Esther Sofía Mamián López

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

CAMPINAS
2013

UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE MATEMÁTICA, ESTATÍSTICA
E COMPUTAÇÃO CIENTÍFICA

ESTHER SOFÍA MAMIÁN LÓPEZ

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

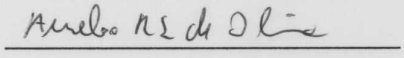
Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em matemática aplicada.

Orientador: Prof. Dr. Aurelio Ribeiro Leite de Oliveira

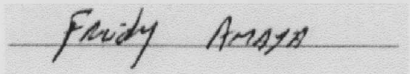
Coorientador: Prof. Dr. Fredy Ángel Miguel Amaya Robayo

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELA ALUNA ESTHER SOFÍA MAMIÁN LÓPEZ, E ORIENTADA PELOPROF. DR. AURELIO RIBEIRO LEITE DE OLIVEIRA.

Assinatura do Orientador



Assinatura do Coorientador



CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

M31m

Mamián López, Esther Sofía, 1985-
Métodos de pontos interiores como alternativa para estimar os parâmetros de uma gramática probabilística livre do contexto / Esther Sofía Mamián López. – Campinas, SP : [s.n.], 2013.

Orientador: Aurelio Ribeiro Leite de Oliveira.
Coorientador: Fredy Angel Amaya Robayo.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Gramáticas probabilísticas livres de contexto. 2. Linguagens formais . 3. Modelamento da linguagem. 4. Métodos de pontos interiores. I. Oliveira, Aurelio Ribeiro Leite de,1962-. II. Amaya Robayo, Fredy Angel. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Interior point methods as an alternative for estimating parameters of a stochastic context-free grammar
Palavras-chave em inglês:
Stochastic context-free grammars
Formal Languages
Language modeling
Interior-point methods
Área de concentração: Matemática Aplicada
Titulação: Mestra em Matemática Aplicada
Banca examinadora:
Aurelio Ribeiro Leite de Oliveira [Orientador]
Márcia Aparecida Gomes Ruggiero
Jair da Silva
Data de defesa: 07-10-2013
Programa de Pós-Graduação: Matemática Aplicada

Dissertação de Mestrado defendida em 07 de outubro de 2013 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Aurelio R L de Oliv

Prof.(a). Dr(a). AURELIO RIBEIRO LEITE DE OLIVEIRA

Márcia Gobbi

Prof.(a). Dr(a). MÁRCIA APARECIDA GOMES RUGGIERO

Jair da Silva

Prof.(a). Dr(a). JAIR DA SILVA

Abstract

In a probabilistic language model (PLM), a probability function is defined to calculate the probability of a particular string ocurring within a language. These probabilities are the PLM parameters and are learned from a corpus (string samples), being part of a language. When the probabilities are calculated, with a language model as a result, a comparison can be realized in order to evaluate the extent to which the model represents the language being studied. This way of evaluation is called perplexity per word. The PLM proposed in this work is based on the probabilistic context-free grammars as an alternative to the classic method inside-outside that can become quite time-consuming, being unviable for complex applications. This proposal is an approach to estimate the PLM parameters using interior point methods with good results being obtained in processing time, iterations number until convergence and perplexity per word.

Keywords: Stochastic context-free grammars, Formal language, Language modeling, Interior-point methods

Resumo

Os modelos probabilísticos de uma linguagem (MPL) são modelos matemáticos onde é definida uma função de probabilidade que calcula a probabilidade de ocorrência de uma cadeia em uma linguagem. Os parâmetros de um MPL, que são as probabilidades de uma cadeia, são aprendidos a partir de uma base de dados (amostras de cadeias) pertencentes à linguagem. Uma vez obtidas as probabilidades, ou seja, um modelo da linguagem, existe uma medida para comparar quanto o modelo obtido representa a linguagem em estudo. Esta medida é denominada perplexidade por palavra. O modelo de linguagem probabilístico que propomos estimar, está baseado nas gramáticas probabilísticas livres do contexto. O método clássico para estimar os parâmetros de um MPL (Inside-Outside) demanda uma grande quantidade de tempo, tornando-o inviável para aplicações complexas. A proposta desta dissertação consiste em abordar o problema de estimar os parâmetros de um MPL usando métodos de pontos interiores, obtendo bons resultados em termos de tempo de processamento, número de iterações até obter convergência e perplexidade por palavra.

Palavras-chave: Gramáticas probabilísticas livres do contexto, Linguagem formal, Modelamento da linguagem, Métodos de pontos interiores.

Sumário

Agradecimentos **ix**

Lista de Figuras **x**

Lista de Tabelas **xi**

Lista de Algoritmos **xii**

1 Introdução **1**

2 Gramáticas Livres do Contexto e Linguagem **4**
2.1 Introdução 4
2.2 Gramáticas e Gramáticas Livres do Contexto 6
2.3 Probabilidade de uma Cadeia de uma Gramática Probabilística Livre do Contexto 10
2.4 Estimação dos parâmetros de uma Gramática Probabilística Livre do Contexto 11
2.5 Função de verossimilhança 12
2.6 Algoritmo Inside-Outside 15
2.6.1 Teorema das Transformações Crescentes 16
2.6.2 Estratégia Inside 16
2.6.3 Estratégia Outside 17
2.6.4 Algoritmo Inside-Outside 18
2.7 Qualidade do modelo obtido 22

3 Métodos de Pontos Interiores **24**
3.1 O Problema 24
3.2 Os Problemas Primal e Dual na Forma Padrão 25
3.2.1 *Gap* de Dualidade 27
3.3 Método de Pontos Interiores Primal-Dual 28
3.4 Variante Preditor-Corretor 34
3.5 Método Barreira Logarítmica 38
3.6 Problema Canalizado 41
3.6.1 Método de Pontos Interiores Barreira Logarítmica 42

3.6.2 Método de Pontos Interiores Barreira Logarítmica Primal-Dual 46
3.7 Controle do Tamanho do Passo 49

4 Experimentos Numéricos **50**
4.1 Introdução 50
4.1.1 Problemas Testados 51
4.1.2 Estratégias Utilizadas na Implementação 53
4.2 Resultados Numéricos 55

5 Conclusões e Perspectivas futuras **62**
5.1 Conclusões 62
5.2 Perspectivas Futuras 63

Referência Bibliográfica **68**

Apêndice **69**

Agradecimentos

Esta dissertação não teria sido possível sem a orientação, apoio e incentivo, do professor Doutor Aurelio Ribeiro Leite de Oliveira e do professor Doutor Fredy Angel Amaya Robayo. Todo meu afeto e agradecimento.

Particularmente, é o resultado de Aydée López e Emiro Mamián, foram a primeira força amorosa e incentiva pela procura do conhecimento, o qual constrói homens e mulheres para a sociedade.

Agradeço a minhas irmãs Mónica, Laura e Angelica Mamián, ao meu irmão Juan Sebastian, pelo amor de sempre.

Juan Carlos, pelo constante apoio, amor e parceria nestes anos.

Agradeço aos amigos, colegas, professores do Instituto de Matemática, Estatística e Computação Científica, a construção do conhecimento sempre é um trabalho coletivo.

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior, pelo apoio financeiro.

Lista de Figuras

2.1 Árvores de derivação dos elementos de Ω 13

2.2 Árvore de derivação d_χ 21

4.1 Número de Iterações × tamanho amostra 58

4.2 Perplexidade por palavra × tamanho amostra 58

4.3 Tempo de convergência × tamanho amostra 59

5.1 Árvore de derivação da cadeia χ 69

5.2 Cálculo do termo e(A<1,k>)e(B<k+1,$|\chi|$>), onde $k = 1$ 71

5.3 Cálculo do termo e(A<1,k>)e(B<k+1,$|\chi|$>), onde $k = 2$ 71

5.4 Cálculo do termo e(A<1,k>)e(B<k+1,$|\chi|$>), onde $k = 3$ 72

5.5 Cálculo do termo e(A<1,k>)e(B<k+1,$|\chi|$>), onde $k = 4$ 72

5.6 Cálculo de $p(A \rightarrow a)$ 75

5.7 Cálculo de $f(A < i, i >)$, onde $i = 2$ 75

5.8 Cálculo de $f(A < 1, 2 >)$ 76

Lista de Tabelas

2.1 Número de vezes que está sendo usada cada regra nas derivações. 14

4.1 Parâmetros usados nas implementações 51

4.2 Características das gramáticas 51

4.3 Caraterísticas dos problemas - Gramática 1 52

4.4 Caraterísticas dos problemas - Gramática 2 53

4.5 Resultados Gramática 1 56

4.6 Resultados Gramática 2 57

Lista de Métodos

3.3.1 Primal-Dual 31
3.5.1 Barreira logarítmica 40
3.6.1 Barreira Logarítmica Canalizado 45
3.6.2 Primal-Dual Canalizado 48
3.7.1 Controle de Passo 49
3.7.2 Controle de Passo factibilidade 49

Capítulo 1

Introdução

Um modelo probabilístico de uma linguagem é um modelo matemático, onde é definida uma função de probabilidade que calcula a probabilidade de ocorrência de uma cadeia em uma linguagem [13]. Os parâmetros do modelo probabilístico de uma linguagem (as probabilidades das cadeias) são aprendidos a partir de uma base de dados (amostra de cadeias) pertencentes à linguagem [19]. Geralmente são usadas duas amostras, a primeira para o processo de aprendizagem e a segunda é usada para validar a qualidade do modelo obtido. Ambas são feitas automaticamente.

O modelo probabilístico da linguagem mais usado é o modelo de n-gramas [1], que está 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, quando aplicado a linguagens naturais ou a problemas de grande complexidade, apresenta problemas de interpretação, porque para calcular a probabilidade de uma palavra utiliza somente informação local (dada pelas n-1 anteriores). Por conseguinte, uma quantidade importante de informação não é considerada durante o processo de estimação, fazendo com que seja atribuído valor zero em frases com alta probabilidade.

Uma alternativa, que melhora o problema de interpretação do modelo de n-gramas, são os modelos probabilísticos baseados em gramáticas livres do contexto.

Uma gramática probabilística livre de contexto é uma gramática formal, composta de duas partes: estrutural e estocástica. Um conjunto de regras formam a componente estrutural da gramática, e as funções de distribuição de probabilidade associadas com as regras, são sua componente estocástica [13].

No processo de aprendizagem das gramáticas probabilísticas livres de contexto é usada uma amostra da gramática da qual espera-se recolher informações tanto da parte estrutural como de sua parte estocástica.

Para o processo de aprendizagem das probabilidades, em uma das abordagens, é necessário definir uma função critério que depende da amostra para otimizar. As amostras podem estar formadas por cadeias que sejam parte ou não da linguagem. Neste estudo as amostras são formadas por cadeias que pertencem à linguagem formal. Uma vez definida a função critério, por exemplo a função de verossimilhança, é escolhido um método de otimização para resolver o problema. Tais modelos apresentam dificuldade em problemas de grande complexidade devido ao alto custo computacional para estimar os parâmetros das gramáticas.

A função de verossimilhança de uma gramática probabilística livre de contexto é um polinômio de várias variáveis, e o algoritmo baseado no teorema de transformações crescentes, algoritmo clássico de estimação para encontrar o máximo desta função, tem um custo computacional alto [22, 4]. Desta forma, vamos analisar o desempenho do método de pontos interiores aplicado ao problema de maximizar os parâmetros da função de verossimilhança.

O método de pontos interiores primal-dual, a variante preditor-corretor e pontos interiores barreira logarítmica, são analisados para resolver esse problema. Os resultados obtidos entre as diferentes implementações estudadas são comparadas entre si. O objetivo consiste em reduzir o custo computacional, assim como proporcionar uma alternativa mais robusta para estimar os parâmetros da função de verossimilhança.

O restante desta dissertação está divida em quatro capítulos; o Capítulo 2 refere-se a uma revisão geral do conceito das gramáticas livres de contexto e linguagens. Os métodos de pontos interiores primal-dual, a variante preditor-corretor e pontos interiores barreira logarítmica são desenvolvidos no Capítulo 3. No quarto capítulo apresentamos experimentos numéricos realizados. O capítulo 5 contém as conclusões e as perspectivas futuras.

Capítulo 2

Gramáticas Livres do Contexto e Linguagem

2.1 Introdução

A origem da teoria da linguagem formal ocorreu na metade da década de 50, com o desenvolvimento de modelos matemáticos de gramáticas relacionadas com um trabalho baseado em linguagem natural, feito por Noam Chomsky [7]. Um dos propósitos nesta área foi desenvolver modelos computacionais de gramáticas capazes de descrever linguagens naturais, tais como Inglês, Português, etc. Fazendo isso, tinha-se a esperança que seria possível "ensinar" as máquinas a interpretar linguagens naturais.

As pesquisas nesta área, obtiveram resultados de impacto significativo em outros campos tais como: projeto de compiladores, linguagens de programação, teoria dos autômatos, reconhecimento de padrões, etc. Em anos posteriores, a utilização das linguagens formais nesta última área, proporcionaram sua aplicação na análise automática de caracteres alfanuméricos e da estrutura de cromossomos [26, 30].

Particularmente, na área de tradução automática existe uma importante aplicação: nos anos 60, havia uma grande esperança de que os computadores seriam capazes de fazer a tradução de uma linguagem natural para outra. Na última década, surgiu um movimento em direção a sistemas de tradução automática baseados em estudos estatísticos. Esta passou a denotar uma abordagem para todo problema de tradução que se baseia na descoberta da tradução mais provável de uma sentença, utilizando dados escolhidos em um corpus bilíngue.

Podemos expressar o problema de traduzir uma sentença E em inglês, para uma sentença F em francês pela aplicação da regra de Bayes a seguir [5, 24]:

$$\begin{aligned} \text{argmax}_F P(F|E) &= \text{argmax}_F \frac{P(E|F)P(F)}{P(E)} \\ &= \text{argmax}_F P(E|F)P(F). \end{aligned} \tag{2.1.1}$$

Essa regra diz que devemos considerar todas a sentenças possíveis em francês F e escolher aquela que maximiza o produto $P(E|F)P(F)$. O fator $P(F)$ é o modelo de linguagem para o francês; ele informa qual é a probabilidade de uma dada sentença estar em francês. $P(E|F)$ é o modelo de tradução; ele informa a probabilidade de uma sentença em inglês ser uma tradução, dada uma sentença em francês.

O modelo de linguagem $P(F)$ pode ser qualquer modelo que forneça uma probabilidade para uma sentença. Com um corpus muito grande, poderíamos estimar $P(F)$ diretamente, contando quantas vezes cada sentença aparece no corpus, mas surgem problemas onde grande parte das contagens podem ser zero. Portanto, uma das opções é usar o modelo de linguagem baseado nas gramáticas probabilísticas livres do contexto. O modelo de tradução $P(E|F)$ é mais difícil de obter. Por um lado, não temos uma coleção pronta de pares de sentenças (inglês, francês) a partir do qual possamos treinar, por outro lado a complexidade do modelo

é maior. Como o estudo do modelo de tradução não pertence ao tema desta dissertação, sugerimos as seguintes referências para estudar sobre este tópico [24, 5].

2.2 Gramáticas e Gramáticas Livres do Contexto

Entre as linguagens formais para processar linguagem natural (ou formal) estão aquelas geradas por gramáticas probabilísticas livres do contexto. As gramáticas probabilísticas livres de contexto geram modelos da linguagem com boa expressividade, ou seja, reconhecem com boa probabilidade cadeias que não estejam nas amostras usadas para o processo de aprendizagem. Desta forma, estas gramáticas cometem menos erros nesta tarefa em comparação a outras gramáticas [22, 25].

Os algoritmos propostos e usados na estimação dos parâmetros das gramáticas probabilísticas livres de contexto demandam uma grande quantidade de memória e o tempo de processamento é alto. Surgem então propostas para estudar outros métodos de solução para o problema de estimação das gramáticas probabilísticas livres de contexto.

Os conceitos necessários para abordar o problema de estimação das gramáticas probabilísticas livres de contexto são apresentados a seguir [14].

Definição 2.2.1.

(a) Um alfabeto ou vocabulário, denotado por Σ, é um conjunto finito de símbolos.

(b) Uma cadeia ou palavra é uma sequência finita de símbolos, pertencentes a um alfabeto Σ.

(c) O tamanho de uma cadeia χ é dado pelo número de símbolos que a compõem, denotado por $|\chi|$.

(d) Uma cadeia ϵ é dita vazia quando está constituída por nenhum símbolo, neste caso $|\epsilon| = 0$.

(e) Σ^* apresenta o conjunto de todas as cadeias de um alfabeto Σ.

(f) Σ^+ apresenta o conjunto de todas as cadeias de Σ, tal que seu tamanho é maior o igual a um.

(g) Uma Linguagem L sobre Σ é definida como um subconjunto do conjunto Σ^*.

Para descrever a gramática de uma linguagem, existem quatro componentes importantes:

- Um alfabeto Σ cujos símbolos são denominados símbolos terminais. Vamos usar letras minúsculas como a, b, c e d para representá-los.
- Um conjunto finito de variáveis ou símbolos não terminais, denotado por N com $N \cap \Sigma = \emptyset$. As letras maiúsculas A, B, C e D representam os símbolos não terminais.
- Uma variável S, denominada variável de partida. $S \in N$.
- Um conjunto finito P de regras de derivação. Cada derivação tem a forma $\alpha \longrightarrow \gamma$, onde α e γ são cadeias de símbolos de $(N \cup \Sigma)^*$. A expressão $\alpha \longrightarrow \gamma$ significa que a cadeia α é substituída por γ. Segue um exemplo de regra:

$$aAB \rightarrow baA.$$

Definição 2.2.2. Uma gramática formal é uma 4-tupla $G = (N, \Sigma, P, S)$, onde N, Σ, P e S estão acordo com os itens acima.

Definição 2.2.3. Sejam $\gamma_1, \gamma_2 \in (N \cup \Sigma)^*$, suponha que existe uma sequência de regras de derivação $q_1, q_2, ..., q_{m-1} \in P$ e cadeias $\alpha_1, \alpha_2, ..., \alpha_m \in (N \cup \Sigma)^*$, $m \geq 1$ tal que

$$\gamma_1 = \alpha_1 \overset{q_1}{\Rightarrow} \alpha_2 \overset{q_2}{\Rightarrow} ... \overset{q_{m-1}}{\Rightarrow} \alpha_m = \gamma_2$$

onde $\alpha_i \overset{q_i}{\Rightarrow} \alpha_{i+1}$ significa que α_{i+1} é derivado de α_i usando a regra q_i uma única vez. Se diz que há uma derivação de γ_1 em γ_2, denotada por $\gamma_1 \overset{*}{\Rightarrow} \gamma_2$.

Definição 2.2.4. Uma derivação à esquerda de uma cadeia $\chi \in \Sigma^+$ em G é uma sucessão de regras de derivação de $d_x = (q_1, q_2, \ldots, q_m)$, $m \geq 1$, tal que: $S = \alpha_1 \overset{q_1}{\Rightarrow} \alpha_2 \ldots \overset{q_{m-1}}{\Rightarrow} \alpha_m = \chi$, onde $\alpha_i \in (N \cup \Sigma)^+$, com $1 \leq i \leq m$ e q_i substitui o símbolo não terminal mais à esquerda de α_i.

A definição à direita de uma cadeia é equivalente. Neste trabalho vamos usar derivação à esquerda, somente. Logo, de agora em diante por derivação entenda-se como derivação à esquerda.

Definição 2.2.5. Uma gramática formal G, na qual o conjunto P de regras de derivação está constituído por regras da forma $A \longrightarrow \alpha$, onde $A \in N$ e $\alpha \in (N \cup \Sigma)^+$, denomina-se gramática livre do contexto.

Definição 2.2.6. Uma gramática[1] livre do contexto G está em forma normal de Chomsky (FNC), quando as regras de derivação estão na forma: $A \longrightarrow BC$ e $A \longrightarrow a$, onde $A, B, C \in N$ e $a \in \Sigma$.

Definição 2.2.7. A linguagem formal $L(G)$ gerada pela gramática formal G consiste no conjunto

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

[1]No restante desta disertação, quando falamos de gramática nos referimos a gramática formal.

Definição 2.2.8. Duas gramáticas livres do contexto G_1 e G_2 são equivalentes quando $L(G_1) = L(G_2)$.

Toda gramática livre do contexto é equivalente a uma gramática na forma normal de Chomsky [13]. Sem perda de generalidade, neste trabalho vamos usar gramáticas na forma normal de Chomsky.

Definição 2.2.9. Uma linguagem probabilística em um alfabeto Σ é um par (L, ϕ), com L uma linguagem formal e $\phi : \Sigma^* \longrightarrow \Re$ é uma função real nas cadeias de Σ^*. A função de probabilidade ϕ satisfaz:

1. $\chi \notin L$, então $\phi(\chi) = 0$ para todo $\chi \in \Sigma^*$.
2. $\chi \in L$, então $0 < \phi(\chi) \leq 1$ para todo $\chi \in \Sigma^*$.
3. $\sum_{\chi \in L} \phi(\chi) = 1$.

Definição 2.2.10. Uma gramática probabilística livre do contexto G_p é um par (G, p) tal que G é gramática livre do contexto e $p : P \longrightarrow (0, 1]$ uma função nas regras da gramática tal que:

$$\forall A \in N, \sum_{(A \longrightarrow \alpha) \in \Gamma_A} p(A \longrightarrow \alpha) = 1 \tag{2.2.1}$$

onde $\Gamma_A \subset P$ representa o conjunto de regras de derivação cujo antecedente é A.

Mais adiante, no Exemplo 2.5.1 ilustramos as definições e conceitos apresentados nas seções 2.2 e 2.3.

2.3 Probabilidade de uma Cadeia de uma Gramática Probabilística Livre do Contexto

Seja G_p uma gramática probabilística livre de contexto, para cada $\chi \in L(G)$ denomina-se D_χ o conjunto formado por todas as derivações d_χ da cadeia χ. Por $Nr(q_i, d_\chi)$ designa-se o número de vezes em que a regra q_i foi usada na derivação d_χ.

Definição 2.3.1. Dada uma gramática probabilística livre do contexto G_p, define-se a probabilidade de uma derivação d_χ da cadeia $\chi \in \Sigma^*$ como:

$$Pr(\chi, d_\chi|G_p) = \prod_{i=1}^{|P|} p(q_i)^{Nr(q_i,d_\chi)}. \tag{2.3.1}$$

Definição 2.3.2. Dada uma gramática probabilística livre do contexto G_p, define-se a probabilidade de uma cadeia $\chi \in \Sigma^*$ como:

$$Pr(\chi|G_p) = \sum_{d_\chi \in D_\chi} Pr(\chi, d_\chi|G_p). \tag{2.3.2}$$

Definição 2.3.3. Define-se a linguagem gerada por uma gramática probabilística livre do contexto $G_p = (G, p)$, como $L(G_p) = \{\chi \in L(G) | Pr(\chi|G_p) > 0\}$.

Definição 2.3.4. Uma gramática probabilística livre do contexto G_p denomina-se consistente, se a linguagem gerada por G_p é estocástica, ou seja:

$$\sum_{\chi \in L(G_p)} \phi(x) = 1. \tag{2.3.3}$$

2.4 Estimação dos parâmetros de uma Gramática Probabilística Livre do Contexto

Um dos problemas da teoria de modelagem da linguagem é conseguir representar uma linguagem formal ou natural usando as gramáticas.

No estudo das gramáticas probabilísticas livres do contexto são dois os grandes problemas para considerar: a aprendizagem, ou seja, obter uma gramática probabilística livre do contexto que represente uma linguagem e sua integração como modelo de interpretação em tarefas de grande complexidade.

Neste trabalho, somente vamos abordar o problema de aprendizagem das gramáticas probabilísticas livres do contexto. Para estudar sua integração em problemas de modelagem da linguagem, ver [22, 24]. Na aprendizagem das gramáticas probabilísticas livres do contexto, geralmente, são usados duas formas: dedutiva e indutiva. A primeira, é feita por um especialista humano, que conhece muito bem a linguagem que vai ser representada, sendo um trabalho muito complexo e inviável em tarefas reais. Na forma indutiva a gramática probabilística livre do contexto é automaticamente construída, usando uma amostra que contém elementos ou não da linguagem a representar, desta forma, é possível obter representações suficientemente apropriadas das linguagens desde que existam algoritmos robustos e amostras suficientemente representativas, embora o volume de dados seja muito grande.

Existem trabalhos em diferentes áreas que obtiveram bons resultados usando a aproximação indutiva: reconhecimento automático de fala [15], tradução automática [6] e modelação de sequências de DNA [9].

Para o processo de aprendizagem da gramáticas probabilísticas livres do contexto, alguns pesquisadores propõem decompor o processo: na aprendizagem da gramática característica (as regras de derivação) e aprendizagem das probabilidades das regras [28].

Nós vamos trabalhar com o problema de aprendizagem das probabilidades das regras, para isso, vamos definir uma função critério que depende de uma amostra da linguagem. Esta função critério é a função de verossimilhança da amostra.

2.5 Função de verossimilhança

Usando a teoria da inferência estatística usamos a técnica da máxima verossimilhança para estimar os parâmetros das probabilidades das regras [1].

Nas gramáticas probabilísticas livres do contexto a função de verossimilhança de uma amostra Ω de $L(G)$ é definida assim:

$$Pr(\Omega|G_p) = \prod_{\chi \in \Omega} Pr(\chi|G_p). \tag{2.5.1}$$

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

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

Exemplo 2.5.1. Seja $G = (N, \Sigma, P, S)$, onde $N = \{A, B, C, S\}$, $\Sigma = \{a, b\}$, S representa o símbolo inicial e P é o conjunto das regras assim definidas:

$$\begin{array}{lll} 1)\ q_1 = S \to AB & 4)\ q_4 = A \to a & 7)\ q_7 = C \to AB \\ 2)\ q_2 = S \to BC & 5)\ q_5 = B \to CC & 8)\ q_8 = C \to a. \\ 3)\ q_3 = A \to BA & 6)\ q_6 = B \to b & \end{array}$$

A cada regra associamos uma probabilidade:

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

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 2.1 e a Tabela 2.1 relaciona o número de vezes $Nr(q_i, d_j)$ que a regra q_i está sendo usada em cada derivação d_j.

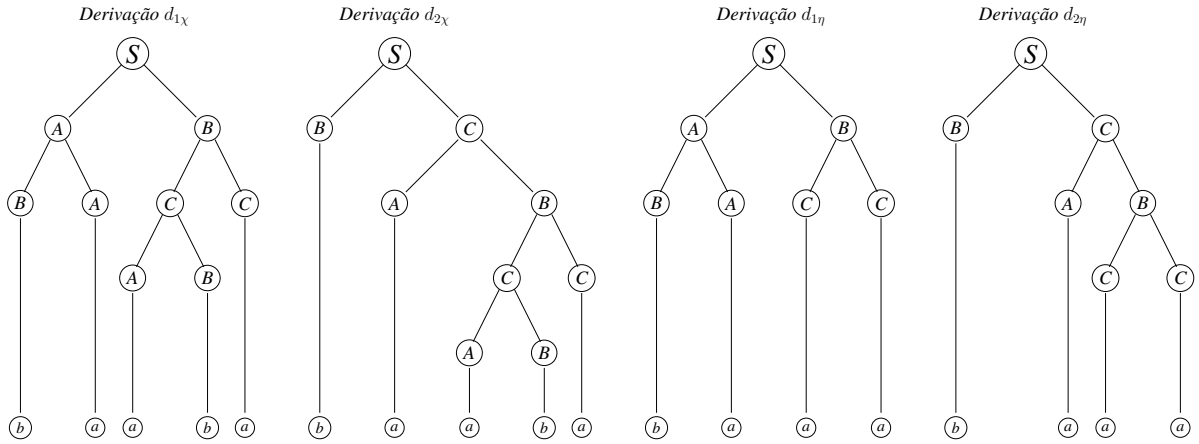


Figura 2.1: Árvores de derivação dos elementos de Ω

Da equação (2.3.1) temos que:

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

Regra	D_χ		D_η		Probabilidade associada
	$Nr(q_i, d_{1\chi})$	$Nr(q_i, d_{2\chi})$	$Nr(q_i, d_{1\eta})$	$Nr(q_i, d_{2\eta})$	
$q_1 = S \to AB$	1	0	1	0	x_1
$q_2 = S \to BC$	0	1	0	1	x_2
$q_3 = A \to BA$	1	0	1	0	x_3
$q_4 = A \to a$	2	2	1	1	x_4
$q_5 = B \to CC$	1	1	1	1	x_5
$q_6 = B \to b$	2	2	1	1	x_6
$q_7 = C \to AB$	1	2	0	1	x_7
$q_8 = C \to a$	1	1	2	2	x_8

Tabela 2.1: Número de vezes que está sendo usada cada regra nas derivações.

agora, usando (2.3.2), 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 (2.5.1) temos a função de verossimilhança da amostra Ω:

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

Assim, conforme visto anteriormente e lembrando a Definição 2.5.1, podemos concluir que a função $Pr(\Omega|G_p)$ corresponde a um polinômio em várias variáveis, não linear e que o problema de estimação dos parâmetros é equivalente a otimizar um polinômio sujeito a restrições

lineares:

$$\begin{aligned} \text{maximizar} \quad & Pr(\Omega|G_p) \\ \text{sujeito a} \quad & \sum_{x_i \in \Psi_A} x_i = 1, \quad \forall\ \Psi_A : A \in N \\ & 0 \leq x_i \leq 1, \qquad i = 1, ..., |P|, \end{aligned} \tag{2.5.2}$$

onde Ψ_A representa o conjunto de todas as probabilidades de regras cujo antecedente é A.

No caso particular do exemplo o problema consiste em:

$$\begin{aligned} \text{maximizar} \quad & x_1^2 x_3^2 x_4^3 x_5^2 x_6^3 x_7 x_8^3 + x_1 x_2 x_3 x_4^3 x_5^2 x_6^3 x_7^2 x_8^3 \\ & + x_1 x_2 x_3 x_4^3 x_5^2 x_6^3 x_7^2 x_8^3 + x_2^2 x_4^3 x_5^2 x_6^3 x_7^3 x_8^3 \\ \text{sujeito a} \quad & x_1 + x_2 = 1 \\ & x_3 + x_4 = 1 \\ & x_5 + x_6 = 1 \\ & x_7 + x_8 = 1 \\ & 0 \leq x_i \leq 1, \qquad\qquad\qquad\qquad i = 1, ..., 8 \end{aligned} \tag{2.5.3}$$

Observe que a matriz das restrições $E_{m \times n}$ é sempre uma matriz na forma de escada, onde $m \leq |P|$ e $n = |P|$, no exemplo $n = 8$. Portanto, as dimensões da matriz E não variam quando modificamos a amostra da linguagem.

2.6 Algoritmo Inside-Outside

Vamos apresentar um algoritmo clássico de estimação no problema de aprendizagem das gramáticas probabilísticas livres do contexto. O algoritmo Inside-Outside [17] está baseado no teorema de transformações crescentes [2] e nos algoritmos Inside e Outside descritos a seguir.

2.6.1 Teorema das Transformações Crescentes

Vamos iniciar com o teorema das transformações crescentes que, como foi dito, é a base principal do algoritmo Inside-Outside.

Teorema 2.6.1. Seja $F(x) = F(\{x_{ij}\})$ um polinômio homogêneo com coeficientes não negativos e de grau d em suas variáveis $\{x_{ij}\}$. Seja $x = \{x_{ij}\}$ um ponto do domínio $D = \{x_{ij} \geq 0 \Big| \sum_{j=1}^{q_i} x_{ij} = 1, i = 1, \ldots, p, j = 1, \ldots, q_i\}$. Para um dado $x = \{x_{ij}\} \in D$, o ponto $J(x) = J(\{x_{ij}\})$ em D, está definido em cada i, j como:

$$J(x)_{ij} = \frac{\left(x_{ij} \frac{\partial F}{\partial x_{ij}}\Big|_{(x)}\right)}{\sum_{j=1}^{q_i} x_{ij} \frac{\partial F}{\partial x_{ij}}\Big|_{(x)}}. \tag{2.6.1}$$

Então $F(J(x)) > F(x)$, exceto se $J(x) = x$.

Se $F(x)$ possui uma valor máximo finito, então é possível mostrar que toda sequência de pontos gerados pela transformação (2.6.1) converge para um máximo local de $F(x)$. Este resultado e a demostração deste teorema podem ser vistos em [2].

Usando o Teorema 2.6.1 em um polinômio F cujo domínio seja D e usando um algoritmo de máxima subida, é possível obter um máximo local de F.

2.6.2 Estratégia Inside

O algoritmo Inside é usado em problemas de análises sintática probabilística de cadeias, ou seja, determinar $Pr(\chi|G_p) > 0$. Para resolver este problema é suficiente encontrar uma derivação d_χ cuja probabilidade seja maior que zero e que usando d_χ seja possível derivar a cadeia χ a partir do símbolo inicial S ($S \stackrel{*}{\Rightarrow} \chi$).

Este algoritmo determina se $Pr(\chi|G_p) > 0$ calculando a probabilidade da cadeia a partir de todas as possíveis derivações. Para isso, define: $e(A < i, j >) = Pr(A \stackrel{*}{\Rightarrow} \chi_i, \ldots, \chi_j|G_p)$, que é a probabilidade de que a sub-cadeia $\chi_i, \ldots, \chi_j$ seja gerada a partir de $A \in N$. Para calcular $e(A < i, j >)$ são usadas as fórmulas a seguir:

$$e(A < i, i >) = p(A \longrightarrow \chi_i), \ 1 \leq i \leq |\chi| \tag{2.6.2a}$$

$$e(A < i, j >) = \sum_{B,C \in N} p(A \longrightarrow BC) \sum_{k=i}^{j-1} e(B < i, k >) e(C < k+1, j >), \tag{2.6.2b}$$

$$1 \leq i < j \leq |\chi|,$$

onde $A \longrightarrow \chi_i$, $A \longrightarrow BC \in P$.

Finalmente, $Pr(\chi|G_p) = e(S < 1, |\chi| >)$. O algoritmo tem complexidade $O(|\chi|^3|P|)$.

2.6.3 Estratégia Outside

Da forma semelhante ao algoritmo Inside, este algoritmo determina se $Pr(\chi|G_p) > 0$ a partir de todas as possíveis derivações.

Define-se $f(A < i, j >) = Pr(S \stackrel{*}{\Rightarrow} \chi_1 \ldots \chi_{i-1} A \chi_{j+1} \ldots \chi_{|\chi|}|G_p)$, como a probabilidade de que a partir do símbolo inicial seja gerada a sub-cadeia $\chi_1 \ldots \chi_{i-1}$, depois o símbolo não terminal A e em seguida a sub-cadeia $\chi_{j+1} \ldots \chi_{|\chi|}$. Note que o símbolo não terminal A tem que gerar

a sub-cadeia $\chi_i \dots \chi_j$. Para calcular $f(A < i, j >)$, são usadas a seguintes fórmulas:

$$\forall A \in N$$

$$f(A < 1, |\chi| >) = \begin{cases} 1 & \text{se } A = S \\ 0 & \text{se } A \neq S \end{cases} \tag{2.6.3a}$$

$$f(A < i, j >) = \sum_{B,C \in N} \left(p(B \longrightarrow CA) \sum_{k=1}^{i-1} f(B < k, j >) e(C < k, i-1 >) + p(B \longrightarrow AC) \sum_{k=j+1}^{|\chi|} f(B < i, k >) e(C < j+1, k >) \right) \tag{2.6.3b}$$

$$1 \leq i < j \leq |\chi|.$$

Logo, $Pr(\chi|G_p) = \sum_{A \in N} f(A < i, i >) p(A \longrightarrow \chi_i)$, $1 \leq i \leq |\chi|$. O cálculo das fórmulas acima tem complexidade $O(|\chi|^3 |P|)$.

Um exemplo para calcular a probabilidade de uma cadeia utilizando as estratégias Inside e Outside é dado no Apêndice desta dissertação.

2.6.4 Algoritmo Inside-Outside

O algoritmo Inside-Outside maximiza a função de verossimilhança definida na equação (2.5.1). Vamos iniciar o desenvolvimento do algoritmo notando que as probabilidades das regras de uma gramática probabilística livre do contexto, são pontos do domínio do Teorema 2.6.1. Para uma gramática probabilística livre do contexto G_p dada (não necessariamente em forma normal de Chomsky):

$$p(A_i \longrightarrow \alpha_j) = \{x_{ij}\}, \quad i = 1, \dots, |N| \quad j = 1, \dots, |\Gamma_A|$$

onde Γ_{A_i} representa o conjunto de regras da gramática G_p cujo antecedente é A_i. Portanto, a função de verossimilhança que está definida em termos destas probabilidades pode ser maximizada usando o Teorema 2.6.1. É possível definir uma transformação para cada $(A \longrightarrow \alpha) \in P$ como:

$$\overline{p}(A \longrightarrow \alpha) = \frac{p(A \longrightarrow \alpha)\left(\frac{\partial log Pr(\Omega|G_p)}{\partial p(A \longrightarrow \alpha)}\right)}{\sum\limits_{(A \longrightarrow \alpha) \in \Gamma_A} p(A \longrightarrow \alpha)\left(\frac{\partial log Pr(\Omega|G_p)}{\partial p(A \longrightarrow \alpha)}\right)}, \tag{2.6.4}$$

desenvolvendo algebricamente a expressão anterior e levando em consideração que o logaritmo da função de verossimilhança da amostra Ω, dada uma gramática probabilística livre do contexto G_p, é definido como:

$$log\left(Pr(\Omega|G_p)\right) = log\left(\prod_{x \in \Omega} Pr(\chi|G_p)\right),$$

logo para cada $(A \longrightarrow \alpha) \in P$:

$$\overline{p}(A \longrightarrow \alpha) = \frac{\sum\limits_{\chi \in \Omega} \frac{1}{Pr(\chi|G_p)} p(A \longrightarrow \alpha)\left(\frac{\partial Pr(\chi|G_p)}{\partial p(A \longrightarrow \alpha)}\right)}{\sum\limits_{\chi \in \Omega} \frac{1}{Pr(\chi|G_p)} \sum\limits_{(A \longrightarrow \alpha) \in \Gamma_A} p(A \longrightarrow \alpha)\left(\frac{\partial Pr(\chi|G_p)}{\partial Pr(A \longrightarrow \alpha)}\right)}. \tag{2.6.5}$$

Vamos desenvolver primeiramente o numerador da equação (2.6.5) utilizando as equações (2.3.1) e (2.3.2) obtemos:

$$\begin{aligned} p(A \longrightarrow \alpha)\left(\tfrac{\partial Pr(\chi|G_p)}{\partial Pr(A \longrightarrow \alpha)}\right) &= p(A \longrightarrow \alpha) \sum_{d_\chi \in D_\chi} \left(\tfrac{\partial Pr(\chi, d_\chi|G_p)}{\partial Pr(A \longrightarrow \alpha)}\right) \\ &= \sum_{d_\chi \in D_\chi} Nr(A \longrightarrow \alpha, d_\chi) Pr(\chi, d_\chi|G_p). \end{aligned} \tag{2.6.6}$$

Para resolver o denominador de (2.6.5) usamos a expressão (2.6.6) e como o número de vezes que o símbolo não terminal A faz parte de uma regra da derivação d_χ dado por $Nr(A, d_\chi) =$

$\sum_{(A\longrightarrow\alpha)\in\Gamma_A} Nr(A\longrightarrow\alpha, d_\chi)$, portanto

$$\begin{aligned}\sum_{(A\longrightarrow\alpha)\in\Gamma_A} p(A\longrightarrow\alpha)\left(\frac{\partial Pr(\chi|G_p)}{\partial Pr(A\longrightarrow\alpha)}\right) &= \sum_{(A\longrightarrow\alpha)\in\Gamma_A}\sum_{d_\chi\in D_\chi} Nr(A\longrightarrow\alpha, d_\chi)Pr(\chi, d_\chi|G_p)\\ &= \sum_{d_\chi\in D_\chi}\sum_{(A\longrightarrow\alpha)\in\Gamma_A} Nr(A\longrightarrow\alpha, d_\chi)Pr(\chi, d_\chi|G_p)\\ &= \sum_{d_\chi\in D_\chi} Nr(A, d_\chi)Pr(\chi, d_\chi|G_p).\end{aligned}$$

Finalmente, a expressão (2.6.5) fica:

$$\overline{p}(A\longrightarrow\alpha) = \frac{\sum_{\chi\in\Omega}\frac{1}{Pr(\chi|G_p)}\sum_{d_\chi\in D_\chi} Nr(A\longrightarrow\alpha, d_\chi)Pr(\chi, d_\chi|G_p)}{\sum_{\chi\in\Omega}\frac{1}{Pr(\chi|G_p)}\sum_{d_\chi\in D_\chi} Nr(A, d_\chi)Pr(\chi, d_\chi|G_p)}, \tag{2.6.7}$$

para cada $(A\longrightarrow\alpha)\in P$.

O algoritmo Inside-Outside aplica iterativamente essa transformação até maximizar localmente a função de verossimilhança. O ponto ótimo atingido depende dos valores iniciais. A convergência do algoritmo é garantida pelo Teorema 2.6.1 e acontece quando as probabilidades da gramática não mudam de iteração para iteração.

Para fazer os cálculos de uma forma eficiente, vamos considerar uma gramática em forma normal de Chomsky e uma regra da gramática $A\longrightarrow BC$, onde $A, B, C\in N$. Vamos supor que d_χ seja uma derivação da cadeia χ e que a regra $A\longrightarrow BC$ faz parte da sequência das regras da derivação d_χ. Figura 2.2.

Assim, a partir do símbolo inicial S é gerada a cadeia $\chi_1,\ldots,\chi_{i-1}A\chi_{j+1},\ldots,\chi_{|\chi|}$, o símbolo não terminal A gera BC, deste modo B tem que gerar $\chi_i,\ldots,\chi_k$ e C deve gerar $\chi_{k+1},\ldots,\chi_j$,

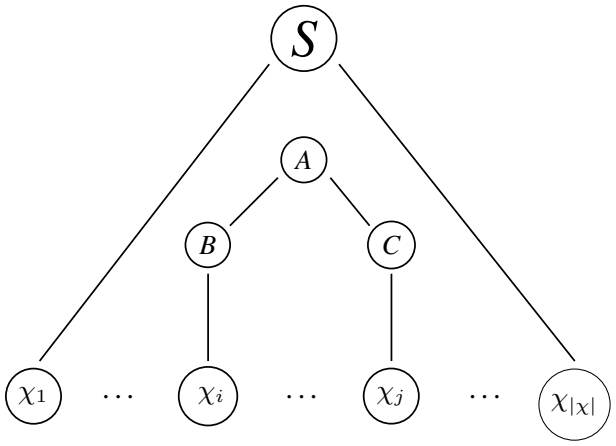


Figura 2.2: Árvore de derivação d_χ

para algum $i < k < j$.

Somando a probabilidade de todas as derivações em que a regra $A \longrightarrow BC$ está limitada para valores de i e j obtemos:

$$\begin{aligned} Pr(S \stackrel{*}{\Rightarrow} \chi_1 \ldots \chi_{i-1} A \chi_{j+1} \ldots \chi_{|\chi|}) p(A \longrightarrow BC) \times & \\ Pr(B \stackrel{*}{\Rightarrow} \chi_i \ldots \chi_k | G_p) Pr(C \stackrel{*}{\Rightarrow} \chi_{k+1} \ldots \chi_j | G_p) \quad &= \quad f(A < i, j >) p(A \longrightarrow BC) \times \\ & \quad e(B < i, k >) e(C < k+1, j >). \end{aligned} \tag{2.6.8}$$

Considerando todos os valores possíveis dos limites (i, j e k) e raciocinando da mesma forma para o denominador de (2.6.7) e para as regras tipo ($A \longrightarrow a$) com $A \in N$ e $a \in \Sigma$, obtemos:

$$\overline{p}(A \longrightarrow a) = \frac{\displaystyle\sum_{\chi \in \Omega} \frac{1}{Pr(\chi | G_p)} \sum_{i=1}^{|\chi|} f(A < i, i >) p(A \longrightarrow \chi_i)}{\displaystyle\sum_{\chi \in \Omega} \frac{1}{Pr(\chi | G_p)} \sum_{i=1}^{|\chi|} \sum_{j=i}^{|\chi|} f(A < i, j >) e(A < i, j >)}. \tag{2.6.9}$$

E para toda regra da forma $(A \longrightarrow BC) \in P$:

$$\overline{p}(A \longrightarrow BC) =$$

$$\frac{\sum_{\chi \in \Omega} \frac{P(A \longrightarrow BC)}{Pr(\chi|G_p)} \sum_{1 \leq i \leq k < j \leq |\chi|} f(A < i, j >) e(B < i, k >) e(C < k+1, j >)}{\sum_{\chi \in \Omega} \frac{1}{Pr(\chi|G_p)} \sum_{i=1}^{|\chi|} \sum_{j=i}^{|\chi|} f(A < i, j >) e(A < i, j >)}. \quad (2.6.10)$$

Observamos que estas expressões necessitam dos mesmos cálculos que os algoritmos Inside e Outside utilizam.

Assim, para alguns valores iniciais das probabilidades das regras da gramática probabilística livre do contexto, aplicamos as fórmulas (2.6.10) e (2.6.9), repetidamente até que os valores de duas iterações consecutivas alcancem uma tolerância determinada. Os valores iniciais das regras condicionam o ótimo atingido [2].

Para cada regra de derivação o algoritmo Inside-Outside necessita usar a estratégia Inside, depois a estratégia Outside e finalmente utiliza a transformação (2.6.1) até convergir. A complexidade em cada iteração é $O(3|\Omega|l_m^3|P|)$, onde $l_m = max_{x \in \Omega}|x|$. Observe que na prática o número de iterações, desse método, é elevado [22, 25].

2.7 Qualidade do modelo obtido

Uma vez, encontrada uma solução, ou seja, estimadas as probabilidades das regras da gramática probabilística livre do contexto, é preciso verificar a qualidade do modelo probabilístico obtido. A perplexidade é uma medida usada para medir e comparar os modelos obtidos in-

dependentemente do método usado. 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 Ts. Quando o modelo está relacionado a uma gramática probabilística livre do contexto, a perplexidade por palavra (PP) é definida usando a teoria da entropia [23]:

$$PP(Ts, G_p) = exp\left\{ -\frac{\sum_{\chi \in Ts} log(Pr(\chi|G_p))}{\sum_{\chi \in Ts} |\chi|} \right\}. \tag{2.7.1}$$

Intuitivamente o conceito de perplexidade é uma medida do tamanho do conjunto de palavras a partir do qual a próxima palavra é escolhida levando em consideração as anteriores. Quanto menor a perplexidade melhor a qualidade do modelo [23, 24].

Capítulo 3

Métodos de Pontos Interiores

Neste capítulo o propósito consiste em desenvolver os métodos que vamos implementar para resolver o problema de estimação dos parâmetros da gramática livre do contexto formulado no capítulo anterior.

3.1 O Problema

$$\begin{aligned} \text{maximizar} \quad & f_p(\Omega) \\ \text{sujeito a} \quad & \sum_{x \in \Psi_A} x_i = 1, \quad \forall\, \Psi_A : A \in \Sigma \\ & 0 \leq x_i \leq 1. \quad i = 1, ..., |P|. \end{aligned} \tag{3.1.1}$$

Vamos apresentar um método de pontos interiores barreira logarítmica primal-dual, sua variante preditor-corretor e o método de pontos interiores barreira logarítmica para problemas de programação linear. Na parte final da seção, apresentamos os métodos de pontos interiores barreira logarítmica e barreira logarítmica primal-dual, desenvolvidos para problemas com função objetivo não linear, e cujas restrições sejam canalizadas e lineares.

3.2 Os Problemas Primal e Dual na Forma Padrão

Em programação linear se diz que um problema está na forma padrão quando sua formulação matemática é:

$$\begin{aligned} \text{minimizar} \quad & c^t x \\ \text{sujeito a} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{3.2.1}$$

onde $c, x \in \Re^n$, $b \in \Re^m$ e $A \in \Re^{m \times n}$. Vamos supor que $m < n$ e posto$(A) = m$.

Todo problema de otimização linear pode ser transformado na forma padrão utilizando variáveis de folga e usando o fato que se o problema consiste em encontrar os valores que maximizam a função objetivo, o mesmo é equivalente a resolver um problema de minimização.

Da teoria da dualidade sabemos que associado ao problema (3.2.1) existe um outro problema denominado dual [3]:

$$\begin{aligned} \text{maximizar} \quad & b^t y \\ \text{sujeito a} \quad & A^T y + z = c \\ & z \geq 0, \end{aligned} \tag{3.2.2}$$

onde $y \in \Re^m$ e $z \in \Re^n$.

As soluções dos problemas (3.2.1) e (3.2.2) devem satisfazer as condições de otimalidade de

primeira ordem ou condições de Karush-Kuhn-Tucker (KKT) [29]:

$$\begin{aligned} A^T y + z &= c && (3.2.3a)\\ Ax &= b && (3.2.3b)\\ x_i z_i &= 0 \qquad i = 1...n && (3.2.3c)\\ (x, z) &\geq 0. && (3.2.3d) \end{aligned}$$

A equação (3.2.3c) implica que para cada $i = 1...n$, pelo menos uma das componentes x_i ou z_i deve ser zero. Esta condição é denominada *condição de complementaridade*. Este sistema é chamado de sistema KKT e um vetor $(x, y, z)^T$ que resolva o sistema KKT é chamado de ponto KKT.

Os valores x^*, (y^*, z^*) que resolvem o sistema (3.2.3) são soluções dos problemas primal e dual, respectivamente.

Considere a seguinte definição:

Definição 3.2.1. O conjunto das restrições ativas $\Lambda(x)$ em um ponto factível x, é o conjunto de índices das restrições de igualdade que estão sendo satisfeitas:

$$\Lambda(x) = \{i : a_i^T x - b_i = 0\}, \tag{3.2.4}$$

com a_i^T a i-ésima linha da matriz A, b_i o i-ésimo componente do vetor b e $1 \leq i \leq m$.

Teorema 3.2.2. Dado um ponto x e o conjunto das restrições ativas $\Lambda(x)$, se diz que a condição de qualificação de independência linear é satisfeita se o conjunto dos gradientes das

restrições ativas $\{\nabla a_i^T(x), i \in \Lambda(x)\}$ é linearmente independente.

Observe que quando todas as restrições $a_i^T x$ são funções lineares e a matriz A é de posto completo a condição de qualificação é satisfeita, e portanto as condições de qualificação do teorema KKT são satisfeitas [27].

O seguinte teorema define a relação importante entre os problemas primal e dual.

Teorema 3.2.3. Teorema da dualidade para programação linear [21]

- Se o problema primal tem uma solução com valor objetivo finito, então garante-se que o problema dual também possui solução finita, e vice-versa. Além disso, o valor ótimo das funções objetivos é o mesmo.
- Se a função objetivo do problema primal (ou dual) é ilimitada, então o problema dual (ou primal) não possui pontos factíveis.

Assim, um vetor (x^*, y^*, z^*) resolve o sistema KKT (3.2.3), se somente se, x^* resolve o problema primal (3.2.1) e (y^*, z^*) resolve o problema dual (3.2.2). O vetor (x^*, y^*, z^*) é dito solução ótima primal-dual.

3.2.1 *Gap* de Dualidade

A partir das condições KKT (3.2.3), para um x, y e z factíveis, o valor $c^T x - b^T y = z^T x$ é a diferença entre o valor da função objetivo primal e o valor da função objetivo dual. Este valor, sempre não negativo, é denominado o *gap* de dualidade.

Pelo Teorema 3.2.3 os valores ótimos das funções primal e dual são iguais ($c^T x^* = b^T y^*$), isto sugere que o *gap* de dualidade é uma medida de quão perto estamos de uma solução ótima [18].

3.3 Método de Pontos Interiores Primal-Dual

Os métodos de pontos interiores pertencem ao grupo de métodos iterativos tais que a cada iteração calculam uma direção de descida e determinam um tamanho do passo para "caminhar" nessa direção. Espera-se que a cada iteração estejamos mais próximos de uma solução. Além disso, os pontos calculados a cada iteração devem ser não negativos. O cálculo da direção e do tamanho do passo caracterizam os métodos de pontos interiores.

Este método procura soluções primal-dual, usando uma variante do método de Newton para resolver o sistema KKT (3.2.3), modificando as direções e o tamanho dos passos, e garantindo que a desigualdade $(x, z) > 0$, seja estritamente satisfeita em cada iteração.

As equações (3.2.3) podem ser reescritas como:

$$F(x,y,z): \quad \begin{pmatrix} A^T y + z - c \\ Ax - b \\ XZe \end{pmatrix} = 0 \qquad (x,z) \geq 0, \tag{3.3.1}$$

onde $F : \Re^{2n+m} \longrightarrow \Re^{2n+m}$ é não linear e $X = diag(x_1...x_n)$, $Z = diag(z_1...z_n)$ e $e = (1...1)^T$, com $e \in \Re^n$.

Usando a série de Taylor de primeira ordem, o método de Newton aproxima F a um modelo linear numa vizinhança de (x, y, z):

$$F(x,y,z) \approx F(x^0,y^0,z^0) + J(x^0,y^0,z^0)((x,y,z) - (x^0,y^0,z^0)), \tag{3.3.2}$$

onde $(x, y, z) \in B_r(x^0, y^0, z^0)$, e J é o Jacobiano de F.

Vamos encontrar o zero do modelo linear local 3.3.2:

$$0 = F(x, y, z) \approx F(x^0, y^0, z^0) + J(x^0, y^0, z^0)((x, y, z) - (x^0, y^0, z^0)),$$

equivalentemente:

$$J(x^0, y^0, z^0)((x, y, z) - (x^0, y^0, z^0)) = -F(x^0, y^0, z^0),$$

logo,

$$(x, y, z) = (x^0, y^0, z^0) - J(x^0, y^0, z^0)^{-1} F(x^0, y^0, z^0).$$

Seja $(\Delta x, \Delta y, \Delta z) = (x, y, z) - (x^0, y^0, z^0)$, assim para encontrar os valores de $(\Delta x, \Delta y, \Delta z)$ resolvemos o sistema linear:

$$J(x, y, z) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -F(x, y, z). \tag{3.3.3}$$

Desta forma obtemos as direções de Newton, e uma nova iteração será dada por:

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

onde $\alpha^k \in (0, 1]$ e k representa a iteração atual.

Dado que, eventualmente, o comprimento do passo pode ser pequeno ($\alpha^k << 1$) para evitar violar a condição $(x^k, z^k) > 0$, o progresso, no sentido de encontrar uma solução, pode ser lento. Para isso, o método de pontos interiores primal-dual modifica o sistema (3.3.3) para

que o novo ponto seja mais centralizado no interior do ortante positivo $(x^k, z^k) > 0$, e assim poder dar passos mais longos antes que x^k ou z^k fiquem negativas.

Assim, o conceito de trajetória central [29], é definido a seguir.

Trajetória Central

A trajetória central C é formada pelo conjunto de pontos $(x, y, z) \in C$ com $\mu > 0$ tal que resolve o sistema:

$$\begin{aligned} A^T y + z &= c && (3.3.4a) \\ Ax &= b && (3.3.4b) \\ XZe &= \mu e && (3.3.4c) \\ (x, z) &> 0. && (3.3.4d) \end{aligned}$$

Outra forma de escrever a trajetória central é

$$\begin{aligned} F(x, y, z) &= [0\ ,\ 0\ ,\ \mu e]^t && (3.3.5) \\ (x, z) &> 0. \end{aligned}$$

A trajetória central é um arco de pontos estritamente factíveis, que satisfazem as equações (3.3.4). Note que quando $\mu \longrightarrow 0$, a solução (x, y, z) de (3.3.4) aproxima-se de uma solução do sistema KKT (3.2.3).

Usando a trajetória central para relaxar as restrições de complementaridade:

$$J(x^k, y^k, z^k) \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{pmatrix} = -F(x^k, y^k, z^k) + \begin{pmatrix} 0 \\ 0 \\ \mu^k e \end{pmatrix}, \tag{3.3.6}$$

obtemos que $(x^k, y^k, z^k) \longrightarrow (x^*, y^*, z^*)$ quando $\mu \longrightarrow 0$.

Vamos apresentar um resumo do método de pontos interiores primal-dual:

Método 3.3.1: Primal-Dual

Dados: $(x^0, z^0) > 0$, y^0, $k = 0$.

1 **início**

2 **Calcule** $\mu^k > 0$;

3 **Resolva**

$$J(x^k, y^k, z^k) \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{pmatrix} = -F(x^k, y^k, z^k) + \begin{pmatrix} 0 \\ 0 \\ \mu^k e \end{pmatrix}; \tag{3.3.7}$$

4 **Atualize**

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

5 onde α^k é escolhido para efetuar um "bom passo", tal que $(x^{k+1}, z^{k+1}) > 0$.

6 **se** *o critério de parada é satisfeito* **então**

7 **parar;**

8 **senão**

9 **faça** $k = k + 1$ atualize μ^k e volte ao Passo **3**.

10 **fim se**

11 **fim**

Para finalizar com o desenvolvimento do método, vamos comentar três aspectos importantes: a resolução do sistema (3.3.7), o cálculo de μ^k, o cálculo de α^k e o critério de parada. Esses dois últimos caracterizam as variações do método de pontos interiores primal-dual.

Resolução do Sistema Linear

O sistema linear (3.3.7) é equivalente a:

$$\begin{pmatrix} 0 & I & A^T \\ A & 0 & 0 \\ Z^k & X^k & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta z^k \\ \Delta y^k \end{pmatrix} = \begin{pmatrix} c - A^T y^k - z^k \\ b - Ax^k \\ \mu^k e - X^k Z^k e \end{pmatrix} \tag{3.3.8}$$

eliminando a variável Δz^k o sistema é reduzido a:

$$\begin{pmatrix} -D^k & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix} = \begin{pmatrix} r_d^k - (X^k)^{-1} r_c^k \\ r_p^k \end{pmatrix} \tag{3.3.9}$$

onde $D^k = (X^k)^{-1} Z^k$, $r_p^k = b - Ax^k$, $r_d^k = c - A^T y^k - z^k$ e $r_c^k = \mu^k e - X^k Z^k e$. Note que Δz^k pode ser calculado por:

$$\Delta z^k = (X^k)^{-1}(r_c^k - Z^k \Delta x^k), \tag{3.3.10}$$

dizemos que (3.3.9) é o *sistema aumentado* [8].

Eliminando Δx^k em (3.3.9) obtemos:

$$M^k \Delta y^k = r_p^k + A((D^k)^{-1} r_d^k - (Z^k)^{-1} r_c^k), \tag{3.3.11}$$

onde $M^k = A(D^k)^{-1} A^T$ é chamado de *Complemento de Schur* [12].

Como A é uma matriz de posto completo e D^k é uma matriz diagonal definida positiva, ao resolver o complemento de Schur é possível aproveitar o fato que M^k é simétrica e definida positiva. No entanto M^k pode ser menos esparsa e ficar mais mal condicionada que a matriz do sistema aumentado (3.3.9). O pior caso acontece quando a matriz A tem pelo menos uma coluna com todas as entradas diferentes de zero, pois perde-se toda a estrutura esparsa da matriz [8] .

A abordagem mais comum para calcular as direções de Newton nos métodos de pontos interiores primal-dual consiste em resolver o complemento de Schur usando decomposição de Cholesky [12].

Cálculo de μ

Uma escolha para atualizar o valor de μ em cada iteração é [29]:

$$\mu^k = \sigma \frac{\gamma^k}{n},$$

onde

$$\gamma^k = \sum_{i=1}^{n} x_i^k z_i^k \quad \text{e} \quad 0 < \sigma < 1, \tag{3.3.12}$$

Note que se x^k e z^k forem factíveis, γ^k é o *gap* na iteração k.

Como o propósito da perturbação consiste em centralizar os iterandos e adicionalmente queremos que $\mu^k \longrightarrow 0$ quando $k \longrightarrow \infty$, usamos um valor no intervalo $(0, 1)$ para σ. Normalmente $\sigma = \frac{1}{\sqrt{n}}$ [29].

Desta forma garante-se que os iterandos estão centralizados e que a cada iteração o valor de μ está sendo reduzido.

Cálculo de α

Para o valor de α^k em cada iteração do Método 3.3.1, usamos o teste da razão:

$$\rho_p^k = \frac{-1}{min_i\left(\frac{\Delta x_i^k}{x_i^k}\right)} \quad \text{e} \quad \rho_d^k = \frac{-1}{min_i\left(\frac{\Delta z_i^k}{z_i^k}\right)}. \tag{3.3.13}$$

Observe que ρ_p^k corresponde ao valor que anula uma componente de x^k, desde que existam componentes negativas em Δx^k, caso não existam atribuímos $\rho_p^k = 1$. Um procedimento similar ocorre para ρ_d^k.

Finalmente calculamos

$$\alpha^k = min\{1, \tau\rho_p^k, \tau\rho_d^k\}, \tag{3.3.14}$$

$0 < \tau < 1$. Desta forma, $(x^{k+1}, z^{k+1}) > 0$.

Critério de parada

A convergência pode ser testada para o valor de $||F(x^{k+1}, y^{k+1}, z^{k+1})||$, ou seja, que os valores dos resíduos convergem para zero na medida que k aumenta.

3.4 Variante Preditor-Corretor

Este método é uma variante do método de pontos interiores primal-dual, que modifica o algoritmo em três aspectos:

- Calcula uma direção preditora.
- O cálculo do parâmetro σ (direção de centragem).
- Calcula uma direção corretora (tenta compensar a aproximação linear de Newton).

A ideia consiste em determinar a direção denominada preditora e estudar o progresso ao longo desta direção atuando na perturbação μ e na correção não linear.

A direção preditora é obtida resolvendo o sistema (3.3.3):

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta\tilde{x} \\ \Delta\tilde{y} \\ \Delta\tilde{z} \end{pmatrix} = \begin{pmatrix} r_p^k \\ r_d^k \\ r_c^k \end{pmatrix} \tag{3.4.1}$$

sendo $r_p^k = b - Ax^k$, $r_d^k = c - A^T y^k - z^k$ e $r_c^k = -X^k Z^k e$.

Uma vez obtidas as direções $(\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{z})^T$, determinamos o progresso ao longo das mesmas calculando:

$$\tilde{\gamma} = (x^k + \tilde{\alpha}^k \Delta\tilde{x})^T (z^k + \tilde{\alpha}^k \Delta\tilde{z})$$

onde $\tilde{\alpha}^k = min\{1, \tau\tilde{\rho}_p^k, \tau\tilde{\rho}_d^k\}$, e as razões $\tilde{\rho}_p^k$, $\tilde{\rho}_d^k$ são calculadas como em (3.3.13).

Se a direção preditora obtém uma boa redução do valor γ^k, garantindo que a condição $(x^k, z^k) > 0$ seja satisfeita, o valor de σ poderá ser mais próximo de zero. Por outro lado, se a redução do valor γ^k é pequena, para não violar a condição $(x^k, z^k) > 0$, vamos necessitar que o valor de σ seja mais próximo de 1, por exemplo [29]:

$$\sigma^k = \begin{cases} \left(\frac{\tilde{\gamma}}{\gamma^k}\right)^3, & \text{se} \quad \gamma^k > 1 \\ \frac{\gamma^k}{\sqrt{n}}, & \text{se} \quad \gamma^k \leq 1 \end{cases}$$

Uma vez obtido o valor de σ^k obtemos as direções corretoras de centragem resolvendo o sistema perturbado (3.3.6) com $\mu^k = \sigma^k \frac{\gamma^k}{n}$:

$$J(x^k, y^k, z^k) \begin{pmatrix} \Delta\hat{x} \\ \Delta\hat{y} \\ \Delta\hat{z} \end{pmatrix} = -F(\tilde{x}, \tilde{y}, \tilde{z}) + \begin{pmatrix} 0 \\ 0 \\ \mu^k e \end{pmatrix} \tag{3.4.2}$$

onde $\tilde{x} = x^k + \Delta\tilde{x}$, $\tilde{y} = y^k + \Delta\tilde{y}$, $\tilde{z} = z^k + \Delta\tilde{z}$.

Observe que se desenvolvemos o lado direito da equação 3.4.2 obtemos:

$$\begin{pmatrix} b - A\tilde{x} \\ c - A^T\tilde{y} - \tilde{z} \\ \mu^k e - \tilde{X}\tilde{Z}e \end{pmatrix} = \begin{pmatrix} b - Ax^k - A\Delta\tilde{x} \\ c - A^T y^k - A^T\Delta\tilde{y} - z^k - \Delta\tilde{z} \\ \mu^k e - (X^k + \Delta\tilde{X})(Z^k + \Delta\tilde{Z})e \end{pmatrix} \tag{3.4.3}$$

ou seja

$$\begin{pmatrix} b - A\tilde{x} \\ c - A^T\tilde{y} - \tilde{z} \\ \mu^k e - \tilde{X}\tilde{Z}e \end{pmatrix} = \begin{pmatrix} r_p^k - A\Delta\tilde{x} \\ r_d^k - A^T\Delta\tilde{y} - \Delta\tilde{z} \\ \mu^k e + r_c^k - Z^k\Delta\tilde{X}e - X^k\Delta\tilde{Z}e - \Delta\tilde{X}\Delta\tilde{Z}e \end{pmatrix} \tag{3.4.4}$$

como $(\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{z})^T$ são soluções do sistema (3.4.1) então:

$$\begin{pmatrix} b - A\tilde{x} \\ c - A^T\tilde{y} - \tilde{z} \\ \mu^k e - \tilde{X}\tilde{Z}e \end{pmatrix} = \begin{pmatrix} r_p^k - r_p^k \\ r_d^k - r_d^k \\ \mu^k e + r_c^k - r_c^k - \Delta\tilde{X}\Delta\tilde{Z}e \end{pmatrix}. \tag{3.4.5}$$

Assim, para obter as direções corretoras $(\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{z})^T$, resolvemos o sistema perturbado:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta\hat{x} \\ \Delta\hat{y} \\ \Delta\hat{z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu^k e - \Delta\tilde{X}\Delta\tilde{Z}e \end{pmatrix} \tag{3.4.6}$$

e por fim fazemos:

$$\begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{pmatrix} = \begin{pmatrix} \Delta\tilde{x} + \Delta\hat{x} \\ \Delta\tilde{y} + \Delta\hat{y} \\ \Delta\tilde{z} + \Delta\hat{z} \end{pmatrix} \tag{3.4.7}$$

Desta maneira encontramos as direções factíveis e determinamos um novo ponto como:

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \end{pmatrix} = \begin{pmatrix} x^k + \alpha^k \Delta x^k \\ y^k + \alpha^k \Delta y^k \\ z^k + \alpha^k \Delta z^k \end{pmatrix} \tag{3.4.8}$$

onde α^k é calculado como em (3.3.14).

O resumo do método da variante preditor-corretor seria feito, a grosso modo, como no resumo do Método 3.3.1, com a alteração que no Passo 3 necessita-se da solução de dois sistemas lineares.

Ainda que seja necessário resolver um segundo sistema linear por iteração com a mesma matriz, espera-se que a variante preditor-corretor obtenha uma redução significativa no número de iterações do método primal-dual. Na prática o esforço para resolver o sistema linear adicional é compensado pelo número de iterações que são realizadas [20].

3.5 Método Barreira Logarítmica

Este método pertence ao conjunto de métodos chamado de função barreira [27].

Diferentemente do método pontos interiores primal-dual, o método barreira logarítmica encontra soluções primais x^*, ou seja, encontra alguma solução do problema (3.2.1).

Este método transforma o problema (3.2.1) em um problema equivalente, utilizando de uma função barreira:

$$\begin{aligned} \text{minimizar} \quad & c^T x - \mu \sum_{i=1}^{n} log(x_i) \\ \text{sujeito a} \quad & Ax = b, \end{aligned} \tag{3.5.1}$$

onde $\mu > 0$.

O termo $\sum_{i=1}^{n} log(x_i)$ é chamado de "barreira logarítmica" porque impede que as variáveis x_i sejam negativas. Obtemos soluções cada vez mais próximas de x^* quando $\mu \longrightarrow 0$.

Uma das abordagens para resolver o problema (3.5.1), consiste em utilizar a função de Lagrange para obter um problema equivalente irrestrito:

$$\text{minimizar} \quad \mathcal{L}(x, y) \tag{3.5.2}$$

com $\mathcal{L}(x, y) = c^T x - \mu \sum_{i=1}^{n} log(x_i) + y^T(b - Ax)$ para cada $\mu > 0$, onde $y \in \Re^m$.

Desta forma podemos procurar soluções para o problema com restrições de igualdade (3.5.1) no conjunto de pontos estacionários da função de Lagrange associada. O vetor y é denominado

multiplicador de Lagrange da restrição de igualdade $Ax = b$.

$$\begin{aligned} \nabla_x \mathcal{L} := \quad & c - \mu X^{-1} e - A^T y \quad = \quad 0 \\ \nabla_y \mathcal{L} := \quad & b - Ax \qquad\qquad\quad = \quad 0 \end{aligned} \tag{3.5.3}$$

Observe que o sistema (3.5.3) é não linear. Uma das possibilidades para resolver o sistema consiste em aplicar o método Newton de forma semelhante como foi apresentado em (3.3), obtendo um sistema:

$$\begin{pmatrix} \mu X^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} \mu c - \mu X^{-1} e - A^T y \\ b - Ax \end{pmatrix}, \tag{3.5.4}$$

a ser resolvido a cada iteração.

Um algoritmo resumido do método barreira logarítmica é:

Método 3.5.1: Barreira logarítmica

Dados: $x^0 > 0$, y^0, $k = 0$

início

  Calcula $\mu^k > 0$;

  Resolva o sistema (3.5.4) para as direções $(\Delta x^k, \Delta y^k)$;

  Atualize

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

  onde α é escolhido para efetuar um "bom passo", tal que $x^k > 0$. **se** *o critério de parada é satisfeito* **então**

    parar;

  senão

    faça $k = k + 1$ atualize μ^k e volte ao Passo **3**.

  fim se

fim

Da mesma forma que na Seção 3.3 o sistema (3.5.4) pode ser resolvido como (3.3.9). O valor de α é obtido usando um teste da razão:

$$\alpha = min\{1, \tau\rho^k\} \quad \text{com } 0 < \tau < 1, \tag{3.5.5}$$

onde

$$\rho^k = \frac{-1}{min_i \left(\frac{\Delta x_i^k}{x_i^k}\right)}.$$

Por outro lado, o valor de μ deve diminuir em cada iteração, uma das fórmulas usadas para o cálculo do mesmo é $\mu^{k+1} = \frac{\mu^k}{2}$. Fórmulas mais robustas $\mu^{k+1} = \varphi(\mu^k, c^T(|x^k - x^{k-1}|))$ podem

ser encontrados por exemplo em [29, 11].

O critério de parada para este método pode ser feito sobre o valor de $c^T x$ ($|c^T x^k - c^T x^{k-1}| < \epsilon$) ou sobre o valor de x^k ($||x^k - x^{k-1}|| < \epsilon$).

3.6 Problema Canalizado

Como nossa função objetivo em (3.1.1) é não linear a partir desta seção vamos considerar problemas cuja função objetivo tem essa característica.

Um problema de otimização é dito canalizado quando existem variáveis com limite superior e inferior:

$$\begin{aligned} \text{minimizar} \quad & f(x) \\ \text{sujeito a} \quad & Ax = b \\ & l \leq x \leq u \end{aligned} \tag{3.6.1}$$

onde $f : \Re^n \longrightarrow \Re$, $x \in \Re^n$, $A \in \Re^{m \times n}$, $b \in \Re^m$ e $l, u \in \Re^n$.

Note que é possível, usando mudança de variáveis, transformar o problema (3.6.1) em um equivalente com $l = 0$ [18]. Desta forma vamos considerar no nosso problema (3.6.1) que l é o vetor de zeros.

Usando as variáveis v denominadas de folga, obtemos um problema equivalente ao problema

(3.6.1)

$$\begin{array}{lll} \text{minimizar} & f(x) & \\ \text{sujeito a} & Ax & = b \\ & x + v & = u \\ & v, x & \geq 0. \end{array} \tag{3.6.2}$$

Para resolver o problema (3.6.2), vamos expandir a teoria para programação linear, desenvolvida nas seções anteriores.

Ao invés da organização do início deste capítulo, vamos iniciar com o método barreira logarítmica, para que posteriormente a afinidade entre o mesmo e o método de pontos interiores primal-dual seja mais clara.

3.6.1 Método de Pontos Interiores Barreira Logarítmica

Da mesma forma como foi desenvolvido na teoria de programação linear, este método aproxima o problema a um outro:

$$\begin{array}{ll} \text{minimizar} & f(x) - \mu \left(\sum_{i=1}^{n} log(x_i) + \sum_{i=1}^{n} log(v_i) \right) \\ \text{sujeito a} & Ax = b \\ & x + v = u, \end{array} \tag{3.6.3}$$

onde $\mu > 0$.

Para cada valor fixo de μ e usando o Lagrangiano associado:

$$\mathcal{L}(x, v, y, w) = f(x) - \mu \left(\sum_{i=1}^{n} log(x_i) + \sum_{i=1}^{n} log(v_i) \right) + y^T(Ax - b) + w^T(u - v - x)$$

temos que uma solução x^* do problema (3.6.3), deve satisfazer as condições de otimalidade [29]:

$$\begin{pmatrix} \nabla f(x) - \mu X^{-1}e + A^T y - w \\ w - \mu V^{-1}e \\ Ax - b \\ u - v - x \end{pmatrix} = 0. \tag{3.6.4}$$

Os vetores $y \in \Re^m$ e $w \in \Re^n$ são os multiplicadores de Lagrange.

Dado que (3.6.4) é um sistema não linear, usamos o método de Newton para aproximá-lo a um modelo linear numa vizinhança de (x, v, y, w) [18]:

$$\begin{pmatrix} H(x) + \mu X^{-2} & 0 & A^T & -I \\ 0 & \mu V^{-2} & 0 & I \\ A & 0 & 0 & 0 \\ I & I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \\ \Delta y \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_d \\ r_b \\ r_p \\ r_u \end{pmatrix}, \tag{3.6.5}$$

onde $H(x) = \nabla^2 f(x)$, $r_d = \mu X^{-1} - A^T y - w - \nabla f(x)$, $r_b = \mu V^{-1}e - w$, $r_p = b - Ax$ e $r_u = u - x - v$.

A partir deste ponto vamos omitir o superescrito k para evitar uma notação sobre carregada.

Note que em (3.6.4), a segunda linha $w - \mu V^{-1}e = 0$ pode ser reescrita como $VWe - \mu e = 0$,

assim, redefinimos $r_b = \mu e - VWe$ obtemos o sistema:

$$\begin{pmatrix} H(x) + \mu X^{-2} & 0 & A^T & -I \\ 0 & W & 0 & V \\ A & 0 & 0 & 0 \\ I & I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \\ \Delta y \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_d \\ r_b \\ r_p \\ r_u \end{pmatrix}. \tag{3.6.6}$$

Eliminando a variável Δv obtemos um novo sistema reduzido:

$$\begin{pmatrix} H(x) + \mu X^{-2} & A^T & -I \\ A & 0 & 0 \\ I & 0 & W^{-1}V \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_d \\ r_p \\ r_u - W^{-1} r_b \end{pmatrix}, \tag{3.6.7}$$

onde $\Delta v = W^{-1}(r_b - V\Delta w)$.

Fazendo $\Delta w = V^{-1}W(\Delta x - r_u + W^{-1}r_b)$ reduzimos o sistema (3.6.7) para:

$$\begin{pmatrix} \hat{H} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_d + V^{-1}W(r_u - W^{-1}r_b) \\ r_p \end{pmatrix}, \tag{3.6.8}$$

onde $\hat{H} = H(x) + \mu X^{-2} + V^{-1}W$.

Resolvendo este último sistema obtemos as direções de Newton $(\Delta x, \Delta y)$. E as direções Δv e Δw, da mesma forma que no desenvolvimento do sistema (3.6.6).

A atualização das variáveis é feita da seguinte forma:

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

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, da forma similar que em (3.3.13) [29]: $\alpha = min(1, \tau\rho)$, 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 canalizado:

Método 3.6.1: Barreira Logarítmica Canalizado

Dados: $(x^0, v^0) > 0$, y^0, w^0, $\sigma = \frac{1}{\sqrt{n}}$ $k = 1$

1 **início**

2 **Calcular** o valor para $\mu = \sigma\frac{\gamma}{n}$, onde $\gamma = (v^k)^t w^k$;

3 **Calcular** as direções $\Delta x, \Delta v, \Delta y, \Delta w$ resolvendo o sistema linear (3.6.8);

4 **Determinar** α usando o teste da razão;

5 **Atualizar** o passo:

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

se *o critério de parada (3.6.9) é satisfeito* **então**

6 **parar;**

7 **senão**

8 **faça** $k = k + 1$ e volte ao Passo 2;

9 **fim se**

10 **fim**

O critério de parada que vamos usar na implementação do método está baseado nas condições

de otimalidade (3.6.4):

$$\begin{pmatrix} \frac{\|r_d\|}{\|y\|+1} \\ \frac{\gamma}{n} \\ \frac{\|r_p\|}{\|b\|+1} \\ \frac{\|r_u\|}{\|u\|+1} \end{pmatrix} \leq \epsilon. \tag{3.6.9}$$

3.6.2 Método de Pontos Interiores Barreira Logarítmica Primal-Dual

Assim como no método barreira logarítmica, este método encontra uma solução para o problema (3.6.1).

Vamos desenvolver o método de pontos interiores primal-dual barreira logarítmica, a partir do sistema não linear (3.6.5) do método barreira logarítmica. Para simplificar a leitura vamos denominá-lo, no restante do texto, como método de pontos interiores primal-dual.

Seja $\mu X^{-1}e = z$, ou seja, $XZe = \mu e$ e escrevendo novamente $\mu V^{-1}e = w \Leftrightarrow VWe = \mu e$ em (3.6.4), obtemos o sistema não linear relaxado [10]:

$$\begin{pmatrix} \nabla f(x) - z + A^T y - w \\ VWe - \mu e \\ XZe - \mu e \\ Ax - b \\ x + v - u \end{pmatrix} = 0, \quad (x, v, w, z) \geq 0, \quad \mu > 0. \tag{3.6.10}$$

Aplicando o método de Newton para resolver o sistema não linear (3.6.10) obtemos:

$$\begin{pmatrix} H(x) & 0 & -I & A^T & -I \\ 0 & W & 0 & 0 & V \\ Z & 0 & X & 0 & 0 \\ A & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \\ \Delta z \\ \Delta y \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_d \\ r_b \\ r_c \\ r_p \\ r_u \end{pmatrix}, \tag{3.6.11}$$

onde $r_d = z - \nabla f(x) - A^T y - w$, $r_c = \mu e - XZe$ e r_b, r_p, r_u como no método barreira logarítmica.

Eliminando as variáveis Δv, Δz e Δw do sistema (3.6.11) obtemos:

$$\begin{pmatrix} \hat{H} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_d - X^{-1} r_c + V^{-1}(r_b - w r_u) \\ r_p \end{pmatrix}, \tag{3.6.12}$$

onde $\hat{H} = H(x) + X^{-1}Z + V^{-1}W$. Os valores das variáveis Δv, Δz e Δw são:

$$\Delta v = W^{-1}(r_b - V\Delta w) \tag{3.6.13}$$

$$\Delta z = X^{-1}(r_c - Z\Delta x) \tag{3.6.14}$$

$$\Delta w = V^{-1}(W(r_u - \Delta x) - r_b). \tag{3.6.15}$$

Finalmente, resolvemos este último sistema (3.6.12) para $(\Delta x, \Delta y)$. Como $\hat{H}$ não é uma matriz diagonal não fazemos eliminação das variáveis no sistema (3.6.12).

Uma vez obtidas as direções de Newton $(\Delta x, \Delta v, \Delta z, \Delta y, \Delta w)$, para garantir que as variáveis $x^{k+1}, v^{k+1}, w^{k+1}, z^{k+1}$ não se tornem negativas quando o ponto é atualizado usamos o teste

da razão:

$$\alpha = min(1, \tau\rho_p, \tau\rho_d), \quad \tau \in (0,1],$$

$$\text{onde,} \qquad \rho_p = \frac{-1}{min_i(\frac{\Delta x_i}{x_i}, \frac{\Delta v_i}{v_i})} \quad e \quad \rho_d = \frac{-1}{min_i(\frac{\Delta z_i}{z_i}, \frac{\Delta w_i}{w_i})}. \tag{3.6.16}$$

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

Método 3.6.2: Primal-Dual Canalizado

Dados: y^0, $(x^0, v^0, w^0, z^0) > 0$, $\sigma = \frac{1}{2n}$, $k = 1$

início

  Calcular $\mu = (\frac{\gamma}{2n})$, onde $\gamma = (x)^t z + (v)^t w$;

  Calcular as direções $\Delta x, \Delta v, \Delta y, \Delta w$ e Δz resolvendo o sistema linear (3.6.12) ;

  Determinar α usando o teste da razão;

  Atualizar o passo

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

  se *o critério de parada (3.6.9) é satisfeito* **então**

    parar;

  senão

    faça $k = k + 1$ e volte ao Passo 2;

  fim se

fim

3.7 Controle do Tamanho do Passo

Como o problema (3.6.1) pode ser ou não convexo, as direções dadas pelo sistema (3.6.5) e (3.6.11) nem sempre garantem progresso com o tamanho do passo dado por α que é calculado usando o teste da razão.

Para abordar esta dificuldade usamos um controle de passo simples (*backtracking*) para a direção Δx:

Método 3.7.1: Controle de Passo

Dados: $0 < \rho < 1$, $\alpha > 0$

1 **enquanto** $f(x^k + \alpha\Delta x) > f(x^k)$

2 $\quad \alpha = \rho\alpha$;

3 **fim enquanto**

Além disso exigimos que o valor dos resíduos e do *gap* seja reduzido para satisfazer a factibilidade da solução, para isso usamos de novamente um controle de passo simples (*backtracking*) para o valor dos resíduos e do *gap* :

Método 3.7.2: Controle de Passo factibilidade

Dados: $0 < \rho < 1$, $\alpha > 0$ e $(\Delta x, \Delta v, \Delta y, \Delta w)$

1 **enquanto** $(||r_p^k|| > ||r_p^{k+1}||)$ *e* $(||r_d^k|| > ||r_d^{k+1}||)$ *e* $(||r_u^k|| > ||r_u^{k+1}||)$ *e* $(||\gamma^k|| > ||\gamma^{k+1}||)$

2 $\quad \alpha = \rho\alpha$;

3 **fim enquanto**

Acrescentando o Método 3.7.1 e Método 3.7.2 antes de continuar com o Passo 5 nos Métodos 3.6.1 e 3.6.2 buscamos um máximo local factível do problema.

Capítulo 4

Experimentos Numéricos

No capítulo anterior descrevemos os métodos que empregados para estimar os parâmetros da gramática probabilística livre do contexto. Uma vez implementados os algoritmos fizemos testes para validar a implementação, fazer a análise do tempo de processamento, a análise da convergência dos métodos, e a qualidade dos modelos obtidos.

4.1 Introdução

Implementamos os métodos de pontos interiores barreira logarítmica e barreira logarítmica primal-dual desenvolvidos no Capítulo 3, usando o software matemático MatLab®, versão 7.8.0, em uma máquina com processador Intel CORE i3 e memória RAM de 3.7Gb, com o sistema operacional Linux 3.0.0-23-generic.

Os valores dos parâmetros usados nas implementações do método de pontos interiores barreira logarítmica e o método de pontos interiores barreira logarítmica primal-dual estão na Tabela 4.1. Estes valores foram estabelecidos experimentalmente em testes nas implementações feitas nesta disertação.

	Método de pontos interiores barreira logarítmica primal-dual	Método de pontos interiores barreira logarítmica
Parâmetros	$\epsilon = 10^{-8}$ $\tau = 0,99995$ $\mu = \sigma\frac{\gamma}{2n}$, $\sigma = \frac{1}{\sqrt{2n}}$ $\gamma = x^T z + v^T w$	$\epsilon = 10^{-8}$ $\tau = 0,99995$ $\mu = \frac{v^T w}{n^2}$

Tabela 4.1: Parâmetros usados nas implementações

O valor inicial do parâmetro μ no método pontos interiores barreira logarítmica foi igual a 100.

4.1.1 Problemas Testados

Com o objetivo de testar os métodos implementados, usamos duas gramáticas, as quais estão na forma normal de Chomsky. Para cada uma delas, usamos amostras de tamanhos diferentes para construir os problemas. As características das gramáticas usadas estão especificadas na Tabela 4.2.

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 4.2: Características das gramáticas

A função objetivo (polinômio em várias variáveis) dos problemas é construída a partir de uma amostra da gramática. Usamos o software desenvolvido por Alvarez e Ibarra [16] que nos proporciona as informações necessárias para a implementação tanto da função objetivo como do gradiente e Hessiana. O código deste software foi modificado para que os dados que

obtemos a partir dele estejam no formato que precisamos.

	Tamanho da amostra	Grau do polinômio $f(x)$	Restrições $g(x)$
Gramática 1	20	270	
	50	770	$\left(\sum_{i=1}^{8} x_i\right) - 1$
	70	1114	
	100	1656	$\left(\sum_{i=9}^{13} x_i\right) - 1$
	130	2366	
	200	3992	$\left(\sum_{i=14}^{16} x_i\right) - 1$
	220	4474	
	316	6030	$x_{17} - 1$
	350	6698	$x_{18} - 1$
	434	8664	$x_{19} - 1$
	500	9794	$x_{20} - 1$
	550	10850	$x_{21} - 1$
	600	12058	$x_{22} - 1$
	700	13792	$x_{23} - 1$
	800	16092	$x_{24} - 1$
	1000	20104	$x_{25} - 1$
	2000	40208	
	4000	80416	

Tabela 4.3: Caraterísticas dos problemas - Gramática 1

Os problemas de otimização construídos a partir das amostras das gramáticas, estão detalhados na Tabela 4.3 e na Tabela 4.4. Nelas relacionamos cada uma das gramáticas que foram usadas, com o tamanho de suas amostras, o grau do polinômio obtido a partir de cada uma das amostras e na última coluna das tabelas as restrições lineares dos problemas. Note que estas restrições não variam de amostra para amostra numa mesma gramática, pois elas dependem do conjunto de regras de derivação da gramática. Observe também que x_i, $i = 1...|P|$ representam as probabilidades das regras que estão enumeradas em alguma ordem pré-estabelecida.

No caso da Gramática 1, pela particularidade de algumas das restrições ($x_i = 1$, com $i =$

	Tamanho da amostra	Grau do polinômio $f(x)$	Restrições $g(x)$
Gramática 2	20	254	$\left(\sum_{i=1}^{4} x_i\right) - 1$
	50	672	$\left(\sum_{i=5}^{12} x_i\right) - 1$
	100	1364	$\left(\sum_{i=13}^{15} x_i\right) - 1$
	160	2190	$\left(\sum_{i=16}^{18} x_i\right) - 1$
	200	2720	$\left(\sum_{i=19}^{20} x_i\right) - 1$
	260	3556	$\left(\sum_{i=21}^{29} x_i\right) - 1$
	300	4094	$\left(\sum_{i=30}^{37} x_i\right) - 1$
	400	5392	$x_{38} - 1$
	500	6732	$\left(\sum_{i=39}^{41} x_i\right) - 1$
	1000	13488	$\left(\sum_{i=42}^{44} x_i\right) - 1$
	4000	54430	$x_{45} - 1$ $x_{46} - 1$ $x_{47} - 1$

Tabela 4.4: Caraterísticas dos problemas - Gramática 2

17...25) foi possível reduzir o número de variáveis do problema, de 25 para 16. E na Gramática 2 o número de variáveis do problema foi reduzido de 47 para 43.

4.1.2 Estratégias Utilizadas na Implementação

Uma vez testadas as implementações com amostras de tamanho pequeno (10, 20, 50), usamos amostras maiores (4000) para fazer os testes finais. Durante o desenvolvimento dos testes aumentando o tamanho da amostra, percebemos algumas dificuldades que vamos comentar e descrever como foram superadas nos parágrafos a seguir.

Dado que a função objetivo depende da amostra e, consequentemente o grau do polinômio aumenta na medida que o tamanho desta aumenta, para amostras maiores um dos principais problemas são os erros numéricos (valores muito pequenos ou grandes) que surgem no cálculo da função objetivo e portanto no cálculo do gradiente e Hessiana. Para remover este inconveniente usamos uma constante c e resolvemos os problemas com a função objetivo $cf(x)$, determinamos a constante c, utilizando o número de termos (polinômios) que estão sendo multiplicados na função objetivo, lembre-se que esta é um produtório de polinômios, e note também que para um problema com amostra tamanho 20, o número de termos do produtório de fato é 20. Com essa informação escolhemos $c = r^{|\Omega|}$, onde r é um valor múltiplo de 10, este valor é escolhido experimentalmente, onde $|\Omega|$ representa o tamanho da amostra.

Uma das principais dificuldades apresentadas pelos problemas é a procura de um "bom" ponto inicial de modo a garantir a convergência dos métodos. Estes pontos foram procurados aleatoriamente para amostras pequenas e escolhidos aqueles que forneceram melhores resultados em termos da perplexidade, tempo de convergência e número de iterações. Aplicamos os métodos às amostras pequenas e uma vez que convergiram, escolhemos pontos intermediários da sequência convergente e os usamos como pontos iniciais para problemas com amostras maiores. Repetimos o procedimento, até obter um ponto inicial para o problema com a amostra de maior tamanho com que trabalhamos (4000).

Em termos de tempo de processamento, nas implementações dos métodos pontos interiores barreira logarítmica e barreira logarítmica primal-dual, o cálculo da matriz Hessiana tem um custo elevado. Foram implementadas duas rotinas, uma delas calcula a Hessiana exata e a outra calcula uma aproximação da Hessiana usando diferenças finitas. Realizamos vários experimentos nos diversos problemas, testando as duas implementações onde o erro relativo entre as duas Hessianas é da ordem de 10^{-8}. Os resultados obtidos usando um método ou

outro não apresentam maior variação no ponto ótimo local encontrado, nem no valor da perplexidade, embora o tempo de convergência seja menor no caso da Hessiana aproximada. Assim, reduzimos o tempo de processamento usando o método das diferenças finitas para o cálculo da Hessiana.

Nas implementações dos métodos, na rotina do *backtracking* está sendo utilizada um valor máximo de 40 iterações do *backtracking* buscando reduzir o valor da função objetivo e máximo de 30 iterações para tentar diminuir o valor dos resíduos. Estes valores são fixos para todos os testes.

4.2 Resultados Numéricos

Primeiramente vamos apresentar resultados numéricos obtidos para os problemas da Gramática 1 e Gramática 2. Com cada uma das amostras das duas tabelas construímos os problemas e aplicamos os dois métodos. Para determinar o valor da perplexidade por palavra, em todos os casos, utilizamos uma amostra de tamanho 2000, de tal forma que a interseção entre as amostras utilizadas para construir os problemas e a amostra utilizada para determinar o valor da perplexidade por palavra é vazia.

Alguns resultados numéricos para os problemas da Gramática 1 estão detalhados na Tabela 4.5, e para os problemas da Gramática 2 apresentamos três resultados na Tabela 4.6. Nestas tabelas detalhamos os tamanhos das amostras, método utilizado, número de iterações e tempo de processamento em segundos até convergir, assim como a perplexidade por palavra.

Nos problemas da Gramática 1, tanto para o método pontos interiores barreira logarítmica

como para método pontos interiores barreira logarítmica primal-dual, foi utilizado um mesmo ponto inicial. A estratégia utilizada para encontrar o ponto inicial foi a seguinte: começamos com o problema de amostra tamanho 20, para o qual determinamos seu ponto inicial através de testes com valores aleatórios entre (0,1) para todas as variáveis. Entre todos os que convergiram, selecionamos aquele que obteve o menor valor de perplexidade. Em seguida escolhemos o ponto da sequência convergente do método barreira logarítmica. Neste caso particular os valores da iteração 23 (última iteração), como ponto inicial para os problemas de amostra de tamanho no intervalo $[50, 2000]$. E por fim para o problema da amostra 4000, usamos como ponto inicial um dos últimos pontos da sequência convergente do método primal-dual barreira logarítmica aplicado ao problema com tamanho de amostra 500.

Para os problemas da Gramática 2 não foi possível encontrar, com o procedimento descrito acima, um ponto inicial de modo que os dois métodos convergissem. No caso do método pontos interiores barreira logarítmica utilizamos a estratégia de escolher pontos da sequência convergente (estratégia descrita acima) e para o método pontos interiores barreira logarítmica primal-dual usamos como ponto inicial, o ponto inicial calculado aleatoriamente no problema da amostra menor (20), ou seja um mesmo ponto inicial para todos os problemas.

Tamanho amostra	Método	Número iterações	Tempo (seg)	Perplexidade por Palavra
20	Barreira Logarítmica	23	14,6692	3,3422
	Primal dual	9	8,2426	3,5171
500	Barreira Logarítmica	19	670, 4786	3,4382
	Primal dual	1	8, 2426	3,4422
4000	Barreira Logarítmica	20	12448, 4759	3,4382
	Primal dual	1	8, 2426	3,4382

Tabela 4.5: Resultados Gramática 1

Uma vez que o ponto inicial para problemas com amostras maiores depende da solução de

Tamanho amostra	Método	Número iterações	Tempo (seg)	Perplexidade por Palavra
20	Barreira Logarítmica	8	27,6678	19,4277
	Primal dual	8	28,4952	22,9811
500	Barreira Logarítmica	8	379,3173	19,5804
	Primal dual	8	314,8892	22,9811
4000	Barreira Logarítmica	17	28498,3234	19,3310
	Primal dual	8	9031,0355	23,0644

Tabela 4.6: Resultados Gramática 2

problemas menores, o tempo da soluções desses é acrescido ao tempo de solução dos maiores.

Com o propósito de comparar o desenvolvimento dos dois métodos na medida que o tamanho da amostra aumenta, apresentamos em três gráficos o comportamento tanto do número de iterações, como do tempo e da perplexidade por palavra.

Para estes gráficos usamos os resultados dos problemas da Gramática 1 (Tabela 4.3). Usamos a estratégias já descritas, obtendo os seguintes resultados: a partir da Figura 4.1 podemos ver que, no caso do método pontos interiores barreira logarítmica, o número de iterações se conserva igual em todos os problemas onde não foi mudado o ponto inicial e para amostra tamanho 4000, quando é preciso alterar o ponto inicial, apresenta-se uma pequena variação. Mudamos o ponto inicial quando o valor da perplexidade piora ou quando acontecem erros numéricos significativos (valores muito grandes e/ou muito pequenos) no cálculo do gradiente ou Hessiana. Para o caso do método pontos interiores barreira logarítmica primal-dual e para todos os tamanhos das amostras utilizadas, o ponto inicial adotado já é um ponto ótimo dos problemas maiores.

Na Figura 4.2 observamos que, o valor da perplexidade aumenta quando o tamanho da amostra do problema fica muito maior que aquela que foi usada para determinar o ponto inicial.

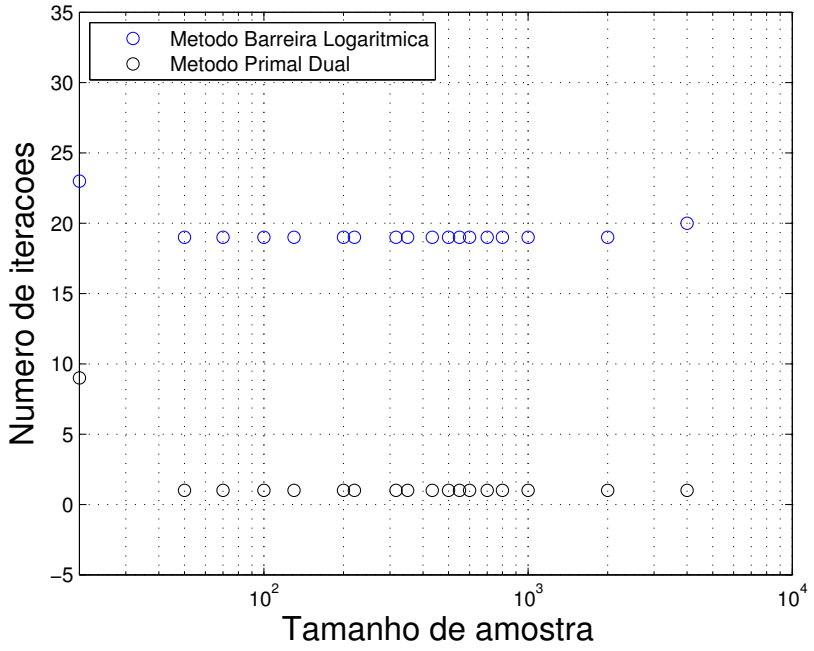


Figura 4.1: Número de Iterações × tamanho amostra

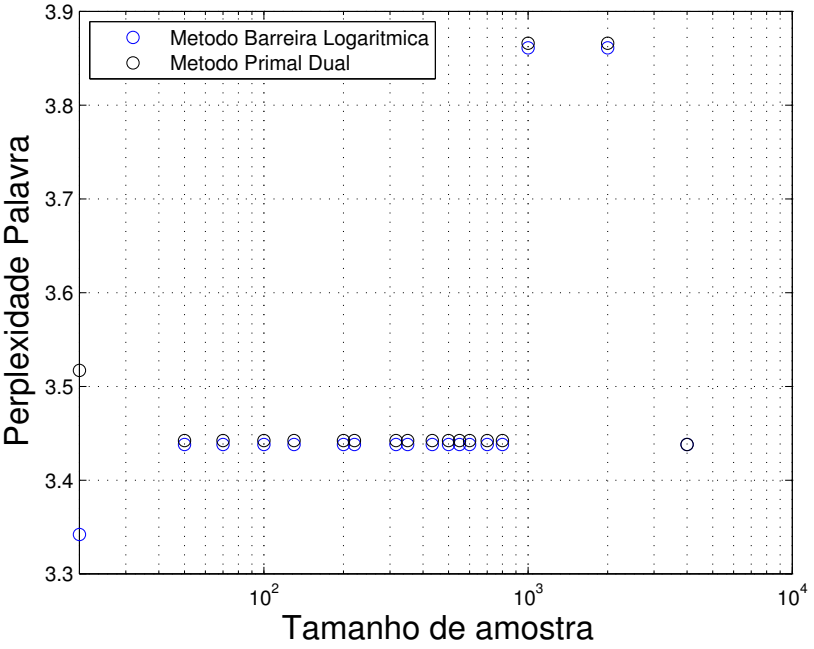


Figura 4.2: Perplexidade por palavra × tamanho amostra

Como foi descrito acima, mudamos o ponto inicial para o problema com tamanho de amostra 4000 e vemos que valor da perplexidade melhora.

A medida que aumenta o tamanho das amostras, acrescenta-se novos monômios no polinômio e portanto aumenta o custo computacional para o cálculo da matriz Hessiana e como o número de iterações não apresenta grandes variações de uma amostra para outra, espera-se que o tempo de convergência do método aumente de acordo com a dimensão da Hessiana. Na Figura 4.3 observamos este comportamento, mas observa-se também que, para alguns tamanhos de amostras próximos, este fato nem sempre ocorre - o tempo de convergência da amostra 316 diminuiu em relação ao problema de amostra 220 - este comportamento pode ser explicado pelo fato de usar *backtracking* na implementação, pois o número de *bactrakings* pode variar muito de um problema para outro.

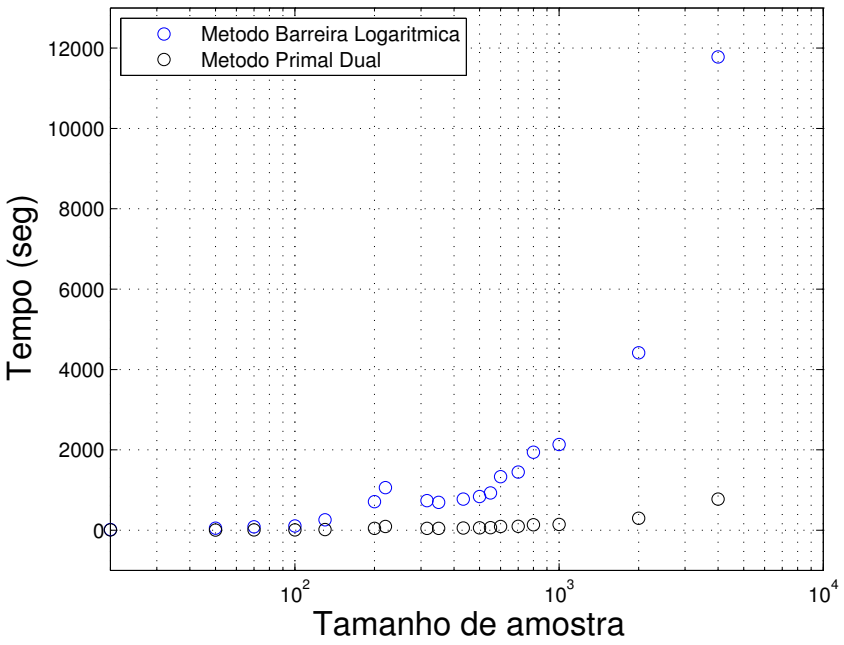


Figura 4.3: Tempo de convergência × tamanho amostra

Comentários

Foi usada a estratégia de procurar os pontos iniciais de forma aleatória em problemas de amostras maiores (> 500), não obtendo bons resultados: primeiramente a busca de um ponto inicial que convirja e cujo valor da perplexidade por palavra seja baixo, consiste em um espectro muito amplo de possibilidades, e com o aumento da dimensão dos problemas, o tempo de processamento também aumenta, tornando pouco viável o uso desta estratégia. Além disso, dado que, a constante c deve ser calibrada de acordo com os valores iniciais para não atingir o limite da precisão da máquina, a procura do ponto inicial torna-se mais complexa e demorada.

Os problemas da Gramática 1 foram resolvidos novamente usando o método de pontos interiores barreira logarítmica primal-dual. Nestes problemas a tolerância de parada foi reduzida até 10^{-60} em alguns casos. Os métodos convergiram obtendo um erro relativo da ordem de 10^{-15}, entre os valores da perplexidade das tolerâncias 10^{-8} e 10^{-60}. Observe-se também um crescimento do número de iterações até convergir, aumentando assim o tempo de processamento. Logo, para analisar estes problemas, trabalhar com uma tolerância de 10^{-8} é suficiente.

Alterando o número máximo de iterações da implementação do *backtracking*, em alguns casos reduz o número de iterações até convergir, ou seja, a implementação do método pode ser melhorada neste passo.

A partir dos resultados, observa-se que escolhendo pontos da sequência de convergência do método de pontos interiores barreira logarítmica, como pontos inciais do método pontos interiores barreira logarítmica primal-dual obtemos bons resultados em termos de tempo, número de iterações e perplexidade por palavra. Comparando com o método Inside-Outside, a per-

plexidade por palavra obtida para tamanho de amostra 4000 da Gramática 1, foi de 25,424 [16]. Observe que este valor é muito maior que obtido pelos métodos de pontos interiores para o mesma amostra: 3,4382 (vide Tabela 4.5).

A análise foi realizada também para os resultados dos problemas da Gramática 2, obtendo comportamento similar à Gramática 1. Logo, as conclusões para a Gramática 2 são análogas às obtidas para a Gramática 1.

Capítulo 5

Conclusões e Perspectivas futuras

Nesta última seção apresentamos as conclusões a partir dos resultados obtidos e da análise das implementações. Adicionalmente propomos trabalhos futuros para melhorar as implementações e testar gramáticas com amostras de grande porte.

5.1 Conclusões

Uma vez obtidos os resultados numéricos das implementações e analisado os mesmos, podemos concluir o seguinte:

- com a estratégia que utiliza pontos inicias intermediários, os problemas convergem com tempos de processamento melhores e valor da perplexidade por palavra menor, que aqueles que convergiram com pontos procurados aleatoriamente;
- tanto os ajustes dos métodos, como os estudos de convergência e dos valores da perplexidade por palavra, podem ser feitos com problemas de amostras menores que necessitam menor tempo. Com a estratégia usada para encontrar o ponto inicial, é possível obter soluções muito próximas em amostras maiores. Isto significou uma vantagem pois po-

demos trabalhar com problemas menores, para depois generalizar estes resultados para os problemas maiores;

- a abordagem dos métodos de pontos interiores é muito robusta, convergindo para tolerâncias extremamente pequenas;
- os resultados obtidos para 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.

5.2 Perspectivas Futuras

- Implementar eficientemente os métodos na linguagem C++, para melhorar os tempos de processamento.
- Implementar e aplicar nos problemas os métodos de pontos interiores barreira logarítmica preditor-corretor e pontos interiores barreira logarítmica primal-dual preditor-corretor e fazer o estudo respectivo. Esperamos diminuir o número de iterações até obter convergência com esses métodos.
- Procurar alternativas para o cálculo da Hessiana, pois o cálculo da mesma é o que consome maior tempo na implementação, junto com a estratégia do *backtracking*.
- Estudar e implementar estratégias robustas de *backtraking*.
- Estudar a viabilidade dos métodos em linguagens naturais.
- Estudar a variação do valor da perplexidade quando a tolerância é reduzida.
- Experimentar uma abordagem híbrida, utilizando a solução obtida pelo método pontos interiores, para acelerar a convergência do método *Inside-Outside*.

- Em gramáticas de grande porte, as dimensões da matriz das restrições do problema pode ser bem maior. Portanto, seria preciso estudar métodos para resolver o sistema linear que envolve esta matriz, e que além disso, explore sua estrutura matricial.

Referências Bibliográficas

[1] Lalit R Bahl, Frederick Jelinek, and Robert L Mercer, *A maximum likelihood approach to continuous speech recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **5** (1983), 179–190.

[2] Leonard E. Baum and J. A. Eagon, *An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology*, Bulletin of the American Mathematical Society **73** (1966), 360–363.

[3] Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali, *Linear programming and network flows*, John Wiley and sons, 2010.

[4] José Miguel Benedí and Joan Andreu Sánchez, *Estimation of stochastic context-free grammars and their use as language models*, Computer Speech & Language **19** (2005), no. 3, 249–274.

[5] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin, *A statistical approach to machine translation*, Computational linguistics **16** (1990), no. 2, 79–85.

[6] Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer, *The mathematics of statistical machine translation: Parameter estimation*, Computational Linguistics **19** (1993), 263–311.

[7] Noam Chomsky, *Systems of syntactic analysis.*, Journal of Symbolic Logic **18** (1953), no. 3, 242–256.

[8] Iain.S. Duff, A.M. Erisman, and J.K. Reid, *Direct methods for sparse matrices*, Oxford: Claredon, 1989.

[9] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison, *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*, Cambridge University Press, 1998.

[10] AS El-Bakry, Richard A Tapia, T Tsuchiya, and Yin Zhang, *On the formulation and theory of the newton interior-point method for nonlinear programming*, Journal of Optimization Theory and Applications **89** (1996), no. 3, 507–541.

[11] Michel C. Ferris, Olvi L. Mangasarian, and Stephen J. Wright, *Linear programming with matlab*, SIAM, 2007.

[12] Gene H. Golub and Charles F. Van Loan., *Matrix computations*, The Johns Hopkins University Press, 1996.

[13] Rafael C. Gonzales and Michel G. Thomanson, *Syntactic pattern recognition*, Addison-Wesley Publishing Company, 1978.

[14] John E Hopcroft and Jeffrey D Ullman, *Formal languages and their relation to automata*, (1969).

[15] XD Huang, Y Ariki, MA Jack, and Michael L Mauldin, *Hidden Markov models for speech recognition*, Computational Linguistics **19**, no. 1.

[16] Johnny Ferney Ibarra and Astrid Yineeth Alvarez, *Uso del algoritmo inside-outside en la estimación de parámetros de una gramática incontextual probabilística*, Universidad del Cauca. Departamento de Matemáticas, 2009.

[17] Karim Lari and Steve J Young, *The estimation of stochastic context-free grammars using the inside-outside algorithm*, Computer speech & language **4** (1990), no. 1, 35–56.

[18] David G. Luenberger and Yinyu Ye, *Linear and nonlinear programming*, 3a ed., Springer, 2008.

[19] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz, *Building a large annotated corpus of English: The penn treebank*, Computational Linguistics **19** (1993), no. 2, 313–330.

[20] Sanjay Mehrotra, *On the implementation of a primal-dual interior point method*, Siam Jornual on Optimization **2** (1992), no. 4, 575–601.

[21] Jorge Nocedal and Stephen J Wright, *Numerical optimization*, 2a ed., Springer Science, 2006.

[22] Joan Andreu Sánchez Peiró, *Estimación de gramáticas incontextuales probabilísticas y su aplicación en modelación del lenguaje*, Ph.D. thesis, Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y computación, 1999.

[23] Salim Roukos, *Language representation*, Survey of the State of the Art in Human Language Technology, 1996, pp. 28–33.

[24] Stuart Russell and Peter Norvig, *Inteligência artificial*, Elsevier, 2004.

[25] Joan-Andreu Sanchez and José-Miguel Benedi, *Estimation of the probability distributions of stochastic context-free grammars from the k-best derivations*, Fifth International Conference on Spoken Language Processing, 1998.

[26] David B Searls, *The language of genes*, Nature **420** (2002), no. 6912, 211–217.

[27] Wenyu Sun and Ya-Xiang Yuan, *Optimization theory and methods nonlinear programming*, 1a ed., Springer Science, 2006.

[28] Enrique Vidal, *Grammatical inference: an introductory survey*, Grammatical Inference and Applications, Springer, 1994, pp. 1–4.

[29] Stephen J. Wright, *Primal-dual interior-point methods*, SIAM, 1997.

[30] Takashi Yokomori and Satoshi Kobayashi, *DNA evolutionary linguistics and RNA structure modeling: a computational approach*, Intelligence in Neural and Biological Systems, 1995. INBS'95, Proceedings., First International Symposium on, IEEE, 1995, pp. 38–45.

Apêndice

Estratégia Inside

Considere o Exemplo 2.5.1 e a cadeia $\chi = baaba \in \Omega$. Dentre suas árvores de derivação Figura 2.1, vamos escolher a Derivação $d_{1\chi}$

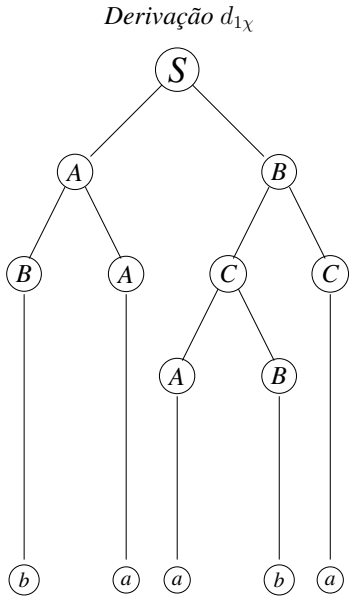


Figura 5.1: Árvore de derivação da cadeia χ

Vamos calcular, utilizando a estratégia Inside, a probabilidade da cadeia χ, ou seja

$$e(S < 1, |\chi| >) = \sum_{A,B \in N} p(S \to AB) \sum_{k=1}^{|\chi|-1} e(A < 1, k >) e(B < k+1, |\chi| >). \tag{5.2.1}$$

Vamos supor que a cadeia χ somente possui uma árvore de derivação $d_{1\chi}$, note que com esta suposição o primeiro somatório da equação (5.2.1) se reduz a um único termo:

$$e(S < 1, |\chi| >) = p(S \to AB) \sum_{k=1}^{|\chi|-1} e(A < 1, k >) e(B < k+1, |\chi| >). \tag{5.2.2}$$

Como $x_1 = p(S \to AB)$, temos:

$$e(S < 1, |\chi| >) = x_1 \sum_{k=1}^{|\chi|} e(A < 1, k >) e(B < k+1, |\chi| >). \tag{5.2.3}$$

Vamos analisar o primeiro termo do somatório ($k = 1$):

$$e(A < 1, 1 >) e(B < 2, 5 >), \tag{5.2.4}$$

ou seja, a probabilidade que a partir do não terminal A pode-se gerar a subcadeia b pela probabilidade que a partir do não terminal B o restante da cadeia seja gerado, isto é $aaba$. Na Figura 5.2, observamos que a partir do não terminal B, não existe uma sequência de regras de derivação tal que gere a subcadeia $aaba$, portanto $(B < 2, 5 >) = 0$, assim o primeiro termo do somatório é igual a zero.

De forma semelhante ao feito no primeiro termo do somatório, determinamos o segundo termo: $e(A < 1, 2 >) e(B < 3, 5 >) \neq 0$. Ver Figura 5.3.

Da Figura 5.4 e Figura 5.5 podemos verificar que o terceiro e o último termo do somatório são iguais a zero.

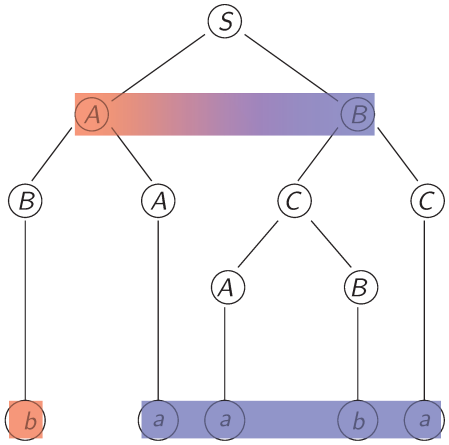


Figura 5.2: Cálculo do termo e(A<1,k>)e(B<k+1,|χ|>), onde $k = 1$

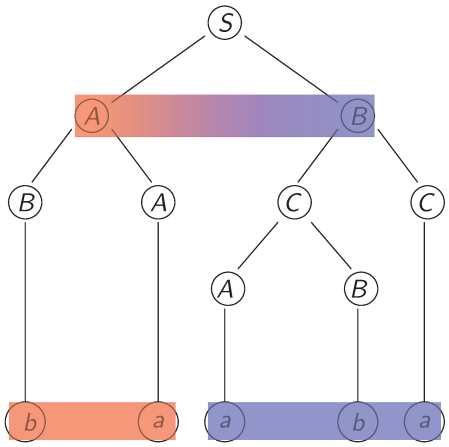


Figura 5.3: Cálculo do termo e(A<1,k>)e(B<k+1,|χ|>), onde $k = 2$

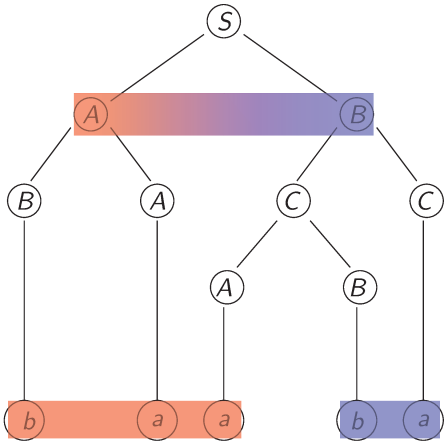


Figura 5.4: Cálculo do termo e(A<1,k>)e(B<k+1,|χ|>), onde $k = 3$

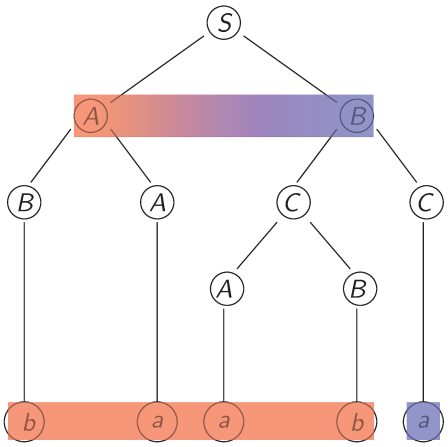


Figura 5.5: Cálculo do termo e(A<1,k>)e(B<k+1,|χ|>), onde $k = 4$

Assim, a probabilidade da cadeia χ é dada por:

$$e(S < 1, |\chi| >) = x_1 e(A < 1, 2 >) e(B < 3, 5 >). \tag{5.2.5}$$

Para calcular os termos $e(A < 1, 2 >)$ e $e(B < 3, 5 >)$ aplicamos novamente a estratégia

Inside:

$$\begin{aligned} e(A<1,2>) &= p(A\to BA)e(B<1,1>)e(A<2,2>) \\ e(B<3,5>) &= p(B\to CC)\left(e(C<3,3>)e(C<4,5>)+e(C<3,4>)e(C<5,5>)\right) \end{aligned}$$

Utilizando a equação (2.6.2a) e dado que para cada regra da gramática foi associada uma probabilidade (ver Tabela 2.1), temos que:

$$\begin{aligned} e(A<1,2>) &= x_3x_6x_4 \\ e(B<3,5>) &= x_5\left(x_8e(C<4,5>)+e(C<3,4>)x_8\right), \end{aligned}$$

mas observe que $e(C<4,5>)=0$ e que aplicando novamente a estratégia Inside ao termo $e(C<3,4>)$, obtemos:

$$e(A<1,2>) = x_3x_6x_4 \tag{5.2.6}$$

$$e(B<3,5>) = x_5x_7x_4x_6x_8. \tag{5.2.7}$$

Portanto:

$$Pr(\chi|G_p) = e(S<1,|\chi|>) = x_1x_3x_4^2x_5x_6^2x_7x_8. \tag{5.2.8}$$

Estratégia Outside

Com as mesmas considerações de antes, vamos calcular a probabilidade da cadeia χ aplicando a estratégia Outside, ou seja:

$$Pr(\chi|G_p) = \sum_{A \in N} f(A < i, i >) p(A \longrightarrow \chi_i), \tag{5.2.9}$$

com $1 \leq i \leq |\chi|$.

Considere $i = 2$, o somatório da equação (5.2.9) indica que devemos procurar dentre todos os símbolos não terminais, aquele que derive no símbolo terminal $\chi_2 = a$, ou seja, que exista a regra de derivação $A \longrightarrow a$, onde A pode ser qualquer elemento de N. Dado que estamos supondo que χ possui uma árvore de derivação (ver Figura 5.6), então:

$$\begin{aligned} Pr(\chi|G_p) &= f(A < 2, 2 >) p(A \longrightarrow a) \\ Pr(\chi|G_p) &= f(A < 2, 2 >) x_4. \end{aligned}$$

Aplicamos novamente a estratégia Outside ao termo $f(A < 2, 2 >)$ (ver Figura 5.7), ou seja procuramos dentre todas as regras de derivação aquela que seja da forma $B \to CA$ ou $B \to AC$ onde B e C correspondem a qualquer símbolo não terminal. Observe que na equação (2.6.3b) é necessário utilizar a estratégia Inside:

$$\begin{aligned} f(A < 2, 2 >) &= p(A \to BA) \left(\sum_{k=1}^{1} f(A < k, 2 >) e(B < k, 1 >) \right) \\ f(A < 2, 2 >) &= x_3 f(A < 1, 2 >) e(B < 1, 1 >) \\ f(A < 2, 2 >) &= x_3 f(A < 1, 2 >) x_6 \end{aligned}$$

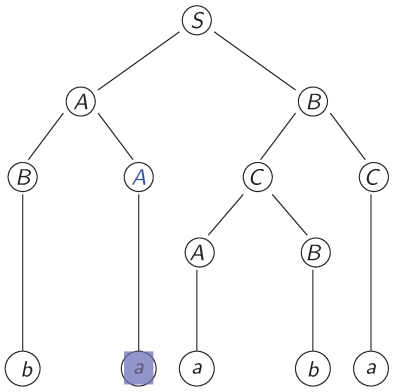


Figura 5.6: Cálculo de $p(A \rightarrow a)$

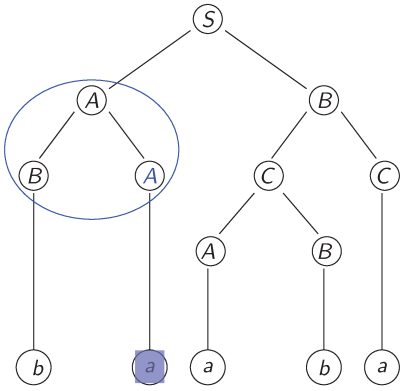


Figura 5.7: Cálculo de $f(A < i, i >)$, onde $i = 2$

Aplicando novamente a estratégia Outside (ver Figura 5.8):

$$
\begin{aligned}
f(A<1,2>) &= p(S \to AB)\left(\sum_{k=3}^{5} f(S<1,k>)e(B<3,k>)\right) \\
f(A<1,2>) &= p(S \to AB)\,(f(S<1,3>)e(B<3,3>) + (f(S<1,4>)e(B<3,4>)+ \\
&\quad f(S<1,5>)e(B<3,5>))
\end{aligned}
$$

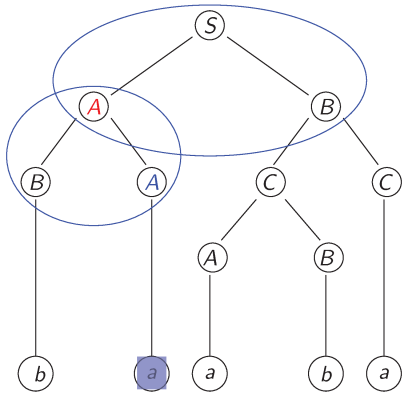


Figura 5.8: Cálculo de $f(A<1,2>)$

Note que $f(S<1,3>) = f(S<1,4>) = 0$ porque as cadeias $\chi_1...\chi_3 = baa$ e $\chi_1...\chi_4 = baab$ não pertencem à amostra Ω, logo:

$$
\begin{aligned}
f(A<1,2>) &= p(S \to AB)(S<1,5>)e(B<3,5>) \\
f(A<1,2>) &= x_1 e(B<3,5>)
\end{aligned}
$$

utilizando a estratégia Inside calculamos $e(B<3,5>)$, mas este termo já foi calculado na equação (5.2.7), portanto:

$$f(A < 1, 2 >) = x_1 x_5 x_7 x_4 x_6 x_8$$

então:

$$f(A < 2, 2 >) = x_3 x_1 x_5 x_7 x_4 x_6 x_8 x_6,$$

finalmente:

$$Pr(\chi|G_p) = f(A < 2, 2 >)x_4 = x_3 x_1 x_5 x_7 x_4 x_6 x_8 x_6 x_4 = x_1 x_3 x_4^2 x_5 x_6^2 x_7 x_8.$$

Observe que a probabilidade da cadeia χ calculada através das duas estratégias corresponde ao mesmo polinômio.