

## MÉTODOS DE PONTOS INTERIORES COMO ALTERNATIVA PARA ESTIMAR OS PARÂMETROS DE UMA GRAMÁTICA PROBABILÍSTICA LIVRE DO CONTEXTO

**Esther Sofía Mamián López**

Universidade Estadual de Campinas  
Rua Sérgio Buarque de Holanda, 651 – Campinas – Brasil  
esmamian@gmail.com

**Aurelio Ribeiro Leite de Oliveira**

Universidade Estadual de Campinas  
Rua Sérgio Buarque de Holanda, 651 – Campinas – Brasil  
aurelio@ime.unicamp.br

**Fredy Amaya Robayo**

Universidad del Cauca  
Calle 5 No. 4 - 70 – Popayán – Cauca – Colombia  
famaya@unicauca.edu.co

### RESUMO

Os modelos de linguagem são modelos matemáticos usados como componentes importantes em aplicações computacionais: reconhecimento automático da fala, reconhecimento ótico de caracteres, recuperação da informação, etc. O modelo de linguagem probabilístico mais usado é o modelo de n-gramas, baseado na frequência de ocorrência de uma cadeia em uma amostra da linguagem. Embora o modelo de n-gramas seja fácil de implementar e também seja um bom modelo para representar linguagens, quando aplicado a linguagens naturais ou a problemas de grande complexidade, apresenta dificuldades de interpretação. Uma alternativa para esses casos são os modelos probabilísticos baseados em gramáticas livres do contexto. Embora os mesmos apresentem dificuldade em problemas de grande complexidade devido ao alto custo computacional para estimar os parâmetros das gramáticas. Neste trabalho, vamos a apresentar resultados preliminares de métodos de pontos interiores primal-dual e método barreira logarítmica, como alternativa para estimar os parâmetros da gramática.

**PALAVRAS CHAVE:** Modelos de linguagem, gramáticas probabilísticas livres do contexto, métodos de pontos interiores.

**Área Principal:** Programação Matemática

### ABSTRACT

Language models are mathematical models used as important tools in computational applications: automatic speech recognition, optical character recognition, data recovery, etc. The most used probabilistic language model is the n-gram model, which is based in the word frequency of occurrence on a given language corpus. Despite the model n-gram is easy to implement and its several applications in language representation, its use in natural language and problems of greater complexity presents interpretation difficulties. An alternative for this case are probabilistic models using context-free grammar, though such models exhibit difficulties in complex programs since the computational cost estimation of parameters grammar is too high. This work presents preliminary results, for interior-point methods such as logarithmic barrier and primal-dual as an alternative for the parameters grammar estimation.

**KEYWORD :** Language models, probabilistic context-free grammar, interior point methods.

**Main area:** Mathematical Programming

## 1 Introdução

Um modelo probabilístico de uma linguagem (MPL) é um modelo matemático, onde é definida uma função de probabilidade que calcula a probabilidade de ocorrência de uma cadeia  $\chi$  em uma linguagem [3]. Os parâmetros de MPL (as probabilidades das cadeias) são aprendidos a partir de uma base de dados (amostra de cadeias) pertencentes à linguagem [5]. Geralmente a amostra é dividida em duas partes, a primeira para o processo de aprendizagem que é feito automaticamente. A segunda parte da amostra é usada para validar a qualidade do modelo obtido.

Vamos iniciar com os conceitos básicos que definem os modelos de linguagem: gramática formal, gramática livre do contexto e gramática probabilística livre do contexto [4]. A seguir vamos desenvolver o método de pontos interiores barreira logarítmica e o método de pontos interiores primal-dual, que são propostos para o problema de estimação dos parâmetros de uma gramática probabilística livre do contexto. Finalmente apresentamos os resultados preliminares e, usando a medida de perplexidade por palavra (PP) [6], fazemos a análise dos modelos propostos e resultados numéricos obtidos.

## 2 Gramáticas Livres do Contexto

Vamos definir os principais termos da teoria de linguagem formal. Alfabetos, cadeias e linguagem [3].

### Definição

- 2.1 Um alfabeto ou vocabulário, denotado por  $\Sigma$ , é um conjunto finito de símbolos.
- 2.2 Uma cadeia ou palavra é uma sequência finita de símbolos, pertencentes a um alfabeto  $\Sigma$ .
- 2.3 O tamanho de uma cadeia  $\alpha$  é o número de símbolos que a compõem, denotado por  $|\alpha|$ .
- 2.4 Uma cadeia  $\epsilon$  é dita vazia quando está constituída por zero símbolos ( $|\epsilon| = 0$ ).
- 2.5  $\Sigma^*$  apresenta o conjunto de todas as cadeias de um alfabeto  $\Sigma$ .
- 2.6  $\Sigma^+$  apresenta o conjunto de todas as cadeias de  $\Sigma$ , tal que seu tamanho é maior o igual a um.
- 2.7 Uma linguagem  $L$  sobre  $\Sigma$  é definida como um subconjunto do conjunto  $\Sigma^*$ .

### 2.1 Gramáticas e Gramáticas Livres do Contexto

Há quatro componentes importantes na descrição gramatical de uma linguagem:

- 2.8 Um conjunto finito  $\Sigma$  de símbolos que formam as cadeias da linguagem que está sendo gerada.
- 2.9 Um conjunto finito de variáveis ou não terminais, denotado por  $N$  com  $N \cap \Sigma = \emptyset$ .
- 2.10 Uma variável  $S$ , denominada variável de partida.

2.11 Um conjunto finito  $P$  de regras de derivação. Cada derivação tem a forma  $\alpha \rightarrow \gamma$ , onde  $\alpha$  (antecedente) e  $\gamma$  (consequente) são cadeias de símbolos de  $(N \cup \Sigma)^*$ . A expressão  $\alpha \rightarrow \gamma$  significa que a cadeia  $\alpha$  é substituída por  $\gamma$ . Segue um exemplo de regra:  $aAB \rightarrow baA$ .

**Definição 2.12** Uma gramática formal é uma 4-tupla  $G = (N, \Sigma, P, S)$ , onde  $N, \Sigma, P$  e  $S$  conforme definidos em 2.8 a 2.11.

**Definição 2.13** Uma gramática  $G$  é dita gramática livre do contexto (GLC), quando o conjunto  $P$  de regras de derivação está constituído por regras da forma  $A \rightarrow \alpha$  onde  $A \in N$  e  $\alpha \in (N \cup \Sigma)^+$ .

Neste trabalho vamos usar gramáticas na forma normal de Chomsky (FNC), ou seja gramáticas tais que seu conjunto  $P$  contém regras da forma  $A \rightarrow BC$  ou  $A \rightarrow v$  onde  $A, B, C \in N$  e  $v \in \Sigma$ . Como toda gramática livre do contexto é equivalente a uma gramática na forma normal de Chomsky [4], não há perda de generalidade.

**Definição 2.14** Sejam  $\gamma_1, \gamma_2 \in (N \cup \Sigma)^*$ , suponha que exista uma sequência de regras de derivação  $q_1, q_2, \dots, q_m \in P$  e cadeias  $\alpha_1, \alpha_2, \dots, \alpha_m \in (N \cup \Sigma)^*$ ,  $m \geq 0$  tal que

$$\gamma_1 = \alpha_1 \xrightarrow{q_1} \alpha_2 \xrightarrow{q_2} \dots \xrightarrow{q_m} \alpha_m = \gamma_2$$

onde  $\alpha_i \xrightarrow{q_i} \alpha_{i+1}$  significa que  $\alpha_{i+1}$  é derivado de  $\alpha_i$  usando a regra  $p_i$  uma única vez. Se diz que há uma derivação de  $\gamma_1$  em  $\gamma_2$ , denotada por  $\gamma_1 \xRightarrow{*} \gamma_2$ .

**Definição 2.15** A *Linguagem*  $L(G)$  gerada pela gramática  $G$  é o conjunto

$$L(G) = \{\chi \in \Sigma^+ : S \xRightarrow{*} \chi\}.$$

Uma cadeia pode ser obtida a partir de uma ou mais derivações.

## 2.2 Gramáticas Probabilísticas Livres-do-Contexto

Uma gramática probabilística livre do contexto (GPLC)  $G_p$  é um par  $(G, p)$  tal que  $G$  é gramática livre do contexto e  $p : P \rightarrow (0, 1]$  uma função definida nas regras da gramática que de acordo com [3]:

$$\forall A \in N, \sum_{(A \rightarrow \alpha) \in \Gamma_A} p(A \rightarrow \alpha) = 1 \quad (1)$$

com  $\Gamma_A \in P$  representa o conjunto de regras de derivação onde o antecedente é  $A$ . Vamos denotar a GPLC como  $G_p = (G, p)$ .

## 2.3 Probabilidade de uma Cadeia em uma GPLC

Seja  $G_p$  uma GPLC, para cada  $\chi \in L(G)$  denomina-se  $D_\chi$  o conjunto formado por todas as derivações  $d_\chi$  da cadeia  $\chi$ . Por  $N(q_i, d_\chi)$  designa-se o número de vezes em que a regra  $q_i \in P$  foi usada na derivação  $d_\chi$ .

**Definição 2.16** Dada uma gramática probabilística livre do contexto  $G_p$ , a probabilidade de uma derivação  $d_\chi$  da cadeia  $\chi \in \Sigma^*$  define-se como:

$$Pr(\chi, d_\chi | G_p) = \prod_{i=1}^{|\chi|} p(q_i)^{N(q_i, d_\chi)}. \quad (2)$$

**Definição 2.17** Dada uma gramática probabilística livre do contexto  $G_p$ , a probabilidade de ocorrência da cadeia  $\chi \in \Sigma^*$  é definida como:

$$Pr(\chi | G_p) = \sum_{d_\chi \in D_\chi} Pr(\chi, d_\chi | G_p). \quad (3)$$

## 2.4 Estimação dos Parâmetros de uma GPLC

Para abordar o problema de estimação das probabilidades da gramática, vamos definir uma função critério que depende de uma amostra da linguagem. Esta função critério é a função de verossimilhança da amostra [7].

Usando a teoria da inferência estatística usamos a técnica da máxima verossimilhança para estimar os parâmetros das probabilidades das regras.

Nas GPLCs a função de verossimilhança de uma amostra  $\Omega$  de  $L(G)$  denotada por  $Pr(\Omega | G_p)$  é definida assim:

$$Pr(\Omega | G_p) = \prod_{\chi \in \Omega} Pr(\chi | G_p). \quad (4)$$

Note que quando ordenamos as regras e definimos a probabilidade da  $i$ -ésima regra como uma variável  $x_i$  ( $p(q_i) = x_i$ ),  $i = 1, 2, \dots, |P|$ , obtemos um polinômio de várias variáveis em  $x$ .

No exemplo a seguir observamos como construir a função de verossimilhança para uma gramática dada:

**Exemplo 2.1** Seja  $G = (N, \Sigma, P, S)$ , onde  $N = \{A, B, C, S\}$ ,  $\Sigma = \{a, b\}$ ,  $S$  representa o símbolo inicial e  $P$  as regras assim definidas:

$$\begin{array}{llll} 1) S \rightarrow AB & 3) A \rightarrow BA & 5) B \rightarrow CC & 7) C \rightarrow AB \\ 2) S \rightarrow BC & 4) A \rightarrow a & 6) B \rightarrow b & 8) C \rightarrow a. \end{array}$$

A cada regra associamos uma probabilidade:

$$\begin{array}{llll} x_1 = p(S \rightarrow AB) & x_3 = p(A \rightarrow BA) & x_5 = p(B \rightarrow CC) & x_7 = p(C \rightarrow AB) \\ x_2 = p(S \rightarrow BC) & x_4 = p(A \rightarrow a) & x_6 = p(B \rightarrow b) & x_8 = p(C \rightarrow a). \end{array}$$

Seja  $\Omega = \{baaba, baaa\} \subset L(G)$  uma amostra da linguagem  $L$ . Sejam  $\chi = baaba$  e  $\eta = baaa$ . As árvores de derivação para as cadeias da amostra estão dadas pela Figura 1:

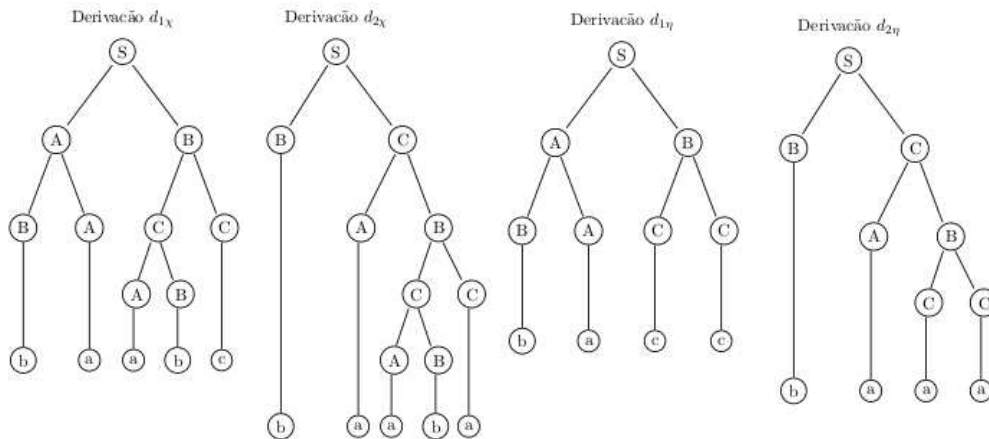


Figura 1: Árvores de derivação dos elementos de  $\Omega$

Regra	$D_\chi$		$D_\eta$		Probabilidade associada
	$N(d_{1\chi})$	$N(d_{2\chi})$	$N(d_{1\eta})$	$N(d_{2\eta})$	
$S \rightarrow AB$	1	0	1	0	$x_1$
$S \rightarrow BC$	0	1	0	1	$x_2$
$A \rightarrow BA$	1	0	1	0	$x_3$
$A \rightarrow a$	2	2	1	1	$x_4$
$B \rightarrow CC$	1	1	1	1	$x_5$
$B \rightarrow b$	2	2	1	1	$x_6$
$C \rightarrow AB$	1	2	0	1	$x_7$
$C \rightarrow a$	1	1	2	2	$x_8$

Tabela 1: Número de vezes que é usada cada regra nas derivações.

A Tabela 1 relaciona o número de vezes  $N(d_i)$  que a regra  $d_i$  está sendo usada.

Da equação (2) temos que:

$$\begin{aligned} \Pr(\chi, d_{1\chi}|G_p) &= x_1 x_3 x_4^2 x_5 x_6^2 x_7 x_8 & \Pr(\eta, d_{1\eta}|G_p) &= x_1 x_3 x_4 x_5 x_6 x_8^2 \\ \Pr(\chi, d_{2\chi}|G_p) &= x_2 x_4^2 x_5 x_6^2 x_7 x_8 & \Pr(\eta, d_{2\eta}|G_p) &= x_2 x_4 x_5 x_6 x_7 x_8^2, \end{aligned}$$

agora, usando (3), obtemos que:

$$\begin{aligned} \Pr(\chi|G_p) &= \Pr(\chi, d_{1\chi}|G_p) + \Pr(\chi, d_{2\chi}|G_p) = x_1x_3x_4^2x_5x_6^2x_7x_8 + x_2x_4^2x_5x_6^2x_7^2x_8 \\ \Pr(\eta|G_p) &= \Pr(\eta, d_{1\eta}|G_p) + \Pr(\eta, d_{2\eta}|G_p) = x_1x_3x_4x_5x_6x_8^2 + x_2x_4x_5x_6x_7x_8^2. \end{aligned}$$

Finalmente, da equação (4) temos a função de verossimilhança da amostra  $\Omega$ :

$$\begin{aligned} \Pr(\Omega|G_p) &= \Pr(\chi|G_p) * \Pr(\eta|G_p) \\ &= (x_1x_3x_4^2x_5x_6^2x_7x_8 + x_2x_4^2x_5x_6^2x_7^2x_8)(x_1x_3x_4x_5x_6x_8^2 + x_2x_4x_5x_6x_7x_8^2) \\ &= x_1^2x_3^2x_4^3x_5^2x_6^3x_7x_8^3 + x_1x_2x_3x_4^3x_5^2x_6^3x_7^2x_8^3 \\ &\quad + x_1x_2x_3x_4^3x_5^2x_6^3x_7x_8^3 + x_2^2x_4^3x_5^2x_6^3x_7x_8^3. \end{aligned}$$

Assim, do desenvolvimento anterior podemos concluir que a função  $Pr(\Omega|G_p)$  corresponde a um polinômio em várias variáveis, e que o problema de estimação dos parâmetros é equivalente a otimizar um polinômio sujeito a restrições lineares:

$$\begin{aligned} &\text{maximizar} && Pr(\Omega|G_p) \\ &\text{sujeito a} && \sum_{x_i \in \Psi_A} x_i = 1 \\ &&& 0 \leq x_i \leq 1, \quad i = 1, \dots, |P|, \end{aligned} \tag{5}$$

onde  $\Psi_A$  representa o conjunto de todas as probabilidades de regras cujo antecedente é A.

### 3 Método de Otimização

Nesta seção vamos apresentar os métodos que serão utilizados para resolver o problema de estimação dos parâmetros da GPLC formulado em (5): o método barreira logarítmica e o método de pontos interiores primal-dual.

#### 3.1 Método Barreira Logarítmica

Um problema de otimização com restrições lineares canalizado pode ser definido como:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && Ax = b \\ &&& 0 \leq x \leq u \end{aligned} \tag{6}$$

onde  $f(x) \in \mathbb{R}$ ,  $x \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$  e  $u \in \mathbb{R}^n$ .

Usando as variáveis  $v$  denominadas de folga, obtemos um problema equivalente ao problema (6)

$$\begin{aligned} & \text{minimizar} && f(x) \\ & \text{sujeito a} && Ax = b \\ & && x + v = u \\ & && x, v \geq 0. \end{aligned} \tag{7}$$

O método barreira logarítmica procura  $x^*$  que otimiza o problema (7). Este método aproxima o problema a resolver através de um outro problema equivalente definido por:

$$\begin{aligned} & \text{minimizar} && f(x) - \mu \left( \sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \ln(v_i) \right) \\ & \text{sujeito a} && Ax = b \\ & && x + v = u, \end{aligned} \tag{8}$$

onde  $\mu > 0$ .

O termo  $\sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \ln(v_i)$  é chamado de “barreira logarítmica” porque impede que as variáveis  $x_i$  e  $v_i$  se tornem negativas ou nulas. Obtemos soluções cada vez mais próximas de  $x^*$  quando  $\mu \rightarrow 0$  [8].

Para cada valor de  $\mu$  e usando as condições de primeira ordem ou condições de otimalidade de Karush-Kuhn-Tucker (KKT) do problema (8), temos que a solução  $x^*$  deve satisfazer [8]:

$$\begin{pmatrix} g(x) - \mu X^{-1}e - A^t y - t \\ t - \mu V^{-1}e \\ Ax - b \\ x + v - u \end{pmatrix} = 0, \quad (x, v) \geq 0 \tag{9}$$

onde  $g(x) = \nabla f(x)$ .  $y \in \mathbb{R}^m$  e  $t \in \mathbb{R}^n$  são os multiplicadores de Lagrange,  $X = \text{diag}(x_1, x_2, \dots, x_n)$  e  $V = \text{diag}(v_1, v_2, \dots, v_n)$ . As equações deste sistema pode ser visto como uma transformação linear  $F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$  definida como:

$$F(x, v, y, t) = \begin{pmatrix} g(x) - \mu X^{-1}e - A^t y - t \\ t - \mu V^{-1}e \\ Ax - b \\ x + v - u \end{pmatrix} = 0, \tag{10}$$

Dado que esta transformação é não linear, usamos o método de Newton para aproximar  $F$  a um modelo linear numa vizinhança de  $(x, v, y, t)$  [2]:



$$\begin{pmatrix} -H(x) - \mu X^{-2} & 0 & A^t & I \\ 0 & -\mu V^{-2} & 0 & I \\ A & 0 & 0 & 0 \\ I & I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \\ \Delta y \\ \Delta t \end{pmatrix} = \begin{pmatrix} rd \\ rb \\ rp \\ ru \end{pmatrix}, \quad (11)$$

onde  $\nabla^2 f(x) = H(x)$ ,  $r_d = g(x) - A^t y - t - \mu X^{-1}$ ,  $r_b = \mu e - VT$ ,  $r_p = b - Ax$  e  $r_u = u - x - v$ .  
Resolvendo este último sistema obtemos as direções de Newton  $(\Delta x, \Delta v, \Delta y, \Delta t)$ .

A atualização das variáveis é feita da seguinte forma:

$$(x^{k+1}, v^{k+1}, y^{k+1}, t^{k+1}) = (x^k, v^k, y^k, t^k) + \alpha(\Delta x, \Delta v, \Delta y, \Delta t).$$

Para determinar o tamanho de passo  $\alpha \in (0, 1]$  garantindo que os valores de  $x^{k+1}$  e  $v^{k+1}$  não sejam negativos usamos o teste da razão [8]:  $\alpha = \min(1, \tau\rho)$ , com  $\tau \in (0, 1)$ , onde  $\rho = \frac{-1}{\min_i(\frac{\Delta x_i}{x_i}, \frac{\Delta v_i}{v_i})}$ .

A seguir vamos resumir o método barreira logarítmica:

**Método 3.1** *Sejam  $(x^0, v^0, t^0) > 0$ ,  $y^0$  valores iniciais.*

*Para  $k = 1, 2, \dots$*

1. *Calcular o valor para  $\mu = \frac{\gamma}{n^2}$ , onde  $\gamma = s^t t$*
2. *Resolver o sistema linear (11) para  $\Delta x, \Delta v, \Delta y, \Delta t$ .*
3. *Determinar  $\alpha$  usando o teste da razão.*
4. *Atualizar o passo*

$$(x^{k+1}, v^{k+1}, y^{k+1}, t^{k+1}) = (x^k, v^k, y^k, t^k) + \alpha(\Delta x, \Delta v, \Delta y, \Delta t),$$

onde  $\alpha > 0$ .

*Fim\_Para.*

O critério de parada que vamos usar na implementação do método está baseado nas condições de otimalidade (9).

$$\left( \frac{\|r_d\|}{\max(1, \|y\|)}, \frac{\gamma}{n}, \frac{\|r_p\|}{\|b\| + 1}, \frac{\|r_u\|}{\|u\| + 1} \right) \leq \epsilon, \quad (12)$$

onde  $\epsilon > 0$ .

### 3.2 Método de Pontos Interiores Primal-Dual

Assim como no método barreira logarítmica, este método pode ser usado para resolver o problema (6). Observe que definindo  $\mu X^{-1}e = z \Leftrightarrow XZe = \mu e$  e reescrevendo  $\mu V^{-1}e = t \Leftrightarrow VTe = \mu e$  em (9) obtemos [1]:

$$\begin{pmatrix} A^t y - t + z - g(x) \\ VT e - \mu e \\ XZ e - \mu e \\ Ax - b \\ x + v - u \end{pmatrix} = 0, \quad (x, v, t, z) \geq 0. \quad (13)$$

Aplicando o método de Newton para resolver o sistema (13) obtemos:

$$\begin{pmatrix} -\nabla^2 f(x) & A^t & 0 & -I & I \\ 0 & 0 & T & V & 0 \\ Z & 0 & 0 & 0 & X \\ A & 0 & 0 & 0 & 0 \\ I & 0 & I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta v \\ \Delta t \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_d \\ r_b \\ r_c \\ r_p \\ r_u \end{pmatrix}, \quad (14)$$

onde  $r_d = g(x) - A^t y + t - z$ ,  $r_c = \mu e - XZ$  e  $r_b, r_p, r_u$  como no método barreira logarítmica. Resolvendo o sistema (14) obtemos as direções de Newton  $(\Delta x, \Delta y, \Delta v, \Delta t, \Delta z)$ .

Para garantir que as variáveis  $x^{k+1}, v^{k+1}, t^{k+1}, z^{k+1}$  não se tornem negativas quando o ponto é atualizado usamos o teste da razão:

$$\alpha = \min(1, \alpha_p, \alpha_d),$$

$$\text{onde,} \quad \alpha_p = \frac{-\tau}{\min_i(\frac{\Delta x_i}{x_i}, \frac{\Delta v_i}{v_i})} \quad e \quad \alpha_d = \frac{-\tau}{\min_i(\frac{\Delta z_i}{z_i}, \frac{\Delta t_i}{t_i})}, \quad (15)$$

para  $\tau \in (0, 1]$ .

A seguir vamos resumir o método de pontos interiores primal-dual:

**Método 3.2** *Sejam  $y^0, (x^0, v^0, t^0, z^0) > 0$  valores iniciais.*

*Para  $k = 1, 2, \dots$*

1. Escolher  $\mu = (\frac{\gamma}{4n^2})$ , onde  $\gamma = (x)^t z + (v)^t t$ .
2. Resolver o sistema linear (14) para  $\Delta x, \Delta v, \Delta y, \Delta t, \Delta z$ .
3. Determinar  $\alpha$  usando o teste da razão.
4. Atualizar o passo

$$(x^{k+1}, v^{k+1}, y^{k+1}, t^{k+1}, z^{k+1}) = (x^k, v^k, y^k, t^k, z^k) + \alpha(\Delta x, \Delta v, \Delta y, \Delta t, \Delta z),$$

onde  $\alpha > 0$

*Fim\_Para*

O critério de parada, está dado por (12).

## 4 Experimentos Numéricos

Implementamos, os algoritmos de pontos interiores primal-dual e barreira logarítmica desenvolvidos na Seção 3, usando o software matemático MatLab<sup>®</sup>, versão 7.8.0, em uma máquina com processador intel CORE i3 e memória ram de 3.7Gb, sobre o sistema operacional Linux 3.0.0-23-generic. Usamos duas gramáticas das quais obtemos uma amostra. A partir da amostra geramos uma função de verossimilhança que é maximizada com os métodos desenvolvidos na Seção 3. As características das gramáticas usadas estão especificadas na Tabela 2.

Gramática	Número de Terminais $ \Sigma $	Número de não terminais $ N $	Número de regras $ P $
Gramática 1	5	12	25
Gramática 2	14	13	47

Tabela 2: Característica das gramáticas

Para validar a qualidade dos modelos desenvolvidos a partir das gramáticas, a seguir vamos definir o conceito de perplexidade por palavra.

### 4.1 Perplexidade por Palavra

Esta medida é calculada usando outra amostra com um conjunto de dados que não foram usados no processo de aprendizagem, vamos denominá-lo conjunto de teste  $T_s$ . Quando o modelo está relacionado a uma GPLCs a perplexidade por palavra (PP) é definida como:

$$PP(T_s, G_p) = exp \left\{ - \frac{\sum_{\chi \in T_s} \ln(Pr(\chi|G_p))}{\sum_{\chi \in T_s} |\chi|} \right\} \quad (16)$$

Quanto menor for a perplexidade por palavra (PP), o modelo obtido tem maior capacidade expressiva, ou seja reconhece com boa probabilidade cadeias que não estejam nas amostras usadas para o processo de aprendizagem, comete menos erros.

### 4.2 Resultados Numéricos

Os valores e os parâmetros iniciais usados nos algoritmos, foram obtidos experimentalmente, garantindo a convergência dos métodos. Para todos os testes foi usada uma tolerância de  $10^{-8}$ .

Na Tabela 3 apresentamos os resultados numéricos obtidos para os dois métodos aplicados à Gramática 1, usamos uma amostra de tamanho 20 para a estimação das probabilidades e uma amostra ( $T_s$ ) de tamanho 30 para validar a qualidade do modelo obtido. Nas Tabelas 4 e 5 estão os resultados obtidos para os dois métodos aplicados à Gramática 2. Para a mesma usamos duas amostra de diferentes tamanhos: 20 e 80, e o tamanho da amostra ( $T_s$ ) são 40 e 130, respectivamente.

Nestas tabelas detalhamos o número de iterações e o tempo de processamento até convergir, assim como o valor da perplexidade.

Barreira Logarítmica			
Número de iterações	Tempo de processamento (seg)	Tamanho do conjunto Ts	perplexidade por palavra (PP)
13	0.396	30	3.096
Primal-Dual			
8	3.600	30	3.096

Tabela 3: Resultados da Gramática 1. Tamanho da amostra 20

Barreira Logarítmica			
Número de iterações	Tempo de processamento (seg)	Tamanho do conjunto Ts	perplexidade por palavra (PP)
33	4.828	40	12.455
Primal-Dual			
12	4.866	40	13.363

Tabela 4: Resultados da Gramática 2. Tamanho da amostra 20

Barreira Logarítmica			
Número de iterações	Tempo de processamento (seg)	Tamanho do conjunto Ts	perplexidade por palavra (PP)
15	2329	130	14.08
Primal-Dual			
8	21759	130	16.150

Tabela 5: Resultados da Gramática 2. Tamanho da amostra 80

## 5 Conclusões

O tratamento de grande quantidade de dados, principalmente para obter a Hessiana e o gradiente da função objetivo, assim como a busca do tamanho do passo, consome a maior parte do tempo por iteração das implementações.

Os resultados obtidos pelos métodos de pontos interiores resultaram em valores baixos da perplexidade por palavra. Os tempos de processamento são mais baixos que o exigido por outros métodos.

Os resultados preliminares em tempo de convergência, número de iterações e perplexidade por palavra, indicam que existem boas perspectivas para aperfeiçoamento dos métodos de pontos interiores

para a aplicação em gramáticas de maior porte tornando possível estudar linguagens naturais ou de maior complexidade.

## 6 Perspetivas Futuras

- Aperfeiçoar os métodos procurando uma alternativa para obter a Hessiana e o gradiente da função.
- Melhorar os métodos com técnicas de programação para o tratamento de grandes quantidades de dados.
- Implementar eficientemente os algoritmos na linguagem de programação C.

## Referências

- [1] A. EL-BAKRY, R. TAPIA, T. TSUCHIYA, AND Y. ZHANG, *On the formulation and theory of the newton point-pnterior for nonlinear programming*, Journal of Optimization Theory and Applications, (1996).
- [2] D. G. LUENBERGER AND Y. YE, *Linear and Nonlinear Programming*, Springer, terceira ed., 2008.
- [3] R. C. GONZALES AND M. G. THOMANSON, *Syntactic Pattern Recognition*, Addison-Wesley Publishing Company, 1978.
- [4] J. HOPCROFT, R. MOTWANI, AND J. D. ULLMAN, *Introduction to Automata Theory, Languages, And Computation*, Addison-Wesley, segunda ed., 2001.
- [5] M. P. MARCUS, B. SANTORINI, AND M. A. MARCINKIEWICZ, *Building a large annotated corpus of english: The penn treebank*, Computational Linguistics, 19 (1993), pp. 313–330.
- [6] J. A. S. PEIRÓ, *Estimación de Gramáticas Incontextuales Probabilísticas y su Aplicación en Modelación del Lenguaje*, PhD thesis, Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y computación, 1999.
- [7] F. A. A. ROBAYO, *Nuevas alternativas para la estimación de los parámetros en una gramática incontextual probabilística.*, proyecto de investigación, Universidad del Cauca. Departamento de Matemáticas, 2004.
- [8] S. J. WRIGHT, *Primal-Dual Interior-Point Methods*, Siam, 1997.