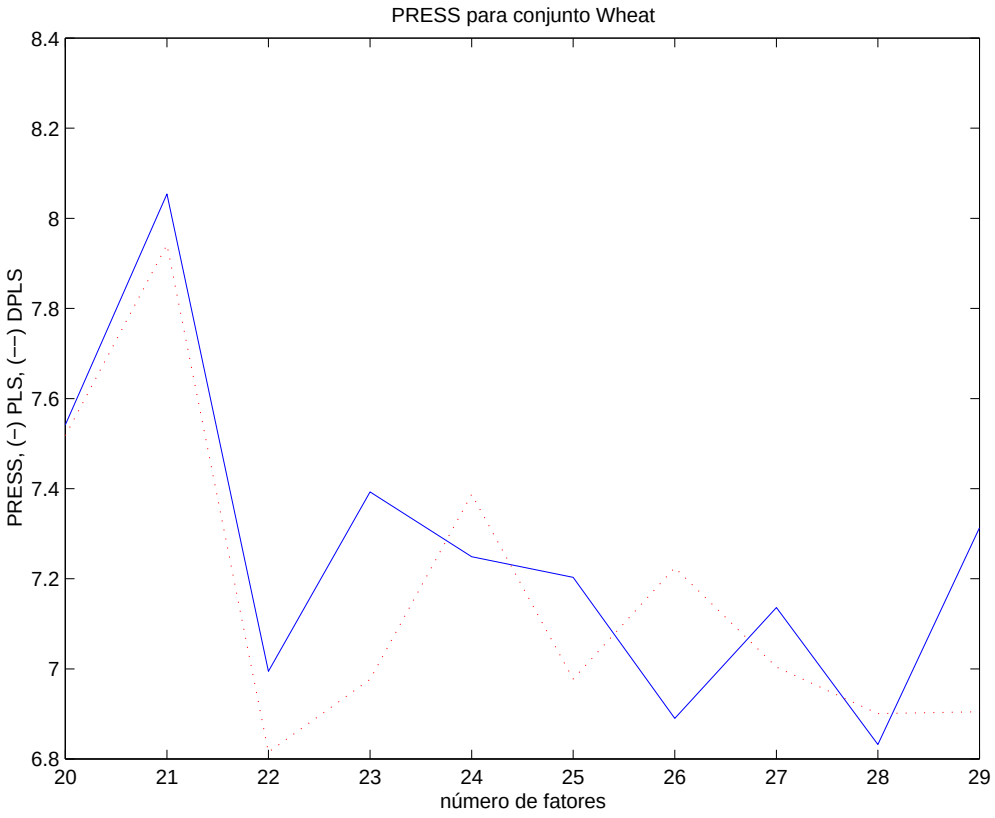

Figura 4.7: PRESS dos modelos PLS e DPLS para o conjunto Wheat, para os fatores escolhidos.

conjuntos de dados, as curvas de PRESS se confundem, como mostrado na figura 4.8 para o experimento com dados *Metal ions*. Outro exemplo do desempenho do DPLS é mostrado na figura 4.9. Como podemos observar, o modelo DPLS possui desempenho competitivo em relação ao PLS2. Na tabela 4.2 são mostrados os resultados obtidos com todos os conjuntos.

Tabela 4.2: PRESS e número de fatores escolhidos para o PLS e o DPLS

Conjunto	N. fatores		PRESS		PLS
	PLS	DPLS	PLS	DPLS	iterações
Wheat	22	22	6,619	6,601	25,7
Light gas oil	7	7	61,79	61,85	49,2
Combustible	5	5	201,88	202,26	11,9
Metal ions	49	46	$2,37 \cdot 10^{-5}$	$2,38 \cdot 10^{-5}$	39,9
Corn	9	9	4,736	4,719	39,4
Wet grass	8	8	123,27	121,90	64,3
Dry grass	14	14	33,90	33,93	47,7
Meat	15	15	897,34	892,49	41,1
Polymer	6	6	1,0650	1,0649	9,9

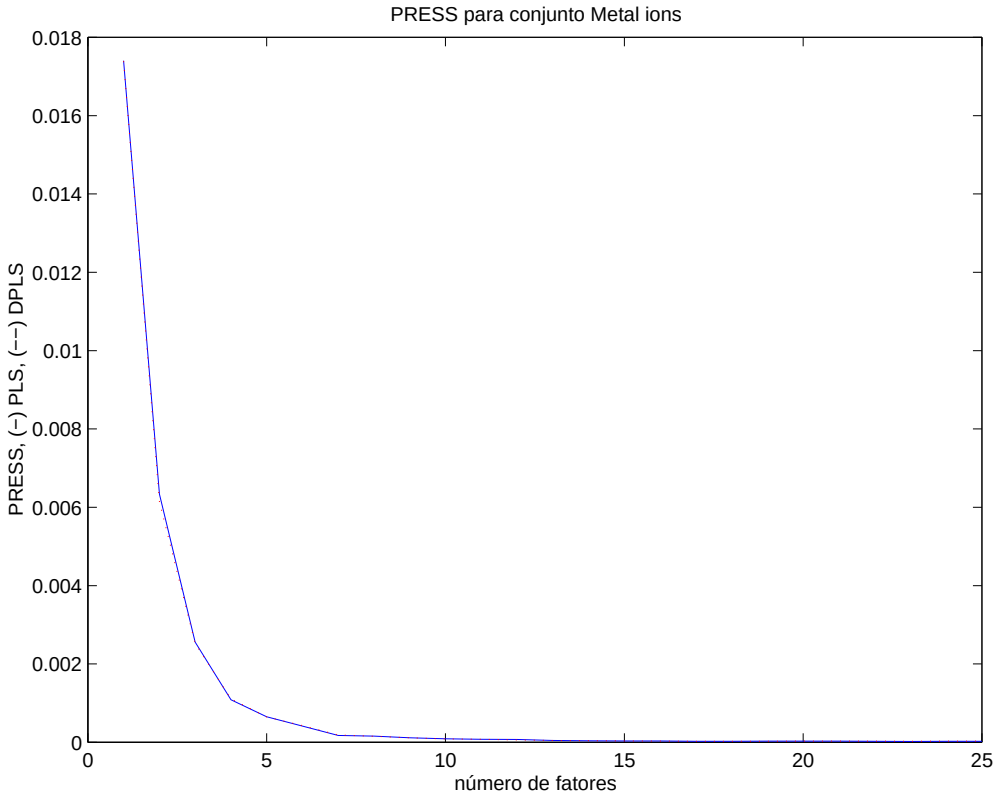

Figura 4.8: PRESS dos modelos PLS e DPLS para o conjunto Metal ions.

Cada conjunto foi sorteado 20 vezes, e na tabela foram reportadas amostragens representativas. No geral, o PRESS mínimo encontrado com ambos os algoritmos PLS2 e DPLS é praticamente o mesmo. A diferença relativa média observada foi de -0,93% a 1,38% para todos os conjuntos. Além disto, o desempenho computacional do DPLS ficou em média acima de 40% melhor do que o do PLS quando calculados apenas os 50 primeiros fatores. De fato, como comentado na seção 4.2.3, o número médio de iterações do algoritmo NIPALS do PLS2, em torno de 35 nos experimentos, faz com que o cálculo do autovetor no PLS2 seja muito mais custoso.

4.2.5 Comentários

Os resultados obtidos com o DPLS mostram que a aproximação no cálculo do autovetor da matriz $\mathbf{X}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{X}$ fornece um método alternativo para regressão por mínimos quadrados parciais. Além disto, o ganho em desempenho computacional faz deste algoritmo candidato para o tratamento de grandes conjuntos de dados para o caso de mais de uma variável dependente. Outra vantagem na eliminação do algoritmo iterativo para o

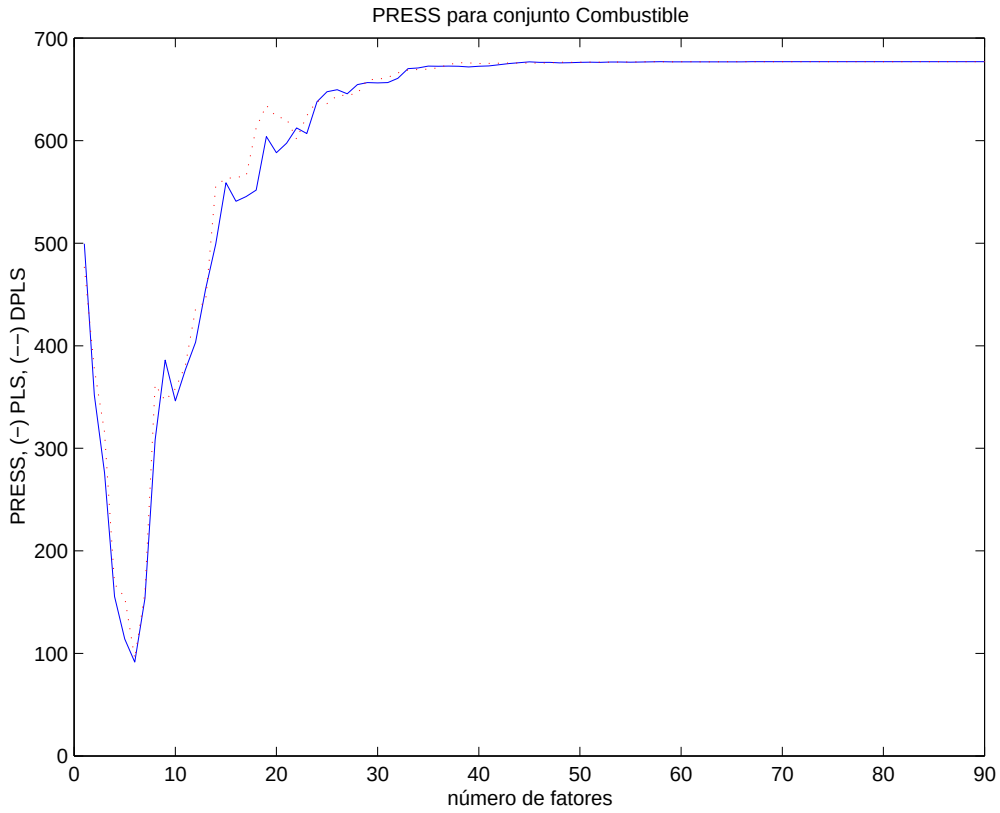

Figura 4.9: PRESS dos modelos PLS e DPLS para o conjunto Combustible.

cálculo do autovetor está na possibilidade de paralelização de forma eficiente seguindo a mesma abordagem do algoritmo PPLS da seção 4.1. Cada nó ao enviar R_i para o nó mestre, passa a enviar uma matriz $(m \times l)$, que então é usada para calcular a aproximação.