Markov Substitute Processes : a statistical model for linguistics

Thomas Mainguy

Université Pierre et Marie Curie

École doctorale de sciences mathématiques de Paris Centre

Département de Mathématiques et Applications de l'École Normale Supérieure
Équipe d'accueil : Probabilités et Statistiques

Processus de substitution Markoviens

un modèle statistique pour la linguistique

Markov Substitute Processes

a statistical model for linguistics

---

par Thomas Mainguy

Thèse de doctorat de Mathématiques

dirigée par Olivier Catoni

Présentée et soutenue publiquement le 11 décembre 2014
devant un jury composé de

M. Pierre Alquier	ENSAE	Examinateur
M. Gérard Biau	UPMC	Examinateur
M. Olivier Cappé	Telecom Paris Tech-CNRS	Examinateur
M. Olivier Catoni	ENS	Examinateur
M. Antoine Chambaz	Université Paris Ouest Nanterre	Examinateur
Mme Elisabeth Gassiat	Université Paris XI Orsay	Rapporteur
Mme Catherine Matias	CNRS	Examinateur

au vu des rapports de

Mme Elisabeth Gassiat	Université Paris XI Orsay
M. Ramon van Handel	Princeton University

Remerciements

Mes premiers remerciements vont à Olivier Catoni, pour m'avoir proposé ce sujet à la fin de mon stage de M2, rebondissant de manière inattendue sur une remarque en passant. Pour avoir su me guider comme il l'a fait, malgré les incertitudes et les difficultés initiales, m'avoir poussé à chercher toujours plus loin, et m'avoir montré ce qu'est le travail de chercheur.

Mes remerciements vont aussi à tous ceux qui m'ont mené à cette thèse, les excellents professeurs de probabilités et statistiques que j'ai eu lors de ma scolarité à l'École et à l'université d'Orsay, Wendelin Werner, Elisabeth Gassiat, Pascal Massart, Gilles Stoltz, ainsi que les linguistes du DEC qui m'ont initié à cette science qui m'attirait depuis tant de temps sans que je le sache, en particulier Dominique Sportiche et Edward Stabler (UCLA). Sans oublier mes professeurs de classes préparatoires Hélène Benhamou et Nicolas Tosel, qui m'ont montré la voie des mathématiques et sans qui je n'aurais sans doute pas eu la chance d'étudier à l'École Normale.

Merci enfin à tous ces non-matématiciens sans qui tout ceci n'aurait pas été possible, pour leur soutien et leur intérêt, à mes parents, qui m'ont guidé et soutenu, poussé et motivé, depuis toujours et particulièrement dans les passages difficiles de cette thèse, toujours là pour écouter mes déboires et victoires, même s'ils ne comprenaient pas tout... Merci aussi à tous mes camarades du club d'escrime, Raphaël, Samuel, Romain, et tous les autres, pour me suivre dans cette passion. Et à Émeline, qui m'a donné tant d'énergie et de motivation. Merci, du fond du cœur.

Abstract:

This thesis proposes a new approach to natural language processing. Rather than trying to estimate directly the probability distribution of a random sentence, we will detect syntactic structures in the language, which can be used to modify and create new sentences from an initial sample.

The study of syntactic structures will be done using Markov substitute sets, sets of strings that can be freely substituted in any sentence without affecting the whole distribution. These sets define the notion of Markov substitute processes, modelling conditional independence of certain substrings (given by the sets) with respect to their context. This point of view splits the issue of language analysis into two parts, a model selection stage where Markov substitute sets are selected, and a parameter estimation stage where the actual frequencies for each set are estimated.

We show that these substitute processes form exponential families of distributions, when the language structure (the Markov substitute sets) is fixed. On the other hand, when the language structure is unknown, we propose methods to identify Markov substitute sets from a statistical sample, and to estimate the parameters of the distribution. Markov substitute sets show some connections with context-free grammars, that can be used to help the analysis. We then proceed to build invariant dynamics for Markov substitute processes. They can among other things be used to effectively compute the maximum likelihood estimate. Indeed, Markov substitute models can be seen as the thermodynamical limit of the invariant measure of crossing-over dynamics.

Keywords: Markov processes; Natural language processing; Metropolis algorithm; PAC-Bayes hypothesis testing; Formal grammars; Statistical machine learning

Résumé :

Ce travail de thèse propose une nouvelle approche au traitement des langues naturelles. Plutôt qu'essayer d'estimer directement la probabilité d'une phrase quelconque, nous identifions des structures syntaxiques dans le langage, qui peuvent être utilisées pour modifier et créer de nouvelles phrases à partir d'un échantillon initial.
L'étude des structures syntaxiques est accomplie avec des ensembles de substitution Markoviens, ensembles de chaînes de caractères qui peuvent être échangées sans affecter la distribution. Ces ensembles définissent des processus de substitution Markoviens qui modélisent l'indépendance conditionnelle de certaines chaînes vis-à-vis de leur contexte. Ce point de vue décompose l'analyse du langage en deux parties, une phase de sélection de modèle, où les ensembles de substitution sont sélectionnés, et une phase d'estimation des paramètres, où les fréquences pour chaque ensemble sont estimées.
Nous montrons que ces processus constituent des familles exponentielles quand la structure du langage est fixée. Lorsque la structure du langage est inconnue, nous proposons des méthodes pour identifier des ensembles de substitution à partir d'un échantillon, et pour estimer les paramètres de la distribution. Les ensembles de substitution ont quelques relations avec les grammaires hors-contexte, qui peuvent être utilisées pour aider l'analyse.
Nous construisons alors des dynamiques invariantes pour les processus de substitution. Elles peuvent être utilisées pour calculer l'estimateur du maximum de vraisemblance. En effet, les processus de substitution peuvent être vus comme la limite thermodynamique de la mesure invariante d'une dynamique de crossing-over.

Mots-clefs : Processus markoviens ; Analyse des langues naturelles ; Algorithme Metropolis ; Tests d'hypothèse PAC-Bayesiens ; Grammaires formelles ; Apprentissage statistique

Contents

Introduction

0.1 Context

0.1.1 Natural language processing

The goal of this thesis is to propose some new theoretical tools for the statistical analysis of natural languages. This will lead us to introduce new models with a quite general purpose, that could be applied to approximate any process on a finite alphabet with a complex dependence structure.

A standard approach to language processing is to model a language as a distribution on sentences. The set of grammatical sentences is then given by the support of the distribution. As an alternative, computing the probability of a sentence provides an indication about its correctness with more gradations.

In such a framework, central questions are to represent and estimate the language distribution, to compute the probability of a sentence and to simulate from the estimated language. Since the support of a natural language is made of a huge number of grammatically correct sentences, an adaptive statistical approach, using a collection of parametric models of various dimensions, and going through the two steps of model selection and parameter estimation is required.

0.1.2 Remarks on the shortcomings of standard models

One approach to language processing is to use Markov chain based models. In these models, a chain of words is seen as the result of a production process that outputs words one at a time, depending only on the near past. In classical Markov chains, the nth word depends only on the $n-1$th, for example:

$$\mathbb{P}(\omega_n|\omega_1 \dots \omega_{n-1}) = \mathbb{P}(\omega_n|\omega_{n-1}).$$

Many variations on this pattern exists, as hidden Markov chains (where we only see a function of the words of a non observed Markov chain), or n-grams, where the mth word depends on the n previous ones. Context trees may also be used to model the dependence on the past context in a more flexible way, using a

suffix tree. More generally, the conditional probabilities may be defined with the help of any context function f with a finite range, according to the formula

$$\mathbb{P}(\omega_n|\omega_1 \dots \omega_{n-1}) = \mathbb{P}(\omega_n|f(\omega_1 \dots \omega_{n-1})).$$

Markov chains have been thoroughly studied, and as such are a good starting point to study many processes that output a chain of words – including natural languages.

However, Markov chains are missing a few important properties pertaining to the structure of natural languages. The description of a sentence likelihood as a product of forward conditional probabilities does not fit easily the long range backward and forward interactions described by linguistics. As a result, increasing the number of parameters (for instance the size of the context tree in a context tree model) to better fit the data, bumps into the curse of dimensionality, the number of parameters required for a decent fit precluding the possibility of a successful statistical estimation of those parameters.

0.1.3 Syntactic analysis

Linguists have developped several models to grasp the structure of a sentence, such as context-free grammars, minimalist grammars, head-driven phrase structure grammars, and many more. Many of these models use the notion of syntactic tree, analysing the structure of a sentence as a tree, each subtree representing a syntactic sub-unit of the sentence. For example, a (simplified) syntactic tree for the sentence "a cat chased the mouse" could be

[.VP [.DP [.D a] [.NP cat]] [.V' [.V chased] [.DP [.D the] [.NP mouse]]]]

In this example, the subtree "the mouse" forms a syntactic unit, a determiner phrase, as "chased the mouse", which is a verbal phrase.

The notion of phrases, or syntactic constituents, is fundamental in linguistics, as the structure of sentences seems to, in a first approach at least, follow simple rules if we consider them to be constituted by nested elements.

The first observation, which is actually the main tool used by syntacticians to identify these categories, is that some strings seem to be mostly interchangeable in a given language, without loss of grammaticality. For example, the two strings "a cat" and "the dog" are mostly interchangeable. In any sentence where "a cat" appears, we can substitute it by "the dog" and keep a grammatical sentence: they are both constituents of the same type, DPs. In the same way, "chased the mouse" and "slept" are both constituents of the same type (called V').

Syntactic relations can apparently reach over huge stretches of the sentence, while actually ignoring only a simple sub-tree. For example, relatives can be added

or subtracted and are effectively invisible to the rest of the sentence: "the dog (that Uncle Jim gave to the son of my neighbour) barked all night".

Other considerations, such as the fact that some constituents can "move" to other specific parts of the tree, add even more weight to such structures. In our work, however, we will mainly be interested in rigid trees, without movement. Since movement is rarely optional (at least in languages with poor flectional morphology), a rigid structure is not so great a restriction. A classic model using such rigid trees are context-free grammars, in which sentences are generated by rewriting non-terminal symbols. In our cat-and-mice example, DP may for example be rewritten as D and NP, D being in turn rewritten as "the" or "a", etc.

There is another property of natural languages which is quite dear to linguists, namely the recursivity of language, as typically examplified by "I tell you that John said that Mary thought that (...)". Such recursivity is very easily obtained by tree-like structures: in the case of context-free grammars, it suffices that a non-terminal symbol may be rewritten (after a number of interations) as itself.

One last property of natural languages that is closely linked to tree-structure and syntactic constituents is dependency. As we hinted before, whole sub-trees can be virtually invisible from the rest of the sentence. One notion that goes in the same direction is that of syntactic head. From outside of a constituent, only its "head" (usually one word) is visible (to check grammaticality). To go back to our example of the barking dog, the head of the constituent "the dog that Uncle Jim gave to the son of my neighbour" is "the dog" (the actual head is theoretically "the", but let's not delve into the linguistics depths here). As such, we could replace the rest of the syntagm by anything we could want (for example, "the dog *of my aunt*" and, provided that the result is still a correct nominal syntagm, put it in any sentence were the original syntagm was grammatical, and it would still be ("the dog of my aunt barked all night").

This indicates that legitimate substitutions may be conditioned by relations between constituents and contexts depending only on appropriate surface labels.

Of course, the theoretical properties of natural language structures are still under heavy research, and the models we briefly exposed here are far from being the latest in terms of linguistic advancement. Even some notions presented here are currently questioned, as the rigid structure of head-context dependencies. However, one cannot deny that theories such as $\bar{X}$, mainly used for the presentation here, are very good first approximations for the analysis of natural languages. They may fail to grasp some intricate subtleties, but still do a good job at providing a rough global picture.

The Markov models presentend in section 0.1.1 on page 1 fail to grasp the recursive nature of natural languages, as well as many syntactic properties, such as far-reaching relations. As such, many efforts have been made to incorporate

syntactic structures in the models (Della Pietra et al. 1994 [DPDPG+94]; B. Roark 2001 [Roa01]; M. Tan et al. 2012 [TZZW12], to cite only a few). The question of learning the structure of the syntax has been also considered, for example by A. Clark [Cla14].

0.1.4 Markov substitute processes and toric grammars in a few words

Our proposal follows the trend described in the previous section, but with a change of perspective. While trying some variations on the theme of Markov models on data, we realized that modeling dependencies in natural languages through an improvement of the Markov model was a daunting task. Therefore at some point, we decided to depart from conventional modeling.

In the classical statistical approach, the tabula rasa is in some sense that all words are independent. In statistical words, this corresponds to taking as the default model the independent word model, or Markov chain of order zero. Starting from there, we may build models that describe some possible dependencies adding progressively to the set of parameters of the independent model. This works only with limited success, the structure of natural languages being too flexible to be grasped by rigid parametric models, defined prior to examining data. Using adaptive model selection may help (as in context tree weighting estimators), but does not solve the problem entirely.

So we propose to go in the opposite direction. This means taking as our tabular rasa the hypothesis that all words are dependent. Accordingly, our default model is the multinomial distribution on the language support. Using this default model, we can only output from any i.i.d. data sample the trivial distribution estimate consisting in the empirical sample distribution.

Starting from the multinomial model, we incorporate incremental conditional independence assumptions to decrease progressively the dimension of the model. These conditional independence assumptions are related to the fact that some constituents may be substituted for one another independently of the value of the context. As we remove parameters in this way, we can build distribution estimators whose support is spread further and further beyond the support of the empirical sample distribution.

We first implemented this idea by describing, with the help of a new type of grammar, that we called toric grammars, what we called sample level kernel estimates. These are language distribution estimates that produce a sentence distribution by recombining the sentences of the statistical sample, performing randomly a set of allowed constituent substitutions. This can be described as applying to the sample a Markov kernel with interactions between sentences, hence

the name sample level kernel estimate. Technically, the distribution estimate is the invariant measure of this Markov kernel, restricted to the communicating class of the statistical sample. This can be seen as some extension of classical kernel density estimates, where a smoothing Markov kernel is applied independently at each point of the statistical sample.

At first, we were not able to provide a principled statistical theory to choose this sample level kernel, (that is to choose the substitution rules), and relied on some heuristic criterion that gave satisfactory empirical results. We then realized that we could relate the sample level kernel estimates of the first chapter to a full fledged statistical model, that in fact appears as a very natural extension of Markov random field models. We called this model Markov substitute processes. In the second chapter, we describe those models in terms of conditional independence assumptions and address the two questions of model selection through multiple statistical tests, and of parameter estimation.

We show that once a set of conditional independence assumptions has been chosen, this results in a parametric model that can be described as an exponential family. The construction of the corresponding potential function is given by a theorem that extends in some way the theorem of Hammersley and Clifford [Bes74] to our model but that does not provide an efficient algorithm to compute the potential in practice (except for toy examples). We show on the other side that the conditional independence assumptions can be tested by a collection of simultaneous tests with a controlled complexity. The model complexity in these tests is handled in a data dependent way, using PAC-Bayesian theorems (a theory first proposed by D. McAllester, see [McA98, McA99]).

We then explored the combinatorics of our conditional independence assumptions. Each assumption is described by a substitute set : a list of constituents that can be substituted independently of the context, according to a substitute probability measure. A Markov substitute model is thus described by a finite family of Markov substitute sets, the set of parameters being the parameters of the substitute measures. Interaction between Markov substitute sets means that we need much less parameters than the number of actual members of each set to describe the substitute measures.

Any family of substitute sets defines a non trivial Markov substitute model, that is a non void exponential family (this is one of our results). Moreover, there is a combinatorics on Markov substitute sets : the fact that some sets are Markov substitute sets implies that other sets are also Markov substitute sets. This combinatorics is related to context-free grammars. Starting from a finite family of Markov substitute sets, we can build a context-free grammar, such that the languages that are generated choosing each non terminal symbol in turn as the start symbol forms a broader collection of Markov substitute sets. This broader collec-

tion is not the largest one. In fact, if we include one point sets in the definition of Markov substitute sets, it is possible to show that the family of maximal Markov substitute sets forms a partition of the sentence space. Anyhow we do not have an efficient algorithms to compute this partition, so that parsing algorithms for context-free grammars are at present our best offer to compute a large amount of Markov substitute sets.

This large collection can be used to build more efficient sample level kernel estimates. On the other hand, the support of our language distribution is not necessarily itself a language generated by a context-free grammar, and can be much bigger. Context-free grammars are used as a tool to manipulate a large family of substitute sets, allowing to simulate faster Monte Carlo sampling dynamics and to compute more efficiently the probability of a single sentence. The interplay between grammars and Markov substitute processes is the subject of the third chapter.

In this chapter, we also describe properly balanced sample level kernel estimates, based on random crossing over between sample sentences or based on more sophisticated and faster mixing sentence recombinations using grammatical parsing. By properly balanced, we mean kernels that are reversible with respect to the sample distribution (that a tensor product of the language distribution).

Since we are dealing with models forming exponential families, we know that the maximum likelihood estimate in each such model has good off-model properties. Namely, it will converge to the projection according to the Kullback-Leibler distance of the actual data distribution on the model, at a rate that depends on the rate of estimation of the expectation of each component of the potential function. The problem is that, as long as we are not able to compute explicitly the potential function, we are not able to compute the maximum likelihood estimator either.

To solve this problem, we show that properly balanced sample level kernel estimates have a thermodynamical limit, when applied to n replicas of the statistical sample, when n tends to infinity, that is equal to the maximum likelihood estimate of the parameters of the corresponding Markov substitute model. This means that, although we cannot provide an efficient algorithm to compute the potential function that describes a Markov substitute process model as an exponential family, we still have a feasible Monte-Carlo algorithm to compute the maximum likelihood estimator in this model. This shows that the model can be used not only on data that would exactly match the required assumptions, but that the model can be used on any data distribution to provide an approximation of this distribution, in the same way as high order Markov chains can be used to approximate non Markovian stationary ergodic processes.

0.2 Overview of the main results

We will now present a quick overview of the notions encountered and the main results of this work. In what follows, we will consider a text sample $S_{1:n}$ of n sentences on a dictionary D. It will sometimes be convenient, since we do not attempt to model interactions between sentences in this work, to see this sample (and any text), as an empirical distribution on $D^+ = \bigcup_{j=1}^{\infty} D^j$, so the order of sentences does not count.

0.2.1 Toric grammars

We began with what we called a communication model, in the form of a kernel modifying sentences and texts according to a certain type of interaction. This point of view was inspired by the linguistic notion of syntactic categories, seen as chains of words that can be substituted regardless of their contexts.

The main tool we developed to build these sample level kernels was toric grammars, a notion closely related to context-free grammars. Indeed, the set $\mathfrak{G}$ of toric grammars is defined in definition 1.2.1 on page 22 as the set of positive measures on context-free rewriting rules $\mathscr{E}$. Using this definition, we can see the set of texts $\mathfrak{T}$ as a subset of the toric grammar set $\mathfrak{G}$, whose support is included in the rewriting rules for the start symbol. For ease of notation, we write the non-terminal symbols as $]_i$, $i \in \mathbb{N}$, and for the left-hand side of production rules, we replace the usual notation $]_i \longrightarrow$ by $[_i$. The start symbol $]_0$ is supposed to never occur in the right-hand side of the rules (as in some normal forms of context-free grammars), so only the symbol $[_0$ appears.

The main difference between toric grammars and context-free grammars could be roughly summarized by the fact that the context-free rewriting rules of toric grammars are more viewed as context-free substitution rules (replace any string that can be generated by a given non-terminal symbol by any other, regardless of the context), than rewriting rules.

These grammars give us a method to modify a text (seen as an empirical measure on sentences, and thus also a toric grammar) by substituting substrings by others. In order to do so, we have to make a distinction between what we chose to call a reference grammar $\mathscr{R}$, used to describe the model of the language considered, and "regular" toric grammars $\mathscr{G}$, obtained from parsing a certain text using the rules of a reference grammar in a possibly random manner.

The reference grammar can thus be seen as a collection of context-free rewriting rules, *without a start symbol*. For these, (in chapter 1, at least), the weights are not important. From this reference grammar, we will define in section 1.3.2 on page 28 a random split-and-merge process G_t, that will swap the occurences of constituents of same type in the text.

This process is implemented in two steps, each step being a time homogeneous stopped Markov chain. The first step, the splitting process S_t, is the decomposition of the initial text into the rules of $\mathscr{R}$ used to produce the text, giving a regular toric grammar. It is stopped at stopping time τ, when no more authorized splits are available. The second step, the production process P_t, is the converse operation of using the same rules back to generate a new text, effectively mixing the constituents of the initial text into a new one. It is stopped at stopping time σ, when there is no possible move left.

As such, the split-and-merge process G_t is a Markov chain on grammars, a grammar on even time being a text produced by the grammar of the previous time using a production process, and a grammar on odd time being a parse of the previous text, obtained with a splitting process.

We show in proposition 1.3.4 on page 30 that both the splitting and the production processes take finite time, with a deterministc bound depending on the size of the initial text. This justifies the definition of the split-and-merge process, whose transitions are then computable in finite time.

Proposition
Let $(S_t)_{t\leqslant\tau}$, $(P_t)_{t\leqslant\sigma}$ and (G_t) be a splitting process, a production process and the corresponding split and merge process, starting from $G_0 = \mathscr{T} \in \mathfrak{T}$. For any $\mathscr{G} \in \mathfrak{G}$, any $\mathscr{T}' \in \mathfrak{T}$, such that $\sum_{t\in\mathbb{N}} \mathbb{P}(G_{2t+1} = \mathscr{G}) > 0$ and $\sum_{t\in\mathbb{N}} \mathbb{P}(G_{2t} = \mathscr{T}') > 0$,

$$\mathbb{P}\Big(\tau \leqslant 2\big[\mathscr{T}\big(DS^*\big) - \mathscr{T}\big(\lfloor_0 S^*\big)\big] \,\Big|\, S_0 = \mathscr{T}'\Big) = 1,$$
$$\mathbb{P}\Big(\sigma \leqslant 2\big[\mathscr{T}\big(DS^*\big) - \mathscr{T}\big(\lfloor_0 S^*\big)\big] \,\Big|\, P_0 = \mathscr{G}\Big) = 1.$$

In other words, the length of all the splitting and production processes involved in the split and merge process have a uniform bound, given by twice the difference between the number of words and the number of sentences in the original text.

Moreover, we prove in proposition 1.4.2 on page 36 that this process is weakly reversible.

Proposition
Given a parsing process S_t based on a reference grammar $\mathscr{R} \in \mathfrak{G}$ and a production process P_t, the corresponding split and merge process G_t is weakly reversible, in the sense that for any $\mathscr{T} \in \mathfrak{T}$, any $\mathscr{G} \in \bigcup_{t\in\mathbb{N}} \operatorname{supp}\big(\mathbb{P}_{G_{2t+1}}\big)$,

$$\mathbb{P}\Big(G_1 = \mathscr{G} \,\Big|\, G_0 = \mathscr{T}\Big) > 0 \iff \mathbb{P}\Big(G_2 = \mathscr{T} \,\Big|\, G_1 = \mathscr{G}\Big) > 0.$$

Consequently, for any $\mathscr{T}, \mathscr{T}' \in \mathfrak{T}$ and any $\mathscr{G}, \mathscr{G}' \in \bigcup_{t\in\mathbb{N}} \operatorname{supp}\big(\mathbb{P}_{G_{2t+1}}\big)$,

$$\mathbb{P}\Big(G_2 = \mathscr{T}' \,\Big|\, G_0 = \mathscr{T}\Big) > 0 \iff \mathbb{P}\Big(G_2 = \mathscr{T} \,\Big|\, G_0 = \mathscr{T}'\Big) > 0,$$
$$\mathbb{P}\Big(G_3 = \mathscr{G}' \,\Big|\, G_1 = \mathscr{G}\Big) > 0 \iff \mathbb{P}\Big(G_3 = \mathscr{G} \,\Big|\, G_1 = \mathscr{G}'\Big) > 0.$$

In other words, the two processes G_{2t} and G_{2t+1} are weakly reversible time homogeneous Markov chains.

As we will also prove that the set of reachable states from any starting point is finite, this proposition shows that the split and merge process defines two recurrent Markov chains, one on texts and one on grammars. These two chains partition their respective state spaces into positive recurrent communicating classes.

This means that the split and merge process can be seen as a communication process, where each speaker, upon hearing a given text, learns a grammar that could have produced it, and utters back a new text to a new speaker, and so forth. What this result shows is that if the reference grammar (which can be seen as an internal language model) is common to all speakers, the asymptotic language, that is, the distribution of $\lim \frac{1}{T} \sum_{t \leqslant T} G_{2t+1}$, is uniquely determined by the recurrent class of the initial text.

The main difference of this approach from usual context-free grammars is that the rules are weighted by a counting measure, counting how many times each rule is to be applied. This means that, for example, the size, vocabulary, etc. remains constant between texts in the same communicating class. A second difference is that the process G_t, going from texts to texts, is iterated. As such, the texts produced are produced by the context-free rules of the reference grammar, complemented by a set of rewriting rules for the start symbol, that may not be the same for all texts in the same communicating class. We will see in section 1.6.2 on page 39 that in some cases this gives a wider generated language than what can be obtained using a context-free grammar.

0.2.2 Markov substitute sets and processes

The study of the first chapter leaves the question of the statistical estimation of the reference grammar essentially unsolved, although encouraging preliminary experimental results could be obtained using some heuristic rules of thumb to learn the reference grammar.

This question led us to relate our toric grammars to more classical statistical models, described in chapters 2 and 3.

In chapter 2, we begin by defining in definition 2.1.1 on page 64 the notion of Markov substitute sets, as a possible formalization of syntactic constituents. These sets are sets of strings that can be substituted in any sentence, regardless of their contexts. In more formal terms, a Markov substitute set B for a distribution $\mathbb{P}_S$ has a substitute measure q_B such that, for any context (x, y), and any string $b \in B$,

$$\mathbb{P}_S(xby) = \mathbb{P}_S(xBy)q_B(b).$$

While this notion is rooted in linguistic considerations, it is in no way restricted to linguistics, and could have applications in other domains. It can indeed be seen as

a generalization of the (one dimensional) Markov field conditional independence property, that corresponds to the case when B is of the form $w_1 D w_2$, where w_1 and w_2 are two boundary words.

To any collection of Markov substitute sets $\mathscr{B}$ corresponds an equivalence relation that relates two strings $s \sim_{\mathscr{B}} s'$ if and only if s and s' can be reached from one another by a sequence of substitutions authorized by $\mathscr{B}$. The components of D^+ for this equivalence relation are all Markov substitute sets.

Conversely, we can define a $\mathscr{B}$-Markov substitute process (for a given finite collection of finite sets $\mathscr{B}$) as any process S such that each set of $\mathscr{B}$ is a Markov substitute set (technically, we have to require also that each element of each set of $\mathscr{B}$ has a positive probability to appear in a sentence). We will prove in section 2.3 on page 74 that Markov substitute processes actually exist for any collection of Markov substitute sets, and that the set of $\mathscr{B}$-Markov substitute processes is an exponential family (technically a union of exponential families, depending on the support of the process).

Proposition
Given any finite set $\mathscr{B}$ of finite subsets of D^+, there is a finite set of pairs, $\mathscr{P}$, that we can choose such that each one is included in a member of $\mathscr{B}$, such that the sets of $\mathscr{B}$-Markov and $\mathscr{P}$-Markov substitute processes are the same, and such that $\mathscr{P}$ is minimal for the inclusion relation (removing a pair from $\mathscr{P}$ would break the above property).

For any $\mathscr{B}$-Markov substitute process S, we can decompose its support as a union of components $\mathscr{C} \subset {}^{D^+}/_{\sim_{\mathscr{B}}}$, such that $\operatorname{supp}(\mathbb{P}_S) = \bigcup \mathscr{C}$ *and such that*

$$\mathscr{B} = \bigcup_{C \in \mathscr{C}} \mathscr{B}_C.$$

For any such subset, there is a free, non empty subset of pairs $\mathscr{F} \subset \mathscr{P}$, a matrix $\Big(e_{j,i}, j \in \mathscr{P} \setminus \mathscr{F}, i \in \mathscr{F}\Big)$, an index set $I(\mathscr{C})$ containing $\mathscr{F}$ and an energy function $\Big(U_i(s), i \in I(\mathscr{C}), s \in C\Big)$, such that the set of $\mathscr{B}$-Markov processes whose support is $\bigcup \mathscr{C}$ is the linear exponential family

$$\mathfrak{M}_{\mathscr{C}}(\mathscr{B}) = \left\{ \left(p(s) = \frac{\exp\Big(- \sum_{i \in I(\mathscr{C})} \beta_i \, U_i(s) \Big)}{\sum_{s' \in \bigcup \mathscr{C}} \exp\Big(- \sum_{i \in I(\mathscr{C})} \beta_i \, U_i(s') \Big)}, s \in \bigcup \mathscr{C} \right), \beta \in \mathfrak{B} \subset \mathbb{R}^{I(\mathscr{C})} \right\},$$

where

$$\mathfrak{B} = \left\{ \beta \in \mathbb{R}^{I(\mathscr{C})}, \sum_{s \in \bigcup \mathscr{C}} \exp\Big(- \sum_{i \in I(\mathscr{C})} \beta_i U_i(s) \Big) < \infty \right\},$$

and such that for any member p of this family, the substitute measure of any pair $i = \{y_{i,0}, y'_{i,1}\} \in \mathscr{F}$ (taken in a suitable order compatible with the definition of U), is given by

$$q_i(y_{i,\sigma}) = \frac{\exp(\sigma\beta_i)}{1+\exp(\sigma\beta_i)}, \qquad \sigma \in \{0,1\},$$

whereas for $j \in \mathscr{P} \setminus \mathscr{F}$,

$$q_j(y_{i,\sigma}) = \frac{\exp(\sigma\beta_j)}{1+\exp(\sigma\beta_j)}, \qquad \sigma \in \{0,1\},$$

where

$$\beta_j = \sum_{i \in \mathscr{F}(C)} e_{j,i}\, \beta_i.$$

We can see here that the problem of analysing any language using this framework can be decomposed into two stages:

- first a model selection stage, where a good collection $\mathscr{B}$ of Markov substitute sets is to be selected,
- and then a parameter estimation stage, in which the substitute measures q_B, for each $B \in \mathscr{B}$ are to be estimated. (We could in principle estimate only the parameters of the exponential family, but the above theorem does not provide an efficient algorithm to compute the energy function U explicitly, so that this will not be feasible, except for some toy models.)

Since Markov substitute sets are a possible formalization of syntactic constituents, we will see in section 3.1 on page 119 that it is possible to relate them to context-free rules. Conversely, a set of context-free rules defines a collection of Markov substitute sets, as the collection of the languages generated by each non-terminal symbol. This duality gives us quite efficient ways to analyse any sentence in terms of Markov substitute sets, using parsing algorithms, as described in section 3.B on page 159.

0.2.3 Model selection: finding a collection of Markov substitute sets

The question is to find Markov substitute sets, or, equivalently, context-free rules.

A good starting point is that two strings form a Markov substitute set if their relative frequencies do not depend on their context. This observation can be extended to two sets of strings, for which we know that each already forms a

Markov substitute set. This can be used to build several test functions to test whether the union of two Markov substitute sets $B_1 \cup B_2$ is still a Markov substitute set. We explore two main categories of possible test functions.

The first category involves the simulation of a parse (X, Y) of a sentence S, so that $S = \alpha(X, Y)$ (meaning that S is the string where Y appears in context X), and $Y \in B_1 \cup B_2 = B$. These tests are based on the empirical mean of test functions of the form

$$F_\theta(X, Y, p) = \mathbb{1}(X \in \theta)\Big[\mathbb{1}(Y \in B_1) - p\, \mathbb{1}(Y \in B_1 \cup B_2)\Big],$$

where $\theta \in \Theta$ is a context type.

The second category is built by computing explicitly conditional expectations with respect to the raw (non parsed) sentence sample. It is of the type

$$F_{B_1,B_2,\theta}(s, p) = \mathbb{E}\Big\{\Big[\mathbb{1}\Big(Y_{B_1\cup B_2} \in B_1\Big) - p\, \mathbb{1}\Big(Y_{B_1\cup B_2} \in B_1 \cup B_2\Big)\Big]\mathbb{1}\Big(X_{B_1\cup B_2} \in \theta\Big)\,\Big|\, S = s\Big\}.$$

The goal is to find test functions $F_{B_1,B_2,\theta}(S, p)$ such that, regardless of the context type θ,

$$\mathbb{E}\Big(F_{B_1,B_2,\theta}(S, p)\Big) = 0$$

when $p = q_{B_1\cup B_2}(B_1)$, in the case when $B_1 \cup B_2$ is indeed a Markov substitute set. In other words, if we define

$$p_+ = \sup\Big\{\mathbb{P}\Big(Y \in B_1 \,|\, Y \in B, X = x\Big); x \in (D^*)^2, \mathbb{P}(Y \in B, X = x) > 0\Big\},$$
$$p_- = \inf\Big\{\mathbb{P}\Big(Y \in B_1 \,|\, Y \in B, X = x\Big); x \in (D^*)^2, \mathbb{P}(Y \in B, X = x) > 0\Big\},$$

the test function allows us to test whether $p_+ = p_-$. A third interpretation is that the test functions define $p(B_1, B_2, \theta)$ so that

$$\mathbb{E}\Big[F\Big(S, p(B_1, B_2, \theta)\Big)\Big] = 0,$$

and we want to test whether $p(B_1, B_2, \theta)$ is independent of θ.

For both of these types of test function, we can define similar tests, for which we can prove accurate bounds for the simultaneous probability of both false rejection and false acceptance.

For the first type of test function, for example, we can prove the following result (propositions 2.6.1 and 2.6.3 on page 89 and on page 92)

Proposition
Let $\mu \in \mathscr{M}_+^1$ be a probability measure depending only on $\Big(X_i, \mathbb{1}(Y_i \in B),\, i \in [\![1, n]\!]\Big)$. Let us define

$$F_{\theta,i}(p) = F_\theta(X_i, Y_i, p).$$

Let Λ be a finite subset of $]0,1[$. With probability at least $1-2\varepsilon$,

$$B_-(p_+) \stackrel{\text{def}}{=} \sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda} \int \sum_{i=1}^n \log\Big(1+\lambda F_{\theta,i}(p_+)\Big)\mathrm{d}\rho(\theta) - \mathscr{K}(\rho,\mu) - \log\Big(|\Lambda|/\varepsilon\Big) \leqslant 0,$$

$$B_+(p_-) \stackrel{\text{def}}{=} \sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda} \int \sum_{i=1}^n \log\Big(1-\lambda F_{\theta,i}(p_-)\Big)\mathrm{d}\rho(\theta) - \mathscr{K}(\rho,\mu) - \log\Big(|\Lambda|/\varepsilon\Big) \leqslant 0,$$

where

$$\mathscr{K}(\rho,\mu) = \begin{cases} \displaystyle\int \log\Big(\frac{\mathrm{d}\rho}{\mathrm{d}\mu}\Big)\,\mathrm{d}\rho, & \text{when } \rho \ll \mu, \\ +\infty, & \text{otherwise.} \end{cases}$$

Therefore, if we reject the hypothesis that B is a Markov substitute set when

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0,$$

the probability of false rejection is at most 2ε.

Let us assume in addition that the number of context types that can be encountered in the language is bounded by m, and that μ is the uniform probability measure on the finite set of encountered context types $\widehat{\Theta}$. Let us put

$$\delta = \frac{\log(mn) + 2\log(\varepsilon^{-1}) + \log(|\Lambda|)}{n},$$

and

$$\chi = \sup_{x\,\in\,[n^{-1/2},\,n^{1/2}]} \inf_{\lambda\,\in\,\Lambda} \cosh\Big[\log\Big(\frac{\lambda}{(1-\lambda)x}\Big)\Big].$$

Let us assume that there are θ_+ and $\theta_- \in \Theta$ such that $\overline{p}_+ = p(\theta_+)$, $\overline{p}_- = p(\theta_-)$, $q_+ = \mathbb{E}\Big[\mathbb{1}(Y\in B)\mathbb{1}(X\in\theta_+)\Big]$ and $q_- = \mathbb{E}\Big[\mathbb{1}(Y\in B)\mathbb{1}(X\in\theta_-)\Big]$ satisfy

$$\begin{aligned} \min\{q_-, q_+\} &\geqslant 8\chi^2\delta, \\ \overline{p}_+ - \overline{p}_- &\geqslant 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1+\frac{4\chi^2\delta}{q_+}\right) + \frac{(2+\sqrt{2})\delta}{q_+} \\ &\quad + 2\chi\sqrt{\frac{\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1+\frac{4\chi^2\delta}{q_-}\right) + \frac{(2+\sqrt{2})\delta}{q_-}. \end{aligned}$$

Then with probability at least $1-2\varepsilon$

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0.$$

Therefore, the test has in this case a probability of false acceptance at most equal to 2ε.

A similar result about the test functions that do not simulate the parse is proved in propositions 2.7.3 and 2.7.7 on page 101 and on page 112. The test proposed in this case also comprises a uniform bound on all tests on pairs of Markov substitute sets chosen in a finite collection $\mathscr{B}$. We propose a novel PAC-Bayes complexity measure, based on the Kullback divergence between a posterior distribution and another observable reference distribution, that plays the role of a prior, but involves only the contexts appearing in the statistical sample. This observable complexity factor is bounding the more classical divergence of the posterior from a prior, according to eq. (2.7.4) on page 98.

We can remark at this point that these tests give a way to test if the union of two Markov substitute sets is still one, without having to actually compute the whole extent of each set. This is interesting, because, if we take the rewriting rule view, the new Markov substitute set can then be described as a simple rewriting rule $B_1 \cup B_2 \longrightarrow B_1|B_2$. This mechanism gives actually a method to build recursively a complete toric grammar, for which the distribution (provided no test failed) is a Markov substitute process.

0.2.4 Parameter estimation

The second chapter proposes tests to select a particular model described by a collection of Markov substitute sets. The next question is, given this collection, how to estimate the parameters q_B for each Markov substitute set in the collection.

While the prospect of estimating the parameter $q_B(x)$ for each member x of B is quite daunting, we will see that we can use the grammar structure to greatly simplify this estimation. In order to understand this result, we first need to see that collection of Markov substitute sets and reference grammars are closely related. Indeed, we can define for any string e of symbols the language set B_e as the set of strings that can be generated from e. We will see in section 3.1 on page 119 that this set B_e is a Markov substitute set, as soon as each $B_i \stackrel{\text{def}}{=} B_{]_i}$ are Markov substitute sets themselves. We call such a grammar a Markov grammar. Conversely, if we build a collection of Markov substitute sets using the tests of chapter 2, we can define a corresponding Markov grammar, as described in section 3.4 on page 144.

Another useful notion is that of parse trees $t \in \mathscr{T}$, defined in section 3.1.2 on page 122. They are parenthetized strings in $\breve{D}^* = (D \cup \{(_i,)_i, i \in \mathbb{N}\})^*$, such that the content y of each pair parentheses $(_i y)_i$ is a member of B_i, that is, a string in the language generated by the grammar with $]_i$ as start symbol. The structure of a given tree gives a very simple decomposition of the string in terms of Markov substitute sets. The surface structure $\varsigma(t)$ of t, which is simply the string t where all pairs of parentheses (and their contents) $(_i y)_i$ are replaced by the corresponding non-terminal symbol $]_i$, gives one language that can generate $\varphi(t)$ (t whithout

parentheses), that is, $\varphi(t) \in B_{\varsigma(t)}$. Using these notions, we will show the following result, lemma 3.2.1 on page 125, giving a formula for any substitute measure:

Lemma
For any $s \in D^+$, and any $t \in \mathscr{T}$ such that $s = \varphi(t)$,

$$q_{B_{\varsigma(t)}}(s) = A_{\varsigma(t)} \prod_{{}_j e \in \mathscr{R}} \Big[A_e q_{B_j}\Big(B_e\Big)\Big]^{\chi(t,j,e)},$$

where

$$\chi(t,j,e) = \sum_{x \in (\breve{D}^*)^2} \sum_{y \in \mathscr{T}} \mathbb{1}\Big[t = \alpha\Big(x, ({}_j y)_j\Big)\Big] \mathbb{1}\Big(\varsigma(y) = e\Big).$$

The advantage of this result is that we need only to compute the parameters A_e and $q_{B_j}(B_e)$ for expressions e that are actually in the grammar, and not for all members of the Markov substitute sets.

The sections 3.2.2 and 3.2.3 on page 127 and on page 129 use this result to propose methods to estimate there parameters and simulate q_B.

0.2.5 Invariant dynamics

The last part of chapter 3 is about three types of invariant dynamics, that we can define with the previous tools.

The first one is a dynamics on sentences, and requires the knowledge of the substitute measures q_B. This dynamics is defined in section 3.2.4 on page 131, and simply identifies members of a Markov substitute sets B in the sentence, and replaces it by another member of B according to q_B.

The two other dynamics work on whole texts, but do not require actual knowledge of the substitute measures q_B.

The second method, described in section 3.5.1 on page 151, is actually a new take on the split and merge process defined in chapter 1, using a Metropolis algorithm to make it reversible.

The third method, described in section 3.2.5 on page 134, is a sort of crossing-over dynamics, that will simply swap elements of the same set found in the text.

These three methods are all made reversible using a slightly more general (to our knowledge) method for the Metropolis algorithm, that allows for an arbitrary number of intermediate steps between the argument and the result. This method is described in section 3.D on page 165.

This last dynamics also gives an interesting view on Markov substitute processes. We mentioned that these could in fact be described as an exponential family. This means that the maximum likelihood estimate for any sample will converge to the projection according to the Kullback Leibler distance on the model.

The problem is that since the energy function is hard to compute, the projection is also difficult to find.

However, we will be able to prove in section 3.3 on page 135 that the crossing-over dynamics from any sample does indeed converge to the projection (with some additional hypotheses, of course).

If we write K_m a crossing-over dynamics on texts of size nm, we can define, for any text $s_{1:n}$, S^m a random uniform shuffle of m copies of $s_{1:n}$, and the distribution

$$p_m = \lim_{t\to\infty} \frac{1}{t} \sum_{u=1}^{t} \mathbb{P}_{S^m} K_m^u.$$

We can then define the random measure

$$N_m = \sum_{i=1}^{nm} \delta_{\overline{S}_{m,i}} \in \mathscr{M}_+(\mathscr{D}),$$

where $\big(\overline{S}_{m,i}, 1 \leqslant i \leqslant nm\big)$ is a random variable distributed according to p_m. Simply put, N_m is the empirical distribution of the crossing-over of m copies of the initial text, randomly shuffled.

We then will prove in proposition 3.3.3 on page 143 that

Proposition
Under some hypotheses described at the beginning of section 3.3 on page 135, the maximum likelihood

$$\sup_{\beta\in\mathbb{R}^I} \sum_{j=1}^{n} \log\big[p_\beta(s_i)\big]$$

is reached at some $\beta_* \in \mathbb{R}^I$. For any $\varepsilon > 0$, there are $\eta > 0$ and $M > 0$ such that for any $m \geqslant M$,

$$\mathbb{P}\Big[d\big(N_m, p_{\beta_*}\big) \geqslant \varepsilon\Big] \leqslant \exp(-nm\eta),$$

where $d(\mu,\nu) = \sqrt{\sum_{s\in\mathscr{D}} \big(\mu(s) - \nu(s)\big)^2}$. Moreover,

$$\lim_{m\to\infty} \mathbb{E}\big(N_m/(nm)\big) = p_{\beta_*}.$$

This means that we can actually project any sample on the Markov substitute model using a crossing-over dynamics, and thus that Markov substitute processes can be seen as thermodynamical limits of crossing-over dynamics.

Chapter 1

Toric grammars and communication models

1.1 Introduction to a new communication model

In the well known kernel approach to density estimation on a measurable space $\mathscr{X}$, the probability distribution $\mathbb{P}$ of a random variable $X \in \mathscr{X}$ is estimated from a statistical sample $(X_1, \ldots, X_n)$ of n independent copies of X as $\frac{1}{n}\sum_{i=1}^n k(X_i, \mathrm{d}x)$, where k is a suitable Markov kernel (also known as a conditional probability kernel). This kernel estimate can be seen as a modification of the empirical measure $\overline{\mathbb{P}} = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$.

Finding sensible kernel estimates or sensible parametric models in the context of natural language processing is a challenge. Therefore, we propose here another route, that we will describe as an alternative way of producing a modification of the empirical measure. The idea is to recombine repeatedly a set of sentences. Let us describe for this a general framework, concerned with an arbitrary countable state space $\mathscr{X}$.

Let

$$\overline{\mathscr{P}}_n = \left\{ \frac{1}{n}\sum_{i=1}^n \delta_{x_i},\ x_i \in \mathscr{X} \right\}$$

be the set of empirical measures of all possible samples of size n. Let us consider a parametric family $\{q_\theta, \theta \in \Theta\}$ of Markov kernels on $\overline{\mathscr{P}}_n$. Let us assume for simplicity that for any $P \in \overline{\mathscr{P}}_n$, the reachable set $\left\{Q \in \overline{\mathscr{P}}_n,\ \sum_{t\in\mathbb{N}} q_\theta^t\Big(P, Q\Big) > 0\right\}$ is finite, where q_θ^t is q_θ composed t times with itself, $t \in \mathbb{N}$, so that for instance $q_\theta^2(P, Q) = \sum_{P' \in \overline{\mathscr{P}}_n} q_\theta(P, P') q_\theta(P', Q)$. In this case we can define the Markov kernel

$$\widehat{q}_\theta(P, Q) = \lim_{k\to\infty} \frac{1}{k}\sum_{t=1}^k q_\theta^t(P, Q).$$

It is such that for any $P \in \overline{\mathscr{P}}_n$, $\widehat{q}_\theta(P, \cdot)$ is an invariant measure of q_θ. More generally $q_\theta \widehat{q}_\theta = \widehat{q}_\theta q_\theta = \widehat{q}_\theta$. The distribution $\widehat{q}_\theta(P, \cdot) \in \mathscr{M}_+^1\left(\overline{\mathscr{P}}_n\right)$ induces a marginal distribution $\widehat{Q}_{\theta,P}$ on $\mathscr{X}$ through the formula

$$\widehat{Q}_{\theta,P} = \sum_{Q \in \overline{\mathscr{P}}_n} \widehat{q}_\theta(P, Q) Q. \tag{1.1.1}$$

We will here be concerned with estimators of the form $\widehat{\mathbb{P}} = \widehat{Q}_{\theta,\overline{\mathbb{P}}}$, if θ is fixed in advance, or of the form $\widehat{Q}_{\widehat{\theta},\overline{\mathbb{P}}}$, if $\widehat{\theta}$ is an estimator of the parameter θ depending also on $\overline{\mathbb{P}}$.

Another interpretation of our framework is to consider q_θ as a communication model. One speaker hears a set of sentences described by its empirical distribution $P \in \overline{\mathscr{P}}_n$ (which means that he will not make use of the special order in which he has heard them). He uses those sentences to learn the corresponding language. Then he teaches another speaker what he has learnt by outputting another random set of sentences, distributed according to $q_\theta(P, \cdot)$. The language model (as opposed to the communication model q_θ), is $\widehat{Q}_{\theta,P}$, the average sentence distribution along an infinite chain of communicating speakers.

If we start from a recurrent state P, and we assume that θ is known, we obtain a communication model where the target sentence distribution $\widehat{Q}_{\theta,P}$ can be learnt without error from the set of sentences output by any involved speaker. Indeed $\widehat{Q}_{\theta,P} = \widehat{Q}_{\theta,Q}$ for any Q in the communicating class of P, which in this situation is also the reachable set from P.

This error free estimation behaviour is desirable for a communication model. It tells us that the language can be transmitted from speaker to speaker without distortion, a desirable feature in the case of a large number of speakers. The model may also account for weak stimulus learning, the fact that human beings learn language through a limited number of examples compared with the variety of new sentences they are able to formulate. Indeed, whereas the size of the support of $P \in \overline{\mathscr{P}}_n$ (the number of sentences heard by one speaker) is constant and equal to n, the support of the language model $\widehat{Q}_{\theta,P}$ may be much larger. We will actually give a toy example where the number of sentences in the language is exponential with n.

Remark on the other hand that if the parameter θ is not known, but rather estimated at each step from the heard sample, the error free property is obviously not true anymore. This view of communication by iterated learning, where a internal grammar θ is learned at each generation and used to produce a learning sample for the next generation, is not new, and has been studied, for example, by Griffiths and Kalish [GK07] to form models of language evolution.

In the language transmission interpretation, we may evaluate the interest of the model by studying whether it can model a large family of sentence distributions.

This richness will depend on the number of recurrent communicating classes of the communication Markov model q_θ, since any invariant distribution $\widehat{q}_\theta(P,\cdot)$ is a convex combination of the unique invariant measures supported by each recurrent communicating class. The situation is even simpler in the case when all $P \in \overline{\mathscr{P}}_n$ are recurrent states (a fact we will be able to prove in our particular model). In this case $\widehat{q}_\theta(P,\cdot)$ is the unique invariant measure supported by the recurrent communicating class to which P belongs.

The parameter θ of our model will be a new kind of grammar, closely related to context-free grammars, in that it will use rewriting rules, but used to generate sentences in a different way. These toric grammars will be defined in section 1.2 below as a weighted collection of rewriting rules, not unlike stochastic context-free grammars.

We will relate the building of a communication kernel presented in this chapter with the construction of more classical statistical models in the following chapters of this work.

1.2 First definitions

1.2.1 Toric grammars

We consider some finite dictionary of words D, and a random sentence S, that is a random sequence of words of random finite length. We will use the notation

$$D^+ = \bigcup_{j=1}^{\infty} D^j$$

for the set of sequences of words of finite positive length. As explained in the introduction, S, or equivalently its probability distribution $\mathbb{P}_S \in \mathscr{M}_+^1\left(D^+\right)$, will be our mathematical representation of a language.

From this language, we will observe a statistical sample of sentences $S_1, \ldots, S_n$, made of n independent copies of S.

In this chapter, we will build estimators by applying a Markov kernel to the empirical sample distribution. To describe this construction, let us introduce the state space of counting measures of weight n as

$$\overline{\mathscr{P}}_n = \left\{ \sum_{i=1}^{n} \delta_{s_i}, s_i \in D^+ \right\}.$$

(Let us remark that in this definition, where δ_{s_i} is the Dirac mass at s_i, the same sentence can be repeated more than once.)

We will call $\overline{\mathscr{P}}_n$ the set of texts of length n. Let us notice that for us, texts are unordered sets of sentences (with possible repetitions). The question of generating meaningful ordered sequences of sentences is also of interest, but will not be addressed in this study.

In order to define a language estimator, we will first define a Markov kernel q (that is a conditional probability kernel) on the state space $\overline{\mathscr{P}}_n$. We will construct this kernel in such a way that the reachable sets $\Big\{Q \in \overline{\mathscr{P}}_n, \sum_{t=0}^{\infty} q^t(P,Q) > 0\Big\}$ are finite (for any starting point P), so that we can define a limit Markov kernel

$$\widehat{q}(P,Q) = \lim_{k\to\infty} \frac{1}{k} \sum_{t=1}^{k} q^t(P,Q),$$

and take as our language estimator

$$\widehat{\mathbb{P}} = \frac{1}{n} \sum_{Q\in\overline{\mathscr{P}}_n} \widehat{q}\Big(\sum_{i=1}^{n} \delta_{S_i}, Q\Big) Q.$$

In order to define the communication kernel q, we will describe random transformations on texts, related to the notion of context-free grammars. Let us start with an informal presentation. The communication kernel will perform random recombinations of sentences.

Our point of view is to see a context-free grammar as the result of some fragmentation process applied to a set of sentences. Let us explain this on a simple example. Consider the sentence

This is my friend Peter.

Imagine we would like to represent this sentence as the result of pasting the expression *my friend* in its context, because we think language is built by cutting and pasting expressions drawn from some large sets of memorized expressions. We can do this by introducing the simple context-free grammar

$$\boxed{0} \to \textit{This is } \boxed{1} \textit{ Peter.}$$
$$\boxed{1} \to \textit{my friend}$$

where we have used numbered framed boxes for non terminal symbols, the start symbol being $\boxed{0}$. The two rules mean that we can rewrite the start symbol $\boxed{0}$ to obtain the right-hand side of the first rule, and that we can then rewrite the non terminal symbol $\boxed{1}$ as the right-hand side of the second rule.

Since we want to see the rules of the grammar as the result of some splitting operation, we are going to use more symmetric notations. Instead of considering that we have described our original sentence with the help of two rules and two non

terminal symbols $\boxed{0}$ and $\boxed{1}$, we may as well consider that we have split our original sentence into two new sentences using *three* non terminal symbols, namely $\boxed{0}\rightarrow$, $\boxed{1}$ and $\boxed{1}\rightarrow$. To emphasize this interpretation, we can adopt more symmetric notations and write these three non terminal symbols as $[_0$, $]_1$ and $[_1$. With these new notations, the representation of our original sentence is now

$$\begin{array}{l}[_0\ \textit{This is}\ ]_1\ \textit{Peter .}\\ [_1\ \textit{my friend}\end{array}$$

In this new representation, the rewriting rules can be replaced by merge operations of the type

$$a\,]_i c + [_i\, b \ \mapsto\ abc$$

We can make this merge operations even more symmetric, if we consider that each expression can be represented by any of its circular permutations. Indeed, each expression contains exactly one non terminal symbol of the form $[_i$, and therefore is uniquely defined by any of its circular permutations (since, due to this feature, we can define the permutation in which the opening bracket $[_i$ comes first as the canonical form, and recover it from any other circular permutation). Using this convention, we can write $a\,]_i c$ as $ca\,]_i$ (note that a starts with an opening bracket and contains no other opening bracket, whereas c does not contain any opening bracket) and describe the merge operation as

$$ca\,]_i + [_i\, b \ \mapsto\ cab,$$

or, renaming ca as a, simply as

$$a]_i + [_i b \ \mapsto\ ab.$$

Let us formalize what we have explained so far. Let D be some dictionary of words (which can be for the sake of this mathematical description any finite set, representing the words of the natural language to be modeled). Let us form the symbol set $S = D \cup \Big\{[_i,]_i, i \in \mathbb{N}\Big\}$. Let us define the set of circular permutations of a sequence of symbols as

$$\mathfrak{S}(w_0, \ldots, w_{\ell-1}) = \Big\{(w_{(i+j \bmod \ell)}, i = 0, \ldots, \ell-1), j = 0, \ldots, \ell-1\Big\},$$

so that for instance $\mathfrak{S}(w_0, w_1, w_3) = \{w_0w_1w_2, w_1w_2w_0, w_2w_0w_1\}$. Let us define its support (the set of symbols included in the sequence) as

$$\mathrm{supp}\Big(w_0, \ldots, w_{\ell-1}\Big) = \Big\{w_0, \ldots, w_{\ell-1}\Big\}.$$

Let $A^+ = \bigcup_{n=1}^{+\infty} A^n$, $]_+ = \Big\{\ ]_i, i \in \mathbb{N} \setminus \{0\}\Big\}$, and consider the set of expressions

$$\mathscr{E} = \Big\{\ e \in \mathfrak{S}\Big([_i\, a\Big),\ i \in \mathbb{N},\ a \in \Big(D \cup\ ]_+\Big)^+ \ \setminus\ ]_+\Big\}.$$

In plain words, an expression is a circular permutation of a finite sequence of symbols starting with an opening bracket, containing no other opening bracket and not reduced to an opening bracket followed by a closing bracket.

This definition mirrors the fact that a given rule of a context-free grammar has exactly one $\boxed{i} \to$ (the left side), and the right side of the rule cannot be just a non terminal symbol $\boxed{j}$. Indeed, if we had allowed $\boxed{i} \to \boxed{j}$, or with our notations $[_i\,]_j$, we could as well have replaced j by i everywhere.

We can now define the notion of toric grammars, which will be the central tool in all of this chapter, both to describe the language structure, and to define the transformations on texts.

Definition 1.2.1
The set of toric grammars is the set $\mathfrak{G}$ of positive measures $\mathscr{G}$ on $\mathscr{E}$ with finite support such that for any circular permutation $e' \in \mathfrak{S}(e)$ of any expression $e \in \mathscr{E}$, $\mathscr{G}(e') = \mathscr{G}(e)$.

In other words, a toric grammar $\mathscr{G}$ is a positive measure with finite support on the set of expressions $\mathscr{E}$ satisfying

$$\mathscr{G}(e) = |\mathfrak{S}(e)|^{-1}\mathscr{G}\Big(\mathfrak{S}(e)\Big).$$

Let us remark that, in our definition of toric grammars, on top of choosing some special notations for context-free grammars, we also introduced positive weights, so that it is more the support of a toric grammar than the grammar itself that corresponds to the usual notion of context-free grammar, while a toric grammar in its entirety corresponds to a stochastic context-free grammar.

The weights will serve to keep track of word frequencies through the process of splitting a set of sentences to obtain a toric grammar.

Our aim is indeed to build a toric grammar from a text. To be consistent with our definition of grammars, we will also define texts as positive measures (only on whole sentences, so that no non-terminal other than $[_0$ is used). Let us give a formal definition. We will forget the sentence order, a text will be an unordered set of sentences with possible repetitions.

Definition 1.2.2
The set $\mathfrak{T}$ of texts is the set of toric grammars with integer weights supported by $\mathfrak{S}\,([_0\,D^+)$, that is the set of toric grammars with integer weights using only one non terminal symbol, the start symbol $[_0$.

In this definition, it should be understood that

$$[_0\,D^+ = \Big\{\Big([_0, w_1, \ldots, w_k\Big),\ \text{where } k \in \mathbb{N} \setminus \{0\} \text{ and } w_i \in D, 1 \leqslant i \leqslant k\Big\},$$

and that

$$\mathfrak{S}\Big([_0\,D^+\Big) = \bigcup_{e \in [_0\,D^+} \mathfrak{S}(e).$$

1.2.2 A roadmap towards a communication model

We will use toric grammars as intermediate steps to define the transition probabilities of our communication model on texts. To this purpose, we will first introduce some general types of transformations on toric grammars (reminding the reader that in our formalism texts are some special subset of toric grammars).

It will turn out that two types of expressions, global expressions and local expressions, will play different roles. Let us define them respectively as

$$\mathscr{E}_g = \mathscr{E} \cap \mathfrak{S}\Big([_0\, S^+\Big),$$
$$\mathscr{E}_\ell = \mathscr{E} \cap \mathfrak{S}\Big([_+\, S^+\Big),$$

where we remind that $[_+ = \Big\{[_i, i \in \mathbb{N} \setminus \{0\}\Big\}$ and $S^+ = \bigcup_{j=1}^{\infty} S^j$. Roughly speaking, global expressions correspond to full sentences, whereas local expressions correspond to partial constituents. Any toric grammar $\mathscr{G} \in \mathfrak{G}$ can be accordingly decomposed into $\mathscr{G} = \mathscr{G}_g + \mathscr{G}_\ell$, where $\mathscr{G}_g(A) = \mathscr{G}\Big(A \cap \mathscr{E}_g\Big)$ and $\mathscr{G}_\ell(A) = \mathscr{G}\Big(A \cap \mathscr{E}_\ell\Big)$, for any subset $A \subset \mathscr{E}$.

The transitions of the communication chain with kernel $q_\theta(\mathscr{T}, \mathscr{T}')$ will be defined in two steps. The first step consists in learning from the text $\mathscr{T}$ a toric grammar $\mathscr{G}$. To this purpose we will split the sentences of $\mathscr{T}$ into syntactic constituents. The second step consists in merging the constituents again to produce a random new text $\mathscr{T}'$. The parameter $\theta = \mathscr{R}$ of the communication kernel q_θ, will also be a toric grammar. The role of this reference grammar $\mathscr{R}$ will be to provide a stock of local expressions to be used when computing $\mathscr{G}$ from $\mathscr{T}$. We will discuss later the question of the estimation of $\mathscr{R}$ itself, (and bring satisfactory statistical answers to this issue only in the forthcoming chapters). For the time being, we will assume that the reference grammar $\mathscr{R}$ is a parameter of the communication chain, known to all involved speakers.

We could have defined a communication kernel $q_{\widehat{\mathscr{R}}(\mathscr{T})}(\mathscr{T}, \mathscr{T}')$, where the reference grammar $\widehat{\mathscr{R}}(\mathscr{T})$ itself is estimated at each step from the current text $\mathscr{T}$, but we would have obtained a model with weaker properties, where, in particular, all the states are not necessarily recurrent states. On the other hand, the proof that the reachable set from any starting point is finite still holds for this modified model, so that it does provide an alternative way of defining a language model as described in the introduction.

We will still need an estimator $\widehat{\mathscr{R}}(\mathscr{T})$ of the reference grammar, in order to provide a language estimator $\widehat{Q}_{\widehat{\mathscr{R}}(\mathscr{T}),\mathscr{T}'}$, where we are using the notations of eq. (1.1.1) on page 18. We will propose a first tentative method for the estimation $\widehat{\mathscr{R}}(\mathscr{T})$ of the reference grammar in this chapter, and study this question in more details in the two following ones.

1.3 Operations on toric grammars

1.3.1 Non stochastic syntax splitting and merging

Let us now describe the model, starting with the description of some non random grammar transformations. We already introduced a model for grammars that includes texts as a special case. We have now to describe how to generate a toric grammar from a text, with, or without, the help of a reference grammar to learn the local component of the grammar. The mechanism producing a grammar from a text will be some sort of random parse algorithm (or rather tentative parse algorithm).

All of this will be achieved by two transformations on toric grammars that will respectively *split* and *merge* expressions (syntagms) of a toric grammar into smaller or bigger ones. We will first describe the sets of possible splits and merges from a given grammar. This will serve as a basis to define random transitions from one grammar to another in subsequent sections.

Let us first introduce some elementary operations involving toric grammars.

$$\begin{aligned} e \oplus f &= \sum_{s\in\mathfrak{S}(e)} \delta_s + \sum_{s\in\mathfrak{S}(f)} \delta_s, && e, f \in \mathscr{E}, \\ e \ominus f &= \sum_{s\in\mathfrak{S}(e)} \delta_s - \sum_{s\in\mathfrak{S}(f)} \delta_s, && e, f \in \mathscr{E}, \\ \rho \otimes e &= \rho \sum_{s\in\mathfrak{S}(e)} \delta_s, && \rho \in \mathbb{R}, e \in \mathscr{E}, \end{aligned}$$

The first operation builds a toric grammar containing expressions e and f with weights 1, and the third one builds a toric grammar containing expression e with weight ρ.

We can generalize these notations to be able to take the sum of a toric grammar and an expression, as well as the sum of two toric grammars.

$$\begin{aligned} \mathscr{G} \oplus e &= \mathscr{G} + \sum_{s\in\mathfrak{S}(e)} \delta_s, && \mathscr{G} \in \mathfrak{G}, e \in \mathscr{E} \\ \mathscr{G} \ominus e &= \mathscr{G} - \sum_{s\in\mathfrak{S}(e)} \delta_s, && \mathscr{G} \in \mathfrak{G}, e \in \mathscr{E} \\ \mathscr{G} \oplus \mathscr{G}' &= \mathscr{G} + \mathscr{G}', && \mathscr{G}, \mathscr{G}' \in \mathfrak{G}. \end{aligned}$$

With these notations, a split is described as

$$\mathscr{G}' = \mathscr{G} \ominus ab \oplus a]_i \oplus [_i b, \qquad \mathscr{G}, \mathscr{G}' \in \mathfrak{G},$$

the fact that $\mathscr{G}, \mathscr{G}' \in \mathfrak{G}$ implying that

$$i \in \mathbb{N} \setminus \{0\}, ab, a]_i, [_i b \in \mathscr{E} \text{ and } \mathscr{G}(ab) \geqslant 1.$$

The (partial) order relation $\mathscr{G} \leqslant \mathscr{G}'$ will also be defined by the rule

$$\mathscr{G} \leqslant \mathscr{G}' \iff \mathscr{G}' - \mathscr{G} \in \mathfrak{G},$$

or equivalently

$$\mathscr{G} \leqslant \mathscr{G}' \iff \mathscr{G}' - \mathscr{G} \in \mathscr{M}_+(\mathscr{E}).$$

Let us resume our example. Starting from the one sentence text

$$\mathscr{T} = 1 \otimes [_0 \ \textit{This is my friend Peter .}$$

we get after splitting the grammar

$$\mathscr{G} = [_0 \ \textit{This is} \ ]_1 \ \textit{Peter .} \oplus [_1 \ \textit{my friend}$$

which can also be written as

$$\mathscr{G} = \ \textit{Peter .} \ [_0 \ \textit{This is} \ ]_1 \oplus [_1 \ \textit{my friend}$$

In this example, as well as in the following, punctuation marks are treated as words, so that here the required dictionary has to include the set

$$\{\textit{is, friend, my, Peter, This, .}\}.$$

Splitting a sentence providing a new label for each split does not create generalization, since it allows only to merge back two expressions that came from the same split. To create a grammar capable of yielding new sentences, we need some label identification scheme. We will perform label identification through the more general process of label remapping, identification being a consequence of the fact that the map may not be one to one. Let

$$\mathfrak{F} = \Big\{ f : \mathbb{N} \to \mathbb{N} \text{ such that } f(0) = 0 \Big\}$$

be the set of label maps. For any symbol $]_i$ or $[_i$, let us define $f(]_i) =]_{f(i)}$ and $f([_i) = [_{f(i)}$. Let us also define for any word $w \in D$, $f(w) = w$ and for any expression $e = (w_0, \ldots, w_{\ell-1})$, $f(e) = (f(w_0), \ldots, f(w_{\ell-1}))$. Since any grammar $\mathscr{G} \in \mathfrak{G}$ is a measure on the set of expressions $\mathscr{E}$, we can define its image measure by f, considered as a map from $\mathscr{E}$ to $\mathscr{E}$. We will put $f(\mathscr{G}) = \mathscr{G} \circ f^{-1}$, meaning that $f(\mathscr{G})(A) = \mathscr{G}(f^{-1}(A))$, for any subset $A \subset \mathscr{E}$.

Definition 1.3.1
Two label maps f and $g \in \mathfrak{F}$ are said to be isomorphic if there is a one to one label map $h \in \mathfrak{F}$ such that $g = h \circ f$. In this case $h^{-1} \in \mathfrak{F}$ and $f = h^{-1} \circ g$. Two grammars $\mathscr{G}$ and $\mathscr{G}' \in \mathfrak{G}$ are said to be isomorphic if there is a one to one label map $f \in \mathfrak{F}$ such that $f(\mathscr{G}) = \mathscr{G}'$. In this case, $f^{-1}(\mathscr{G}') = \mathscr{G}$ and we will write $\mathscr{G} \equiv \mathscr{G}'$. If f and g are two isomorphic label maps, then for any toric grammar $\mathscr{G} \in \mathfrak{G}$, $f(\mathscr{G})$ and $g(\mathscr{G})$ are isomorphic grammars.

In the following of this work, to ease notations and simplify exposition, we will freely identify isomorphic label maps and isomorphic grammars and often speak of them as if they were equal.

This being put, we proceed with the introduction of a set of grammar transformations β that consist in a split with possible label remapping. The *split* will be the core component for generating a toric grammar from a text, by splitting the sentences in smaller parts (syntagms).

Definition 1.3.2 (Splitting rule)
For any $\mathscr{G} \in \mathfrak{G}$, let us consider

$$\beta(\mathscr{G}) = \left\{ f(\mathscr{G}'), f \in \mathfrak{F}, \mathscr{G}' \in \mathfrak{G}, \mathscr{G}' = \mathscr{G} \ominus ab \oplus a\rceil_i \oplus \lfloor_i b \right\} \subset \mathfrak{G}.$$

Let us remark that in this definition, necessarily, $ab, a\rceil_i, \lfloor_i b \in \mathscr{E}$, $i \in \mathbb{N} \setminus \{0\}$, $1 \otimes ab \leqslant \mathscr{G}$, and $a\rceil_i \oplus \lfloor_i b \leqslant \mathscr{G}'$. Let us put

$$\beta^*(\mathscr{G}) = \bigcup_{n=0}^{+\infty} \underbrace{\beta \circ \cdots \circ \beta}_{n \text{ times}}(\mathscr{G}),$$

the set of grammars that can be constructed from repeated invocations of β.

Lemma 1.3.1
Let us recall that $S = D \cup \left\{ \lfloor_i, \rceil_i, i \in \mathbb{N} \right\}$ and let us put $S^* = \bigcup_{n=0}^{+\infty} S^n$. For any text $\mathscr{T} \in \mathfrak{T}$, and any $\mathscr{G} \in \beta^*(\mathscr{T})$, $\mathscr{G}$ is a toric grammar with integer weights,

$$\begin{aligned}
\mathscr{G}(\lfloor_i S^*) &= \mathscr{G}(\rceil_i S^*), && i \in \mathbb{N} \setminus \{0\}, \\
\mathscr{G}(wS^*) &\leqslant \mathscr{T}(wS^*), && w \in \left(D \cup \{\lfloor_0\}\right)^+, \\
\mathscr{G}(wS^*) &= \mathscr{T}(wS^*), && w \in \left(D \cup \{\lfloor_0\}\right),
\end{aligned}$$

and in particular

$$\mathscr{G}(\lfloor_0 S^*) = \mathscr{T}(\lfloor_0 S^*).$$

This means that in any toric grammar obtained by splitting a text, the weights of expressions containing the two forms $\rceil_i$ and $\lfloor_i$ of a label are balanced, the word frequencies are the same in the grammar and in the text, and the number of sentences contained in the text is given by the total weight of expressions containing the start symbol $\lfloor_0$ in the grammar.

Proof. For the first assertion, an induction on the number of applications of β yields the result, since

$$\mathscr{T}([_i S^*) = \mathscr{T}(]_i S^*) = 0, i \in \mathbb{N} \setminus \{0\},$$

and, for any $\mathscr{G}' = \mathscr{G} \ominus ab \oplus a\,]_i \oplus [_i\, b$, and any label $j \in \mathbb{N} \setminus \{0, i\}$,

$$\mathscr{G}'([_j S^*) = \mathscr{G}([_j S^*), \tag{1.3.1}$$
$$\mathscr{G}'(]_j S^*) = \mathscr{G}(]_j S^*), \tag{1.3.2}$$

whereas

$$\mathscr{G}'([_i S^*) = \mathscr{G}([_i S^*) + 1, \tag{1.3.3}$$
$$\mathscr{G}'(]_i S^*) = \mathscr{G}(]_i S^*) + 1. \tag{1.3.4}$$

For the second assertion, it suffices to remark that the weight of expressions beginning with a given sequence of words is not increased by application of β. Indeed, any word sequence $w \in (D \cup \{[_0)^+\}$ appears the same number of times at the beginning of an expression of $1 \otimes ab$ and of $a\,]_i \oplus [_i\, b$ if w is not split between a and b and appears a smaller number of times if w is split between a and b. When w is a single word, it cannot be split, so we have equality, as stated in the third assertion of the lemma. The subsequent application of a label map f to $\mathscr{G}'$ does not change the counts involved in the lemma. □

This lemma is important, because we will subsequently want to impose restrictions on the splitting rule based on word frequencies. Our choice to define a new type of grammar as a positive measure on symbol sequences was made to keep track of word frequencies throughout the construction.

Let us now describe the reverse of a splitting transformation, that we will call a merge transformation. This transformation will be central in generating new texts from a toric grammar, by merging the syntagms into bigger ones, ending with a full sentence.

Definition 1.3.3 (Merge rule)
For any toric grammar $\mathscr{G} \in \mathfrak{G}$ we consider the following set of allowed merge transformations

$$\alpha(\mathscr{G}) = \left\{ \mathscr{G}' \in \mathfrak{G}, \mathscr{G}' = \mathscr{G} \ominus a\,]_i \ominus [_i\, b \oplus ab \right\}.$$

Let us remark that in this definition, necessarily $i \in \mathbb{N} \setminus \{0\}$, $a\,]_i, [_i\, b, ab \in \mathscr{E}$, and $a\,]_i \oplus [_i\, b \leqslant \mathscr{G}$.

The merge transformation is indeed the reverse of the *split*, in the sense that:

Lemma 1.3.2
For any $\mathscr{G}, \mathscr{G}' \in \beta^*(\mathfrak{T})$, $\mathscr{G}' \in \beta(\mathscr{G})$ if, and only if, there is $f \in \mathfrak{F}$ such that $f(\mathscr{G}) \in \alpha(\mathscr{G}')$.

PROOF. Let us suppose that $\mathscr{G}' = f\Big(\mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab\Big)$ is in $\beta(\mathscr{G})$. In this case, $\mathscr{G}' = f(\mathscr{G}) \oplus f(a\,]_i) \oplus f([_i\, b) \ominus f(ab)$, so that both $f(a\,]_i)$ and $f([_i\, b)$ are in $\mathrm{supp}(\mathscr{G}')$, $f(ab) \in \mathrm{supp}\Big(f(\mathscr{G})\Big)$, and consequently $f(a\,]_i), f([_i b)$ and $f(ab) \in \mathscr{E}$. Moreover $f(\mathscr{G}) = \mathscr{G}' \oplus f(a)f(b) \ominus f(a)\,]_{f(i)} \ominus [_{f(i)}\, f(b)$, so that $f(\mathscr{G}) \in \alpha(\mathscr{G}')$.

On the other hand, if for some $f \in \mathfrak{F}$, $f(\mathscr{G}) \in \alpha(\mathscr{G}')$, $f(\mathscr{G}) = \mathscr{G}' \oplus ab \ominus a\,]_i \ominus [_i\, b$. Since $ab \in \mathrm{supp}\Big(f(\mathscr{G})\Big)$, there is $e \in \mathscr{E}$ such that $f(e) = ab$. But this implies that there is $c, d \in S^+$ such that $a = f(c)$ and $b = f(d)$. We can then if needed modify f outside $\Big\{j \in \mathbb{N} : [_j\, S^* \in \mathrm{supp}(\mathscr{G})\Big\}$, to make sure that $i \in f(\mathbb{N})$. Let $f(j) = i$. We now get that $f(\mathscr{G}) = \mathscr{G}' \oplus f(c)f(d) \ominus f(c)\,]_{f(j)} \ominus [_{f(j)}\, f(d)$, so that finally $\mathscr{G}' = f\Big(\mathscr{G} \oplus c\,]_j \oplus [_j\, d \ominus cd\Big)$, proving that $\mathscr{G}' \in \beta(\mathscr{G})$. □

Another useful property of the merge rule is given by the following lemma:

Lemma 1.3.3
For any $f \in \mathfrak{F}$ and any $\mathscr{G} \in \mathfrak{G}$, $f\Big(\alpha(\mathscr{G})\Big) \subset \alpha\Big(f(\mathscr{G})\Big)$.

PROOF. Indeed, any $\mathscr{G}' \in f\Big(\alpha(\mathscr{G})\Big)$ is of the form

$$\begin{aligned}\mathscr{G}' &= f\Big(\mathscr{G} \oplus ab \ominus a\,]_i \ominus [_i\, b\Big) \\ &= f(\mathscr{G}) \oplus f(a)f(b) \ominus f(a)\,]_{f(i)} \ominus [_{f(i)}\, b \in \alpha\Big(f(\mathscr{G})\Big).\end{aligned}$$ □

Unfortunately, repeating the merge transformation will not provide a text in all circumstances. Indeed, we can end up with some expressions of the type $[_i\, a\,]_i b$. However, since an expression is allowed to contain only one opening bracket, we are sure that $[_0 \notin \mathrm{supp}([_i\, a\,]_i\, b)$, and that all sentences (global expressions, beginning with $[_0$) are "complete", in the sense that they do not contain any other non-terminal symbol.

To continue the discussion, we will switch to a random context, where split and merge transformations are performed according to some probability measure.

1.3.2 Random split and merge processes

The grammars we described so far are obtained using splitting rules. Texts can be reconstructed using merge transformations. The splitting rules as well as the merge rules allow for multiple choices at each step. We will account for this by introducing random processes where these choices are made at random.

We will describe two types of random grammar transformations. Each of these will appear as a finite length Markov chain, where the length of the chain is given by a uniformly bounded stopping time.

- The learning process (or splitting process) will start with a text and build a grammar through iterated splits;
- the production process will start with a grammar and produce a text through iterated merge operations.

These two types of processes may be combined into a split and merge process, going back and forth between texts and toric grammars.

Let us give more formal definitions. Learning and parsing processes will be some special kinds of splitting processes, to be defined hereafter.

Definition 1.3.4 (Splitting process)
Given some restricted splitting rule $\beta_r : \mathfrak{G} \to 2^{\mathfrak{G}}$ *from the set of grammars to the set of subsets of* $\mathfrak{G}$*, such that for any* $\mathscr{G} \in \mathfrak{G}$*,* $\beta_r(\mathscr{G}) \subset \beta(\mathscr{G})$*, a* splitting process *is a time homogeneous stopped Markov chain* $S_t, 0 \leqslant t \leqslant \tau$ *defined on* $\mathfrak{G}$ *such that*

$$\tau = \inf\Big\{t \in \mathbb{N} : \beta_r(S_t) = \varnothing\Big\},$$

$$\mathbb{P}\Big(S_t = \mathscr{G}' \,|\, S_{t-1} = \mathscr{G}\Big) > 0 \iff \mathscr{G}' \in \beta_r(\mathscr{G}).$$

Definition 1.3.5 (Production process)
A production process *is a time homogenous stopped Markov chain* $P_t, 0 \leqslant t \leqslant \sigma$ *defined on* $\mathfrak{G}$ *such that*

$$\sigma = \inf\Big\{t \in \mathbb{N}, \alpha(P_t) = \varnothing\Big\},$$

and

$$\mathbb{P}\Big(P_t = \mathscr{G}' \,|\, P_{t-1} = \mathscr{G}\Big) > 0 \iff \mathscr{G}' \in \alpha(\mathscr{G}).$$

Definition 1.3.6 (Split and Merge process)
Given a splitting process $S_t, t \in \mathbb{N}$ *and a production process* $P_t, t \in \mathbb{N}$*, a* split and merge process *is a Markov chain* $G_t \in \mathfrak{G}$*,* $t \in \mathbb{N}$*, with transitions*

$$\mathbb{P}\Big(G_{2t+1} = \mathscr{G}' \,|\, G_{2t} = \mathscr{G}\Big) = \mathbb{P}\Big(S_\tau = \mathscr{G}' \,|\, S_0 = \mathscr{G}\Big), \qquad t \in \mathbb{N},$$
$$\mathbb{P}\Big(G_{2t} = \mathscr{G}' \,|\, G_{2t-1} = \mathscr{G}\Big) = \mathbb{P}\Big(P_\sigma = \mathscr{G}' \,|\, P_0 = \mathscr{G}, P_\sigma \in \mathfrak{T}\Big), \qquad t \in \mathbb{N} \setminus \{0\},$$

whose initial distribution is a probability measure on texts, so that almost surely $G_0 \in \mathfrak{T}$*.*

Let us remark that we have to impose the condition that $P_\sigma \in \mathfrak{T}$, because the production process does not produce a true text with probability one. On the other hand it can yield back G_{2t-2} with positive probability when started at G_{2t-1}, as will be proved later on. Therefore $\mathbb{P}(P_\sigma \in \mathfrak{T} \,|\, P_0 = \mathscr{G}) > 0$ for any $\mathscr{G}$ such that $\mathbb{P}(G_{2t-1} = \mathscr{G}) > 0$. One way to simulate $\mathbb{P}_{G_{2t}\,|\,G_{2t-1}}$ is to use a rejection method, simulating repeatedly from the production process until a true text is produced. In the experiments we made, $\mathbb{P}\Big(P_\sigma \in \mathfrak{T} \,|\, P_0 = \mathscr{G}\Big)$ was close to one and rejection a rare event. We will describe the relations between these processes and parsing rules issued from the statistical model of the following chapters in section 3.5 on page 151.

Proposition 1.3.4
Let S_t, P_t and G_t be a splitting process, a production process and the corresponding split and merge process, starting from $G_0 = \mathscr{T} \in \mathfrak{T}$. For any $\mathscr{G} \in \mathfrak{G}$, any $\mathscr{T}' \in \mathfrak{T}$, such that $\sum_{t\in\mathbb{N}} \mathbb{P}(G_{2t+1} = \mathscr{G}) > 0$ and $\sum_{t\in\mathbb{N}} \mathbb{P}(G_{2t} = \mathscr{T}') > 0$,

$$\mathbb{P}\Big(\tau \leqslant 2\Big[\mathscr{T}\Big(DS^*\Big) - \mathscr{T}\Big(\big[_0\, S^*\Big)\Big]\,\Big|\, S_0 = \mathscr{T}'\Big) = 1, \tag{1.3.5}$$

$$\mathbb{P}\Big(\sigma \leqslant 2\Big[\mathscr{T}\Big(DS^*\Big) - \mathscr{T}\Big(\big[_0\, S^*\Big)\Big]\,\Big|\, P_0 = \mathscr{G}\Big) = 1. \tag{1.3.6}$$

In other words, the length of all the splitting and production processes involved in the split and merge process have a uniform bound, given by twice the difference between the number of words and the number of sentences in the original text.

PROOF. This proof is a bit lengthy and is based on some invariants in the split and merge operations. It has been put off to section 1.A.1 on page 49. □

Proposition 1.3.5
If G_t is a split and merge process starting almost surely from the text $G_0 = \mathscr{T} \in \mathfrak{T}$, there is a finite subset of toric grammars $\mathfrak{G}_\mathscr{T}$ such that with probability equal to one there is for each time t a grammar G'_t isomorphic to G_t such that $G'_t \in \mathfrak{G}_\mathscr{T}$. Thus, after identification of isomorphic grammars, we can analyze the split and merge process as a finite state Markov chain, since the reachable set from any starting point is finite. We should however keep in mind that the finite state space $\mathfrak{G}_\mathscr{T}$ depends on the initial state $\mathscr{T}$, so the state space is still infinite, although any trajectory will almost surely stay in a finite subset of reachable states.

PROOF. Let us assume that the labels of $\mathscr{G}$ are taken from $[\![0, W_\ell(\mathscr{G})]\!]$, where, as defined in the appendix, $W_\ell(\mathscr{G}) = \sum_{i=1}^{\infty} \mathscr{G}(\big[_i\, S^*)$ is the number of labels (with their multiplicities) used in the grammar $\mathscr{G}$. Consequently $\mathscr{G}(\big[_i\, S^*) = 0$ for $i > W_\ell(\mathscr{G})$. This can be achieved, up to grammar isomorphisms, by applying to $\mathscr{G}$ a suitable

label map.
Let us define the set of canonical expressions as

$$\mathscr{E}_c = \mathscr{E} \cap \left(\bigcup_{i \in \mathbb{N}} [_i \, S^* \right),$$

and the canonical decomposition of $\mathscr{G}$ as

$$\mathscr{G} = \sum_{e \in \mathscr{E}_c} \mathscr{G}(e) \otimes e.$$

We see that $\mathscr{G}$ can be described by the concatenation of the canonical expressions, each repeated a number of times equal to its weight, to form a sequence of symbols of length $W_s(\mathscr{G})$. From the proof of the previous proposition, we know that

$$W_s(\mathscr{G}) \leqslant M = 5W_w(\mathscr{T}) - 3W_e(\mathscr{T}) = 5\mathscr{T}(DS^*) - 3\mathscr{T}([_0 S^*),$$

(with notations defined in the appendix). We can represent $\mathscr{G}$ by a sequence of exactly M symbols by padding with trailing $[_0$ symbols the representation described above. Let us give an example

$$\mathscr{G} = 2 \otimes [_0 \, w_1 \,]_1 w_2 \oplus [_1 \, w_3 \oplus [_1 \, w_4$$

can be coded as

$$[_0 \, w_1 \,]_1 w_2 \, [_0 \, w_1 \,]_1 w_2 \, [_1 \, w_3 \, [_1 \, w_4 \, [_0 \, [_0 \, [_0$$

in the case when $M = 15$. Let us consider the set of symbols

$$S_{\mathscr{T}} = D \cup \Big\{ [_0 \, , [_i \, , \,]_i, 0 < i \leqslant 2\Big[\mathscr{T}(DS^*) - \mathscr{T}([_0 \, S^*)\Big]\Big\}.$$

Since $\mathscr{G}$ uses only those symbols, we see from the proposed coding of $\mathscr{G}$ that it can take at most $|S_{\mathscr{T}}|^M$ different values. Since

$$|S_{\mathscr{T}}| = |D| + 1 + 4\Big[\mathscr{T}(DS^*) - \mathscr{T}([_0 S^*)\Big],$$

we have proved that

$$|\mathfrak{G}_{\mathscr{T}}| \leqslant \Big(|D| + 1 + 4\Big[\mathscr{T}(DS^*) - \mathscr{T}([_0 S^*)\Big]\Big)^{5\mathscr{T}(DS^*) - 3\mathscr{T}([_0 \, S^*)}.$$

Let us notice that this bound, while being finite, is very large, and probably quite loose in practice. □

1.3.3 Splitting rules and label identification

In the previous section, we introduced some class of random processes, and studied some of their general properties. In this section, we are going to describe some more specific schemes and go further in the description of split and merge processes that can learn toric grammars in a satisfactory way.

The choice of splitting rules and label identification rules has a decisive influence on the way syntactic categories and syntactic rules are learnt by the split and merge process. While it is necessary as a starting point to consider rules learnt from the text to be parsed itself, it will also be fruitful to consider the case when a previously learnt reference grammar $\mathscr{R} \in \mathfrak{G}$ can be used to govern the splits.

To make things easier to grasp, let us explain on some example the basics of syntactic generalization by label identification. Let us start with the simple text with two sentences.

$$G_0 = \mathscr{T} = [_0 \text{ This is my friend Peter . } \oplus [_0 \text{ This is my neighbour John .}$$

If we split "my friend" and "my neighbour" in the two sentences using the same label, we will form after two splits the grammar

$$\begin{aligned} G_1 = [_0 \text{ This is }]_1 \text{ Peter . } \oplus [_0 \text{ This is }]_1 \text{ John .} \\ \oplus [_1 \text{ my friend } \oplus [_1 \text{ my neighbour} \end{aligned}$$

If no more splits are allowed and we therefore reached the stopping time of the splitting process, so that $\tau = 2$, we can proceed to the production process, and reach after two more steps the new text G_2 that can either be $G_2 = G_0$ or

$$G_2 = [_0 \text{ This is my neighbour Peter . } \oplus [_0 \text{ This is my friend John .}$$

Now is a good time to remind the reader of the distinction made in section 1.2.2 on page 23 about local and global expressions.

Legitimate local expressions will be provided by the reference grammar $\mathscr{R}$, whereas global expressions will be deduced from the text itself. This approach will be particularly efficient in the case when the set of local expressions is smaller than the set of global expressions.

We will need two different kinds of split processes, one to learn the reference grammar from a text and the other one to perform the first part of the transitions of the communication Markov chain.

These split processes may be viewed as performing some parsing of the text they are applied to. Here, we do not use parsing as it is usually used to discover whether a sentence is correct or not, we use it instead to discover new expressions.

We will start by defining the parsing rules to be used in the communication chain. We will call them *narrow* parsing rules. We will then proceed to the

definition of a *broad* parsing rule suitable for a first go at learning the reference grammar $\widehat{\mathscr{R}}(\mathscr{T})$ from a text.

Definition 1.3.7
Let us define the narrow parsing rule with reference grammar $\mathscr{R}$ as

$$\beta_n\Big(\mathscr{G},\mathscr{R}\Big) = \Big\{ \mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab, \\ ab \in \mathscr{E}_g,\ \mathscr{R}\Big([_i\, b\Big) > 0 \Big\}, \quad \mathscr{G} \in \mathfrak{G}.$$

Let us remark that, due to the definition of the set of expressions $\mathscr{E}$ and of toric grammars $\mathfrak{G} \subset \mathscr{M}_+(\mathscr{E})$, the fact that $\mathscr{G}$ and $\mathscr{G}' \in \mathfrak{G}$ implies that $i \in \mathbb{N} \setminus \{0\}$ in this definition, since necessarily $a\,]_i, [_i\, b \in \mathscr{E}$. It implies also that $[_0 \in \mathrm{supp}(a)$, a condition equivalent to $ab \in \mathscr{E}_g$.

The narrow parsing rule depends on $\mathscr{R}$ only through $\mathrm{supp}(\mathscr{R}) \cap \mathscr{E}_l$.

Let us define the broad parsing rule as

$$\beta_b\Big(\mathscr{G},\mathscr{R}\Big) = \Big\{ \mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab, \\ \mathscr{R}\Big(a\,]_i\Big) + \mathscr{R}\Big([_i\, b\Big) > 0, \mathscr{R}\Big(aS^*\Big) \leqslant \mu_1 \mathscr{R}\Big([_0\, S^*\Big), \\ \text{and } \mathscr{R}\Big(bS^*\Big) \leqslant \mu_2 \mathscr{R}\Big([_0\, S^*\Big) \Big\}, \qquad \mathscr{G},\mathscr{R} \in \mathfrak{G},$$

where $\mu_1, \mu_2 \in \mathbb{R}_+$ are two positive real parameters.

Since the reference grammar is under construction during broad parsing, we will mainly use this rule with $\mathscr{R} = \mathscr{G}$, as will be explained later. The same learning parameters μ_1 and μ_2 are present here and in the innovation rule to be described next. They serve to split expressions into sufficiently infrequent halves, in order to constrain the model.

Let us define now maximal sequences, a notion that will be needed to define learning rules.

Definition 1.3.8
Given some toric grammar $\mathscr{G}$, we will say that $a \in S^+$ is $\mathscr{G}$-maximal and write $a \in \max(\mathscr{G})$ when

$$\mathscr{G}(aS^*) > \max\Big\{\mathscr{G}(awS^*), \mathscr{G}(waS^*), w \in S\Big\}.$$

In other words, a is a maximal subsequence among the subsequences with the same weight in $\mathscr{G}$. Note that if a is $\mathscr{G}$-maximal, usually $\mathscr{G}(a) = 0$ (meaning that a is not an expression of the grammar, but only a subexpression) and if moreover the grammar $\mathscr{G}$ has integer weights (which will be the case if it has been produced by a split and merge process), then $\mathscr{G}(aS^*) \geqslant 2$.

Definition 1.3.9 (Innovation rule)
Using the notations $\lfloor_+ = \Big\{\lfloor_i, i \in \mathbb{N}\setminus\{0\}\Big\}$ *and* $\rfloor_+ = \Big\{\rfloor_i, i \in \mathbb{N}\setminus\{0\}\Big\}$, *let us define the innovation rule with reference grammar* $\mathscr{R}$ *as*

$$\beta_i\Big(\mathscr{G},\mathscr{R}\Big) = \Big\{ \mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus a\rfloor_i \oplus \lfloor_i b \ominus ab, \\ \mathscr{R}\Big(\lfloor_i S^*\Big) = 0,\ \{a,b\} \cap \max(\mathscr{R}) \neq \varnothing, \\ \mathscr{R}(aS^*) \leqslant \mu_1 \mathscr{R}\Big(\lfloor_0 S^*\Big),\ \textit{and}\ \mathscr{R}\Big(bS^*\Big) \leqslant \mu_2 \mathscr{R}\Big(\lfloor_0 S^*\Big)\ \Big\}.$$

Here again, the rule will be used while learning the reference grammar with $\mathscr{R} = \mathscr{G}$.

We will now introduce a label map that identifies the labels appearing in the same context.

Definition 1.3.10 (Label identification through context)
Given some toric grammar $\mathscr{G} \in \mathfrak{G}$, *let us consider the relation* $C \in (\mathbb{N}\setminus\{0\})^2$ *defined as*

$$C = \left\{(i,j) \in \Big(\mathbb{N}\setminus\{0\}\Big)^2 : \sum_{a\in S^*} \mathscr{G}(a\rfloor_i)\,\mathscr{G}(a\rfloor_j) + \mathscr{G}(\lfloor_i a)\,\mathscr{G}(\lfloor_j a) > 0\right\}.$$

The smallest equivalence relation containing C *defines a partition of* $\mathbb{N}\setminus\{0\}$ *into equivalence classes. Let* $(A_k)_{k\in\mathbb{N}\setminus\{0\}}$ *be an arbitrary indexing of this partition. Each positive integer falls in a unique class of the partition, so that the relation* $i \in A_{\underline{\chi}_{\mathscr{G}}(i)}$ *defines a label map* $\underline{\chi}_{\mathscr{G}} : \mathbb{N} \to \mathbb{N}$ *in a non ambiguous way. The choice of the indexing of the partition* $(A_k)_{k\in\mathbb{N}\setminus\{0\}}$ *does not matter, since two different choices lead to two isomorphic label maps. When applying* $\underline{\chi}_{\mathscr{G}}$ *to* $\mathscr{G}$ *itself, we will use the short notation* $\underline{\chi}(\mathscr{G}) \stackrel{\text{def}}{=} \underline{\chi}_{\mathscr{G}}(\mathscr{G})$.

Let us consider the evolution of the number of labels used by $\mathscr{G}$*:*

$$L(\mathscr{G}) = \Big|\{i \in \mathbb{N} : \mathscr{G}(\rfloor_i S^*) > 0\}\Big|.$$

It is easy to see that $L\Big(\underline{\chi}(\mathscr{G})\Big) \leqslant L(\mathscr{G})$ *and that* $\underline{\chi}(\mathscr{G}) \equiv \mathscr{G}$ *(where the symbol* $\equiv$ *means isomorphic) if and only if* $L\Big(\underline{\chi}_{\mathscr{G}}(\mathscr{G})\Big) = L(\mathscr{G})$. *Accordingly there is* $k \in \mathbb{N}$ *such that* $\underline{\chi}^{k+1}(\mathscr{G}) \equiv \underline{\chi}^k(\mathscr{G})$, *and we can take it to be the smallest integer such that* $L\Big(\underline{\chi}^{k+1}(\mathscr{G})\Big) = L\Big(\underline{\chi}^k(\mathscr{G})\Big)$. *Consequently,* k *is such that for any* $n \geqslant k$, $\underline{\chi}^n(\mathscr{G}) \equiv \underline{\chi}^k(\mathscr{G})$. *We will define* $\chi(\mathscr{G}) = \underline{\chi}^k(\mathscr{G})$, *up to grammar isomorphisms (so that* $\chi(\mathscr{G})$ *belongs to* $\mathfrak{G}/\equiv$ *rather than to* $\mathfrak{G}$ *itself).*

A characterisation in terms of more elementary label maps will be established in proposition 1.A.6 on page 58. This characterization provides an algorithm to compute χ in practice.

We are now ready to define a learning rule.

Definition 1.3.11
Let us define the learning rule

$$\beta_\ell(\mathscr{G}) = \begin{cases} \beta_i\big(\mathscr{G}, \mathscr{G}\big), & \text{when } \beta_b\big(\mathscr{G}, \mathscr{G}\big) = \varnothing, \\ \Big\{\chi\big(\mathscr{G}'\big) : \mathscr{G}' \in \beta_b\big(\mathscr{G}, \mathscr{G}\big)\Big\}, & \text{otherwise.} \end{cases}$$

We will define two kinds of splitting processes, based on two different choices of the restricted splitting rule β_r.

Definition 1.3.12 (Learning process)
A learning process *is a splitting process with restricted splitting rule*

$$\beta_r(\mathscr{G}) = \beta_\ell(\mathscr{G}).$$

Definition 1.3.13 (Parsing process)
A parsing process *with reference grammar $\mathscr{R} \in \mathfrak{G}$ is a splitting process with restricted splitting rule*

$$\beta_r(\mathscr{G}) = \beta_n(\mathscr{G}, \mathscr{R}).$$

Before we reach the aim of this chapter and describe our statistical language model, we need to explore some of the properties of the production, learning and parsing processes introduced so far.

1.4 Parsing and generalization

Let us introduce some notations for the output of parsing, learning and production processes.

Definition 1.4.1
Let S_t be a parsing process, with reference grammar $\mathscr{R} \in \mathfrak{G}$. We will use the following notation for the distribution of S_τ.

$$\mathbb{G}_{\mathscr{T},\mathscr{R}} = \mathbb{P}_{S_\tau \,|\, S_0 = \mathscr{T}}, \qquad \mathscr{T} \in \mathfrak{T}.$$

We will also use a short notation for the distribution of the output of a production process.

$$\mathbb{T}_{\mathscr{G}} = \mathbb{P}_{P_\sigma \,|\, P_0 = \mathscr{G}, P_\sigma \in \mathfrak{T}}, \qquad \mathscr{G} \in \mathfrak{G}.$$

Eventually, $\mathbb{G}_{\mathscr{T}}$ will be the probability distribution of the output of a learning process S_t, according to the definition

$$\mathbb{G}_{\mathscr{T}} = \mathbb{P}_{S_\tau \,|\, S_0 = \mathscr{T}}, \qquad \mathscr{T} \in \mathfrak{T}.$$

At this point we obviously may consider different notions of parsing that we have to connect together. Namely, we would like to make a link between the following statements:

- $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$, the grammar $\mathscr{G}$ can produce the text $\mathscr{T}$;
- $\mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0$, the text $\mathscr{T}$ can generate the grammar $\mathscr{G}$ when parsed with the help of the grammar $\mathscr{R}$;
- $\mathbb{G}_{\mathscr{T}}(\mathscr{G}) > 0$, the grammar $\mathscr{G}$ can be learnt from the text $\mathscr{T}$.

Lemma 1.4.1
The previous parse notions are related in the following way. For any $\mathscr{G}, \mathscr{R} \in \mathfrak{G}$, and any $\mathscr{T} \in \mathfrak{T}$,

$$\begin{aligned} \mathbb{G}_{\mathscr{T}}(\mathscr{G}) > 0 &\implies \mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0, \\ \mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0 &\implies \mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0, \\ \mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0 &\implies \mathbb{G}_{\mathscr{T},\mathscr{G}}(\mathscr{G}) > 0. \end{aligned}$$

Consequently, for any $\mathscr{G}, \mathscr{R} \in \mathfrak{G}$ such that $\Big(\mathrm{supp}(\mathscr{G}) \cap \mathscr{E}_l\Big) \subset \mathrm{supp}(\mathscr{R})$, and any $\mathscr{T} \in \mathfrak{T}$,

$$\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0 \iff \mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0.$$

PROOF. This is one of the core lemmas of this chapter. The proof is given in section 1.A.2 on page 51, on account of its length. □

It has the following important implication.

Proposition 1.4.2
Given a parsing process S_t based on a reference grammar $\mathscr{R} \in \mathfrak{G}$ and a production process P_t, the corresponding split and merge process G_t is weakly reversible, in the sense that for any $\mathscr{T} \in \mathfrak{T}$, any $\mathscr{G} \in \bigcup_{t\in\mathbb{N}} \mathrm{supp}\Big(\mathbb{P}_{G_{2t+1}}\Big)$,

$$\mathbb{P}\Big(G_1 = \mathscr{G} \,\Big|\, G_0 = \mathscr{T}\Big) > 0 \iff \mathbb{P}\Big(G_2 = \mathscr{T} \,\Big|\, G_1 = \mathscr{G}\Big) > 0.$$

Consequently, for any $\mathscr{T}, \mathscr{T}' \in \mathfrak{T}$ and any $\mathscr{G}, \mathscr{G}' \in \bigcup_{t\in\mathbb{N}} \mathrm{supp}\Big(\mathbb{P}_{G_{2t+1}}\Big)$,

$$\begin{aligned} \mathbb{P}\Big(G_2 = \mathscr{T}' \,\Big|\, G_0 = \mathscr{T}\Big) > 0 &\iff \mathbb{P}\Big(G_2 = \mathscr{T} \,\Big|\, G_0 = \mathscr{T}'\Big) > 0, \\ \mathbb{P}\Big(G_3 = \mathscr{G}' \,\Big|\, G_1 = \mathscr{G}\Big) > 0 &\iff \mathbb{P}\Big(G_3 = \mathscr{G} \,\Big|\, G_1 = \mathscr{G}'\Big) > 0. \end{aligned}$$

In other words, the two processes G_{2t} and G_{2t+1} are weakly reversible time homogeneous Markov chains. As we already proved that the set of reachable states from any starting point is finite, it shows that they are recurrent Markov chains: they partition their respective state spaces into positive recurrent communicating classes.

Proof. Let us remark first that

$$\mathbb{P}\Big(G_1 = \mathscr{G} \,\Big|\, G_0 = \mathscr{T}\Big) > 0 \iff \mathbb{G}_{\mathscr{T},\mathscr{R}}\Big(\mathscr{G}\Big) > 0$$
$$\mathbb{P}\Big(G_2 = \mathscr{T} \,\Big|\, G_1 = \mathscr{G}\Big) > 0 \iff \mathbb{T}_{\mathscr{G}}\Big(\mathscr{T}\Big) > 0.$$

Moreover, since $\mathscr{G} \in \mathrm{supp}(\mathbb{P}_{G_{2t+1}})$ for some $t \in \mathbb{N}$, there is $\mathscr{T}' \in \mathfrak{T}$ such that $\mathbb{G}_{\mathscr{T}',\mathscr{R}}(\mathscr{G}) > 0$, implying that $\mathrm{supp}(\mathscr{G}) \cap \mathscr{E}_l \subset \mathrm{supp}(\mathscr{R})$. This ends the proof according to the last statement of the previous lemma. □

We will see in section 3.5 on page 151 a method to build split and merge processes that are actually reversible, and not only weakly so. This method involves a slightly modified Metropolis algorithm, that allows for intermediate steps (here, the grammar G_{2t+1}).

1.5 Expectation of a random toric grammar

In section 1.4 on page 35, given some text $\mathscr{T} \in \mathfrak{T}$, we defined a random distribution on toric grammars $\mathbb{G}_{\mathscr{T}}$ that we would like to use to learn a grammar from a text. The most obvious way to do this is to draw a toric grammar at random according to the distribution $\mathbb{G}_{\mathscr{T}}$, and we already saw an algorithm, described by a Markov chain and a stopping time, to do this.

The distribution $\mathbb{G}_{\mathscr{T}}$ will be spread in general on many grammars. This is a kind of instability that we would like to avoid, if possible. A natural way to get rid of this instability would be to simulate the expectation of $\mathbb{G}_{\mathscr{T}}$. To do this, we are facing a problem: the usual definition of the expectation of $\mathbb{G}_{\mathscr{T}}$, that is

$$\int \mathscr{G}\, \mathrm{d}\mathbb{G}_{\mathscr{T}}(\mathscr{G}),$$

although well defined from a mathemacial point of view, is a meaningless toric grammar, due to the possible fluctuations of the label mapping. To get a meaningful notion of expectation, we need to define in a meaningful way the sum of two toric grammars. We will achieve this in two steps.

Let us introduce first the *disjoint sum* of two toric grammars. We will do this with the help of two disjoint label maps. Let us define the *even* and *odd* label maps f_e and f_o as

$$f_e(i) = 2i, \qquad f_o(i) = \max\{0, 2i-1\}, \qquad i \in \mathbb{N}.$$

Definition 1.5.1
The disjoint sum *of two toric grammars $\mathscr{G}, \mathscr{G}' \in \mathfrak{G}$ is defined as*

$$\mathscr{G} \boxplus \mathscr{G}' = f_e(\mathscr{G}) + f_o(\mathscr{G}').$$

Definition 1.5.2
Given a probability measure $\mathbb{G} \in \mathscr{M}_+^1(\mathfrak{G})$ with finite support, we define the mean of $\mathbb{G}$ as

$$\oint \mathscr{G} \, \mathrm{d}\mathbb{G}(\mathscr{G}) = \chi\left(\bigboxplus_{\mathscr{G} \in \mathfrak{G}} \mathbb{G}(\mathscr{G}) \, \mathscr{G} \right).$$

Lemma 1.5.1
If G_i is an i.i.d. sequence of random grammars distributed according to $\mathbb{G}$, then almost surely

$$\lim_{n \to +\infty} \frac{1}{n} \chi\left(\bigboxplus_{i=1}^{n} G_i \right) = \oint \mathscr{G} \, \mathrm{d}\mathbb{G}(\mathscr{G}).$$

PROOF. The proof of this result is quite lengthy, and postponed till section 1.A.3 on page 54. □

1.6 Language models

1.6.1 Communication model

We are now ready to define the language model announced in the introduction. Given a reference grammar $\mathscr{R}$, and the corresponding split and merge process $(G_t)_{t \in \mathbb{N}}$ with reference $\mathscr{R}$, we define the communication kernel $q_{\mathscr{R}}(\mathscr{T}, \mathscr{T}')$ on $\mathfrak{T}^2$ as

$$q_{\mathscr{R}}(\mathscr{T}, \mathscr{T}') = \mathbb{P}\Big(G_2 = \mathscr{T}' \,|\, G_0 = \mathscr{T}\Big).$$

According to proposition 1.3.5 on page 30 and proposition 1.4.2 on page 36, $q_{\mathscr{R}}$ has finite reachable sets and is weakly reversible, so that all texts $\mathscr{T} \in \mathfrak{T}$ are positive recurrent states of the communication kernel $q_{\mathscr{R}}$.

Thus to each text $\mathscr{T} \in \mathfrak{T}$ corresponds a unique invariant text distribution $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$, as explained in the introduction. As all states are positive recurrent, $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$ is the unique invariant measure of $q_{\mathscr{R}}$ on the communicating class containing $\mathscr{T}$. Moreover, from the ergodic theorem,

$$\mathbb{P}\left(\widehat{q}_{\mathscr{R}}\Big(\mathscr{T}, \cdot\Big) = \lim_{t \to \infty} \frac{1}{t} \sum_{j=1}^{t} \delta_{G_{2j}} \;\middle|\; G_0 = \mathscr{T} \right) = 1,$$

showing that $\widehat{q}_{\mathscr{R}}(\mathscr{T},\cdot)$ can be computed by an almost surely convergent Monte-Carlo simulation. Eventually, we deduce from the invariant probability measure on texts $\widehat{q}_{\mathscr{R}}(\mathscr{T},\cdot)$, a probability measure on sentences $\widehat{Q}_{\mathscr{R},\mathscr{T}}$ as explained in the introduction, according to the formula

$$\widehat{Q}_{\mathscr{R},\mathscr{T}} = \mathscr{T}\big(\lfloor_0 S^*\big)^{-1} \sum_{\mathscr{T}'\in\mathfrak{T}} \widehat{q}_{\mathscr{R}}\big(\mathscr{T},\mathscr{T}'\big)\mathscr{T}'.$$

(This is the same formula as eq. (1.1.1) on page 18 in the introduction, taking into account the fact that texts in the support of $\widehat{q}_{\mathscr{R}}(\mathscr{T},\cdot)$ are non normalized empirical measures with the same total mass equal to $\mathscr{T}(\lfloor_0 S^*)$, the number of sentences in the text $\mathscr{T}$.)

To obtain a true language estimator, there remains to estimate $\mathscr{R}$ by some estimator $\widehat{\mathscr{R}}(\mathscr{T})$. We could do this as described in section 1.5 on page 37, putting

$$\widehat{\mathscr{R}}(\mathscr{T}) = \oint \mathscr{G}\,\mathrm{d}\mathbb{G}_{\mathscr{T}}(\mathscr{G}).$$

Let us remark that, according to lemma 1.5.1 on the preceding page, $\widehat{\mathscr{R}}(\mathscr{T})$ can be computed from repeated simulations from the distribution $\mathbb{G}_{\mathscr{T}}$.

1.6.2 Comparison with other models

Comparison with context-free grammars

Given a toric grammar $\mathscr{G}\in\beta^*(\mathfrak{T})$, we may consider the split and merge process G_t with reference grammar $\mathscr{G}$ starting at $G_1=\mathscr{G}$ (so here we start at time 1 with an initial state that is a grammar, instead of starting at time 0 with an initial state that is a text). Due to the weak reversibility of proposition 1.4.2 on page 36, G_2 almost surely falls in the same recurrent communicating class of $t\mapsto G_{2t}$, and the unique invariant probability measure supported by this recurrent communicating class defines a probability measure $\overline{\mathbb{T}}_{\mathscr{G}}$ on texts, and therefore a stochastic language model. This way of defining the language generated by the grammar $\mathscr{G}$ can be compared to the usual definition of the language generated by a context-free grammar. Indeed, the support of $\mathscr{G}$ is a context-free grammar, so this is meaningful to consider the language generated by this grammar and to compare it with the support of our stochastic language model.

None of these two sets of sentences is contained in the other one. In our stochastic model, the number of times a rule can be used is bounded, so if the recursive use of some rules is possible, the deterministic language will in this sense be larger. On the other hand, the stochastic model uses both production and parsing to build new sentences, whereas the deterministic model uses only

production rules. In this respect, the stochastic model may, at least in some cases, define a much broader language, as we will show on the following example.

Let us take as dictionary the set

$$D = \{+, =\} \cup [\![1, N]\!],$$

where $[\![1, N]\!] = \Big\{ i \in \mathbb{N}, 1 \leqslant i \leqslant N \Big\}$, and consider the toric grammar

$$\mathscr{G} = N^2 \otimes \text{"}[_0\ ]_N = N\text{"} \ \oplus \bigoplus_{i=1}^{N} N \otimes \ \text{"}[_i\ i\text{"} \ \oplus \bigoplus_{i=2}^{N} N(i-1) \otimes \ \text{"}[_i\ ]_{i-1} + 1\text{"},$$

(where we put expressions into double quotes " to ease the reading of this definition.) Let us also consider the text

$$\mathscr{T} = N \otimes \bigoplus_{i=1}^{N} \text{"}[_0\, i \underbrace{+1 + \cdots + 1}_{N-i \text{ times}} = N\text{"}.$$

It is easy to check that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$, (so that $\mathscr{G} \in \beta^*(\mathscr{T})$,) that indeed the support of $\mathscr{T}$ is the language generated by $\mathrm{supp}(\mathscr{G})$, seen as a context-free grammar, and that the stochastic language $\overline{\mathbb{T}}_{\mathscr{G}}$ generated by $\mathscr{G}$ is able to produce with positive probability a set of sentences

$$\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}}) \stackrel{\text{def}}{=} \bigcup_{\mathscr{T} \in \mathrm{supp}(\overline{\mathbb{T}}_{\mathscr{G}})} \mathrm{supp}(\mathscr{T}),$$

equal to

$$\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}}) = \Big\{ \text{"}[_0\ x_1 + \cdots + x_i = x_{i+1} + \cdots + x_j\text{"},$$
$$1 \leqslant i < j \leqslant 2N,\ x_k \in [\![1, N]\!],\ 1 \leqslant k \leqslant j,\ \sum_{k=1}^{i} x_k = \sum_{k=i+1}^{j} x_k = N \Big\}.$$

Here, the number of sentences produced by the underlying context-free grammar is $|\mathrm{supp}(\mathscr{T})| = N$, whereas the number of sentences produced by our stochastic language model is $|\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}})| = 2^{2(N-1)}$. Thus, in this small example based on arithmetic expressions (admittedly closer to a computing language than it is to a natural language), our new definition of the generated language induces a huge increase in the number of generated sentences.

Note that with usual context-free grammar notations, $\mathrm{supp}(\mathscr{G})$ would have been described as

$$\begin{aligned}
&\boxed{0} \to \boxed{N} = N && \\
&\boxed{i} \to i, && i = 1, \ldots, N, \\
&\boxed{i} \to \boxed{i-1} + 1, && i = 2, \ldots, N,
\end{aligned}$$

where $\boxed{0}$ is the start symbol and $\boxed{i}$, $i = 1, \ldots, N$, are other non terminal symbols.

To count the number of elements in $\text{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}})$, one can remark that the number of ways N can be written as $\sum_{k=1}^{i} x_k$ with an arbitrary number of terms is also the number of increasing integer sequences $0 < s_1 < \cdots < s_{i-1} < N$ of arbitrary length, which is also the number of subsets $\{s_1, \ldots, s_{i-1}\}$ of $\{1, \ldots, N-1\}$, that is 2^{N-1}.

Intuitively speaking, the underlying context-free grammar $\text{supp}(\mathscr{G})$ is limited to producing a small set of global expressions of the form $i + 1 + \ldots + 1 = N$, whereas the stochastic language model incorporates some crude logical reasoning that is capable of deducing from them a large set of new global expressions.

Let us remark also that, when we start as here from a text made of true arithmetic statements, the language generated by our language model is also made of true arithmetic statements. This shows that our approach to language modeling is capable of some sort of logical reasoning.

Comparison with Markov models

The kind of reasoning illustrated in the previous section is related to the fact that we analyse global syntactic structures represented by the global expressions of our toric grammars.

In order to give another point of comparison, we would like in this section to make a qualitative comparison with Markov models, that do not share this feature. To make a parallel between toric grammars and Markov models, we are going to show how a Markov model could be described in terms of toric grammars and label identification rules. More striking relations between our models and Markov models will however be outlined in forthcoming chapters.

To build a Markov model in our framework, we have to use a deterministic splitting (or parsing) rule. This is because in a Markov model, conditional probabilities are specified from left to right in a rigid data independent way. Let us introduce the Markov splitting rule

$$\beta_m(\mathscr{G}) = \Big\{\mathscr{G}' \in \mathfrak{G},\ \mathscr{G}' = \mathscr{G} \ominus [_0\ aw\ ]_i \oplus [_0\ a\ ]_j \oplus [_j\ w\ ]_i, \\ i, j \in \mathbb{N} \setminus \{0\}, a \in D^+, w \in D, \mathscr{G}\Big([_j\ S^*\Big) = 0\Big\}.$$

We will describe now label identification rules using concepts introduced in section 1.A.3 on page 54. Let us say that the pair of labels $p \in \Big(\mathbb{N}\setminus\{0\}\Big)^2$ is $\mathscr{G}$-Markov if there is $w \in D$ such that $\mathscr{G}\Big(w]_{p_1} S^*\Big)\mathscr{G}\Big(w]_{p_2} S^*\Big) > 0$. Let us say that the sequence of pairs $p^1, \ldots, p^k$ is $\mathscr{G}$-Markov if p^j is $\xi_{p^1,\ldots,p^{j-1}}(\mathscr{G})$-Markov. It can be proved as in the case of congruent sequences that if σ is a permutation and p is $\mathscr{G}$-Markov, then $p \circ \sigma$ is also $\mathscr{G}$-Markov. It can also be proved that if p and q are maximal $\mathscr{G}$-Markov

sequences, then $\xi_p \equiv \xi_q$, and therefore $\xi_p(\mathscr{G}) \equiv \xi_q(\mathscr{G})$. We will call $\xi_p(\mathscr{G}) \in \mathfrak{G}/\equiv$ the Markov closure of $\mathscr{G}$ and use the notation $\xi_p(\mathscr{G}) \stackrel{\text{def}}{\equiv} \mu(\mathscr{G})$,where $\mu(\mathscr{G})$ is the Markov pendent of $\chi(\mathscr{G})$ in the construction of toric grammars.

Let S_t, $0 \leqslant t \leqslant \tau$ be a splitting process based on the restricted splitting rule

$$\beta_r(\mathscr{G}) = \Big\{\mu(\mathscr{G}'),\ \mathscr{G}' \in \beta_m(\mathscr{G})\Big\}.$$

It is not very difficult to check that the support of S_τ is contained in a single isomorphic class of grammars, so that, up to label remapping the result of this splitting process is deterministic. More specifically, starting from a text

$$\mathscr{T} = \bigoplus_{j=1}^{n} \lfloor_0\, w_1^j \dots w_{\ell(j)}^j,$$

where $w_i^j \in D \setminus \{.\}$, $1 \leqslant i < \ell(j)$, $1 \leqslant j \leqslant n$, and $w_{\ell(j)}^j = .$, $1 \leqslant j \leqslant n$ so that all sentences end with a period, we obtain a grammar isomorphic to

$$\mathscr{G} = \bigoplus_{j=1}^{n}\Big(\lfloor_0\ w_1^j\ \rfloor_{w_1^j} \bigoplus_{i=2}^{\ell(j)-1} \lfloor_{w_{i-1}^j} w_i^j\ \rfloor_{w_i^j} \oplus \lfloor_{w_{\ell(j)-1}^j} w_{\ell(j)}^j\Big),$$

where we have used words as labels instead of integers, since in this model, due to the label identification rule, labels are functions of words (namely $\rfloor_w$ is the non terminal symbol following the word $w \in D$).

We can now define a Markov production mechanism, to replace the production process. It is described as a Markov chain X_i, $i \in \mathbb{N}$, where $X_i \in D \cup \{\Delta\}$, where $\Delta \notin D$ is a padding symbol used to embed finite sentences into infinite sequences of symbols, all equal to Δ for indices larger than the sentence length. The distribution of the Markov chain X_i is as follows. Its initial distribution is

$$\mathbb{P}(X_0 = w) = \frac{\mathscr{G}\big(\lfloor_0 w\rfloor_w\big)}{\mathscr{G}\big(\lfloor_0 S^*\big)},$$

and its transition probabilities are

$$\begin{aligned}
\mathbb{P}\big(X_i = \Delta \,|\, X_{i-1} = .\big) &= 1, \\
\mathbb{P}\big(X_i = . \,|\, X_{i-1} = w\big) &= \frac{\mathscr{G}\big(\lfloor_w .\big)}{\mathscr{G}\big(\lfloor_w S^*\big)}, && w \in D \setminus \{.\} \\
\mathbb{P}\big(X_i = w' \,|\, X_{i-1} = w\big) &= \frac{\mathscr{G}\big(\lfloor_w w'\rfloor_{w'}\big)}{\mathscr{G}\big(\lfloor_w S^*\big)}, && w, w' \in D \setminus \{.\}.
\end{aligned}$$

Roughly speaking, the difference with the production process P_t defined previously is that in the production process the production rules are drawn at random without replacement whereas here, the production rules are drawn with replacement.

It is easy to see that the initial distribution and transition probabilities of the Markov chain X_i are the empirical initial distribution and empirical transition probabilities of the training text $\mathscr{T}$.

In conclusion, to build a Markov model using the same framework as for toric grammars, we had to modify two steps in a dramatic way:

- we had to change the splitting process, and replace the random splitting process of toric grammars with a non random splitting process which chains forward transitions in a linear way;
- we had to change in a dramatic way the label identification rule to replace the *forward and backward global condition* of toric grammars with a *backward only local condition.*

(The modification of the production process is less crucial and boils down to drawing production rules with or without replacement.)

We hope that this discussion of Markov models will help the reader realize that our model proposal is indeed really different from the Markov model at sentence level. We could have extended easily the discussion to Markov models of higher order, or to more general context tree models. We let the reader figure out the details. All these more sophisticated models show the same differences from toric grammars: a more rigid splitting process and local backward label identification rules.

1.7 A small experiment

Let us end this chapter with a small example. Here we use a small text that is meant to mimic what could be found in a tutorial to learn English as a foreign language. We have added a more elaborate sentence at the end of the text to show its impact. More systematic experiments are yet to be carried out, although the conception of this model was guided by experimental trial and errors with models starting with variable length Markov chains, before we tried global rules leading to grammars.

This is the training text $\mathscr{T}$ (each line shows an expression, starting with its weight) :

```
1 [0 He is a clever guy .
1 [0 He is doing some shopping .
1 [0 He is laughing .
```

```
1 [0 He is not interested in sports .
1 [0 He is walking .
1 [0 He likes to walk in the streets .
1 [0 I am driving a car .
1 [0 I am riding a horse too .
1 [0 I am running .
1 [0 Paul is crossing the street .
1 [0 Paul is driving a car .
1 [0 Paul is riding a horse .
1 [0 Paul is walking .
1 [0 Peter is walking .
1 [0 While I was walking , I saw Paul crossing the street .
```

And now, the new sentences produced by the model (that is by $\widehat{Q}_{\widehat{\mathscr{R}},\mathscr{T}}$, approximated on 50 iterations of the communication chain with kernel $q_{\widehat{\mathscr{R}}}$).

```
1 [0 Paul is driving a car too .
1 [0 Paul is doing some shopping .
1 [0 Paul is laughing .
1 [0 Paul is riding a horse too .
1 [0 Paul is running too .
1 [0 Paul is running .
1 [0 Paul is not interested in sports too .
1 [0 Paul is not interested in sports .
1 [0 Paul is a clever guy too .
1 [0 Paul is a clever guy .
1 [0 Paul is walking too .
1 [0 Peter is driving a car too .
1 [0 Peter is driving a car .
1 [0 Peter is doing some shopping .
1 [0 Peter is laughing .
1 [0 Peter is riding a horse too .
1 [0 Peter is riding a horse .
1 [0 Peter is running too .
1 [0 Peter is running .
1 [0 Peter is not interested in sports .
1 [0 Peter is a clever guy .
1 [0 Peter is crossing the street .
1 [0 He is driving a car too .
1 [0 He is driving a car .
1 [0 He is riding a horse too .
1 [0 He is riding a horse .
1 [0 He is running too .
1 [0 He is running .
```

```
1 [0 He is not interested in sports too .
1 [0 He is crossing the street too .
1 [0 He is crossing the street .
1 [0 He is walking too .
1 [0 I am driving a car too .
1 [0 I am doing some shopping .
1 [0 I am laughing too .
1 [0 I am laughing .
1 [0 I am riding a horse .
1 [0 I am not interested in sports .
1 [0 I am a clever guy .
1 [0 I am crossing the street too .
1 [0 I am crossing the street .
1 [0 I am walking too .
1 [0 I am walking .
1 [0 While I was driving a car , I saw Paul doing some shopping too .
1 [0 While I was driving a car , I saw Paul doing some shopping .
1 [0 While I was driving a car , I saw Paul riding a horse .
1 [0 While I was driving a car , I saw Paul crossing the street .
1 [0 While I was driving a car , I saw Paul walking .
1 [0 While I was driving a car , I saw Peter riding a horse .
1 [0 While I was doing some shopping , I saw Paul riding a horse .
1 [0 While I was doing some shopping , I saw Paul walking .
1 [0 While I was laughing too , I saw Peter crossing the street .
1 [0 While I was laughing , I saw Peter riding a horse .
1 [0 While I was riding a horse , I saw Paul driving a car too .
1 [0 While I was riding a horse , I saw Paul driving a car .
1 [0 While I was riding a horse , I saw Paul laughing .
1 [0 While I was riding a horse , I saw Paul running .
1 [0 While I was riding a horse , I saw Paul walking .
1 [0 While I was riding a horse , I saw Peter not interested in sports .
1 [0 While I was running , I saw Paul laughing .
1 [0 While I was running , I saw Paul not interested in sports .
1 [0 While I was running , I saw Paul a clever guy .
1 [0 While I was running , I saw Paul walking .
1 [0 While I was not interested in sports , I saw Paul driving a car .
1 [0 While I was not interested in sports , I saw Paul riding a horse .
1 [0 While I was a clever guy , I saw Paul running .
1 [0 While I was a clever guy , I saw Paul crossing the street .
1 [0 While I was a clever guy , I saw Paul walking .
1 [0 While I was crossing the street , I saw Paul riding a horse .
1 [0 While I was crossing the street , I saw Paul running .
1 [0 While I was crossing the street , I saw Paul crossing the street .
```

```
1 [0 While I was crossing the street , I saw Paul walking .
1 [0 While I was crossing the street , I saw Peter walking .
1 [0 While I was walking , I saw Paul driving a car .
1 [0 While I was walking , I saw Paul laughing .
1 [0 While I was walking , I saw Paul riding a horse .
1 [0 While I was walking , I saw Paul running .
1 [0 While I was walking , I saw Paul not interested in sports .
1 [0 While I was walking , I saw Paul crossing the street too .
1 [0 While I was walking , I saw Paul walking .
1 [0 While I was walking , I saw Peter not interested in sports .
1 [0 While I was walking , I saw Peter walking .
```

The reference grammar was learnt first, and was computed from 10 samples of $\mathbb{G}_{\mathscr{T}}$. (We did not normalize the weights, since we were interested in the support of the local expressions only.)

```
10 [0 He likes to walk ]6 ]3 streets .
2 [0 ]1 ]8 clever guy .
2 [0 ]1 doing some shopping .
2 [0 ]1 laughing .
2 [0 ]1 not interested ]6 sports .
2 [0 ]1 riding ]8 horse .
2 [0 ]1 riding ]8 horse ]2 .
2 [0 ]1 running .
24 [0 ]7 am ]5 .
28 [0 Paul is ]5 .
40 [0 He is ]5 .
4 [0 ]1 crossing ]3 street .
4 [0 ]1 driving ]8 car .
5 [0 ]4 is ]5 .
6 [0 ]1 walking .
7 [0 Peter is ]5 .
8 [0 While ]7 was ]5 , ]7 saw ]4 ]5 .
10 [1 He is
2 [1 Peter is
2 [1 While ]7 was ]5 , ]7 saw ]4
6 [1 ]7 am
8 [1 Paul is
2 [2 too
30 [3 the
14 [4 Paul
1 [4 Peter
16 [5 crossing ]3 street
16 [5 driving ]8 car
```

```
16 [5 riding ]8 horse
34 [5 walking
8 [5 ]5 too
8 [5 ]8 clever guy
8 [5 doing some shopping
8 [5 laughing
8 [5 not interested ]6 sports
8 [5 running
20 [6 in
50 [7 I
50 [8 a
```

Although we did not yet make the software development effort required to test large text copora, we learnt a few interesting things from what we already tried:

- As it is, the model requires the inclusion of a sufficient number of simple and redundant sentences to start generalizing. At this stage, we do not know whether this could be avoided by changing the learning rules. We made quite a few attempts in this direction. All of them resulted in the production of grammatical nonsense. Breaking the global constraints that are enforced by the model seems to have a dramatic effect on grammatical coherence. This could be a clue that these global conservation rules reflect some fundamental feature of the syntactic structure of natural languages. Including a bunch of "simple" sentences made of frequent words may be seen as introducing a pinch of supervision in the learning process.

- The constraints on subexpressions frequencies in the learning rule definition 1.3.7 on page 33 and 1.3.9 were added to avoid some unwanted generalizations. For instance here we took $\mu_1\mathscr{R}\Big(\big[_0\, S^*\Big) = \mu_2\mathscr{R}\Big(\big[_0\, S^*\Big) = 5$. If we had chosen 10 instead of 5, sentences of the kind

  ```
  [0 While I was walking , I saw He crossing the street .
  ```

  would have emerged, where the pronoun "He" is substituted to a noun in the wrong place. We deliberately wrote the training text in such a way that "He" is more frequent than any noun, since we expect that to be true for any reasonable large corpus. Doing so, we were able to rule out the wrong construction by lowering the frequency constraint to avoid the unwanted substitution.

- Despite all the limitations of this small example, it shows that the model is able to find out non trivial new constructs, like

  ```
  [0 While I was laughing too, I saw Peter crossing the street.
  ```

> where it has discovered that "too" could be added to the subordinate clause opening the sentence. We are quite pleased to see that such things could be learnt along very general label identification rules, while all the generalized sentences remain, if not all grammatically correct, at least all grammatically plausible. Of course this judgement is purely subjective. But since we have no mathematical or otherwise quantitative definition of what natural languages are, we have to be content with a subjective evaluation of models.

Studying how this learning model scales with large corpora is still a work to be done (it will require from us that we optimize our code so that it can run efficiently on large data sets).

1.8 The story so far...

We have built in this chapter a new statistical framework for the syntactic analysis of natural languages.

The main idea pervading our approach is that trying to estimate the distribution of an isolated random sentence is hopeless. Instead we propose to build a Markov chain on sets of sentences (called texts in this work), with non trivial recurrent communicating classes and to define our language model as the invariant measures of this Markov chain on each of these recurrent communicating classes. At each step, the Markov chain recombines the set of sentences constituting its current state, using cut and paste operations described by grammar rules. In this way we define the probability distribution of an isolated random sentence only in an indirect way. We replace the hard question of generating a random sentence by the hopefully simpler one of recombining a set of sentences in a way that keep the desired distribution invariant.

The main result of this chapter is the construction of the split and merge process with reference grammar $\mathscr{R}$. It has non trivial mathematical properties proving that it can be simulated using a bounded number of operations at each step, and that the state space is divided into recurrent communicating classes each including a finite number of states.

Nonetheless, the model needs some further refinements. For example, we did not (yet) propose a simple way to define the actual probabilities on the split and merge process. But more importantly, our method to estimate the reference grammar through the grammar expectation

$$\widehat{\mathscr{R}}(\mathscr{T}) = \oint \mathscr{G} \, \mathrm{d}\mathbb{G}_{\mathscr{T}}\big(\mathscr{G}\big)$$

is clearly lacking. The next chapter will be dedicated to introducing the notion of Markov substitute sets, which will be used to estimate the reference grammar

within a more traditional statistical framework, enjoying proved mathematical properties.

1.A Proofs

1.A.1 Bound on the length of splitting and production processes

Proof of proposition 1.3.4 on page 30. Let us define the length of an expression $e \in S^k \cap \mathscr{E}$ as $\ell(e) = k$. Let us introduce some remarkable weights associated with a grammar $\mathscr{G} \in \beta^*(\mathfrak{T})$.

$$\begin{aligned} W_s(\mathscr{G}) &= \sum_{e \in \mathscr{E}} \mathscr{G}(e), \\ W_e(\mathscr{G}) &= \sum_{e \in \mathscr{E}} \mathscr{G}(e)\ell(e)^{-1}, \\ W_l(\mathscr{G}) &= \sum_{i=1}^{+\infty} \mathscr{G}([_i\, S^*), \\ W_w(\mathscr{G}) &= \sum_{w \in D} \mathscr{G}(wS^*). \end{aligned}$$

Let us define the set of canonical expressions as

$$\mathscr{E}_c = \mathscr{E} \cap \left(\bigcup_{i \in \mathbb{N}} [_i S^* \right).$$

Using previously introduced notations, we can write the grammar as

$$\mathscr{G} = \sum_{e \in \mathscr{E}_c} \mathscr{G}(e) \otimes e.$$

We will call this the canonical decomposition of $\mathscr{G}$. The two weights $W_s(\mathscr{G})$ and $W_e(\mathscr{G})$ are better understood in terms of this canonical decomposition. They can be expressed as

$$\begin{aligned} W_s(\mathscr{G}) &= \sum_{e \in \mathscr{E}_c} \mathscr{G}(e)\ell(e), \\ W_e(\mathscr{G}) &= \sum_{e \in \mathscr{E}_c} \mathscr{G}(e). \end{aligned}$$

This shows that $W_s(\mathscr{G})$ counts the "number of symbols" in the canonical decomposition of $\mathscr{G}$, whereas $W_e(\mathscr{G})$ counts the number of expressions (that is $\mathscr{G}(\mathscr{E}_c)$, the

weight put by the grammar on canonical expressions). We can also see from the definitions that $W_l(\mathscr{G})$ counts the number of canonical expressions starting with a positive (that is non terminal) label, that we will call for short the number of labels, and that $W_w(\mathscr{G})$ counts the number of words.

Since a split increases the number of canonical expressions by one, the number of symbols in canonical expressions by two, the number of labels by one, and keeps the number of words constant, whereas a merge decreases these quantities in the same proportions, the following quantities are invariant in all the toric grammars involved: for any $\mathscr{G} \in \mathfrak{G}$ such that $\sum_{t\in\mathbb{N}} \mathbb{P}\Big(G_t = \mathscr{G}\Big) > 0$,

$$\begin{aligned} W_s(\mathscr{G}) - 2W_e(\mathscr{G}) &= W_s(\mathscr{T}) - 2W_e(\mathscr{T}), \\ W_e(\mathscr{G}) - W_l(\mathscr{G}) &= W_e(\mathscr{T}) - W_l(\mathscr{T}) = W_e(\mathscr{T}), \\ W_w(\mathscr{G}) &= W_w(\mathscr{T}). \end{aligned}$$

Moreover, for the same reasons, for any $\mathscr{T}' \in \mathfrak{T}$ and $\mathscr{G} \in \mathfrak{G}$ such that $\sum_{t\in\mathbb{N}} \mathbb{P}\Big(G_{2t} = \mathscr{T}'\Big) > 0$ and $\sum_{t\in\mathbb{N}} \mathbb{P}\Big(G_{2t+1} = \mathscr{G}\Big) > 0$,

$$\begin{aligned} \mathbb{P}\Big(\tau = W_l(S_\tau) \,\Big|\, S_0 = \mathscr{T}'\Big) &= 1, \\ \mathbb{P}\Big(\sigma = W_l(\mathscr{G}) \,\Big|\, P_0 = \mathscr{G}, P_\sigma \in \mathfrak{T}\Big) &= 1. \end{aligned}$$

Thus, we will prove the lemma if we can bound $W_l(\mathscr{G})$ (or equivalently $W_l(S_\tau)$ when $S_0 = \mathscr{T}'$, since S_τ almost surely satisfies the conditions imposed on $\mathscr{G}$). We can then remark that

$$\begin{aligned} \sum_{e\in\mathscr{E}_c} \mathscr{G}(e)\mathbb{1}\Big[\ell(e) \geqslant 3\Big] &\leqslant \sum_{e\in\mathscr{E}_c} \mathscr{G}(e)\Big[\ell(e) - 2\Big] = W_s(\mathscr{G}) - 2W_e(\mathscr{G}), \\ \sum_{e\in\mathscr{E}_c} \mathscr{G}(e)\mathbb{1}\Big[\ell(e) = 2\Big] &= \sum_{e\in\mathscr{E}} \mathscr{G}(e)\mathbb{1}\Big[\ell(e) = 2\Big] \sum_{w\in D} \mathbb{1}\Big(e \in wS^*\Big) \\ &\leqslant \sum_{e\in\mathscr{E}} \mathscr{G}(e) \sum_{w\in D} \mathbb{1}\Big(e \in wS^*\Big) = W_w(\mathscr{G}), \end{aligned}$$

because any canonical expression of length 2 is of the form $e = [_i w$, with $i \in \mathbb{N}$ and $w \in D$, so that for any $e \in \mathscr{E}_c$ of length 2,

$$\sum_{e'\in\mathfrak{S}(e)} \sum_{w\in D} \mathbb{1}\Big(e' \in wS^*\Big) = 1.$$

Thus

$$W_e(\mathscr{G}) \leqslant W_w(\mathscr{G}) + W_s(\mathscr{G}) - 2W_e(\mathscr{G}),$$

and consequently we can bound $W_l(\mathscr{G})$ by the split and merge invariant bound

$$W_l(\mathscr{G}) \leqslant W_l(\mathscr{G}) - W_e(\mathscr{G}) + W_w(\mathscr{G}) + W_s(\mathscr{G}) - 2W_e(\mathscr{G}).$$

This, added to the fact that $W_l(\mathscr{T}) = 0$ and $W_s(\mathscr{T}) = W_w(\mathscr{T}) + W_e(\mathscr{T})$, proves that

$$W_l(\mathscr{G}) \leqslant 2\Big[W_w(\mathscr{T}) - W_e(\mathscr{T})\Big].$$

This ends the proof, since $W_w(\mathscr{T}) = \mathscr{T}(DS^*)$ and $W_e(\mathscr{T}) = \mathscr{T}([_0 S^*)$. □

1.A.2 Parsing Relations

Proof of lemma 1.4.1 on page 36. The implication

$$\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0 \implies \mathbb{G}_{\mathscr{T},\mathscr{G}}(\mathscr{G}) > 0$$

is less trivial than it may seem. Indeed we can reverse the path of the splitting process S_t, be it a parsing or a learning process, to obtain a path followed with positive probability by the production process, but reversing the production process does not give a parsing process. Let us illustrate this difficulty on a simple example. Consider

$$\mathscr{T} = 1 \otimes [_0 abcd \quad \text{and} \quad \mathscr{G} = [_0 a]_1 \oplus [_1 b]_2 \oplus [_2 c]_3 \oplus [_3 d.$$

The production path

$$\mathscr{G}, \quad [_0\, ab\,]_2 \oplus [_2\, c\,]_3 \oplus [_3\, d, \quad [_0\, ab\,]_2 \oplus [_2\, cd, \quad \mathscr{T}$$

has positive probability. The reverse path may have a positive probability for the learning process but not for the parsing process with reference $\mathscr{G}$, since none of the expressions $[_0\, ab\,]_2$ or $[_2\, cd$ belongs to the support of $\mathscr{G}$. To parse $\mathscr{T}$ according to $\mathscr{G}$, one can instead follow with positive probability such a path as

$$\mathscr{T}, \quad [_0\, abc\,]_3 \oplus [_3\, d, \quad [_0\, ab\,]_2 \oplus [_2\, c\,]_3 \oplus [_3\, d, \quad \mathscr{G}.$$

To prove the lemma, we will have to show that it is always possible to find such an alternative parsing path. This property is fundamental to our approach, since it proves that the toric grammars we build can be used to parse the texts they can produce.

Let us start with the easiest part of the proof. Assume that $\mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0$. This means that there is a path $\mathscr{G}_0, \ldots, \mathscr{G}_k$ such that $\mathscr{G}_0 = \mathscr{T}$, $\mathscr{G}_k = \mathscr{G}$, and at each step $\mathscr{G}_t \in \beta_n(\mathscr{G}_{t-1}, \mathscr{R})$. Anyhow it is easy to check that

$$\mathscr{G}_t \in \beta_n(\mathscr{G}_{t-1}, \mathscr{R}) \implies \mathscr{G}_{t-1} \in \alpha(\mathscr{G}_t),$$

so that the reverse path is followed with a positive probability by the production process. This means that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$.

In the case of the learning process, if $\mathbb{G}_{\mathscr{T}}(\mathscr{G}) > 0$, there is a path $\mathscr{G}_t, 0 \leqslant t \leqslant k$, such that $\mathscr{G}_t \in \beta_\ell(\mathscr{G}_{t-1})$, $\mathscr{G}_0 = \mathscr{T}$ and $\mathscr{G}_k = \mathscr{G}$, consequently there is a label map $f_t \in \mathfrak{F}$ such that $f_t(\mathscr{G}_{t-1}) \in \alpha(\mathscr{G}_t)$. We can then remark that

$$f_k \circ \cdots \circ f_t(\mathscr{G}_{t-1}) \in \alpha\Big(f_k \circ \cdots \circ f_{t+1}(\mathscr{G}_t)\Big),$$

because as already proved before in lemma 1.3.3 on page 28, $f\Big(\alpha(\mathscr{G})\Big) \subset \alpha\Big(f(\mathscr{G})\Big)$. Let us consider the path $\widetilde{\mathscr{G}_t} = f_k \circ \cdots \circ f_{k-t+1}(\mathscr{G}_{k-t})$. It begins at $\widetilde{\mathscr{G}_0} = \mathscr{G}_k = \mathscr{G}$ and ends at $\widetilde{\mathscr{G}_k} = f_k \circ \cdots \circ f_1(\mathscr{G}_0) = \mathscr{T}$. According to the previous remark, this path is followed by the production process with positive probability, proving that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$.

Let us now come to the proof of the third implication of the lemma. For this let us assume now that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$. Consider a path $\mathscr{G}_0, \ldots, \mathscr{G}_k$ such that $\mathscr{G}_0 = \mathscr{G}, \ldots, \mathscr{G}_k = \mathscr{T}$ and $\mathscr{G}_t \in \alpha(\mathscr{G}_{t-1})$. We are going to define some *decorated* path $\widetilde{\mathscr{G}_0}, \ldots, \widetilde{\mathscr{G}_k}$ with some added parentheses. Introduce a new set of symbols $B = \Big\{(_i\ ,\ )_i, i \in \mathbb{N} \setminus \{0\}\Big\}$ and assume that it is disjoint from the other symbols used so far, so that $B \cap S = \varnothing$. Consider the set of toric grammars $\widetilde{\mathfrak{S}}$ based on the enlarged dictionary $D \cup B$, and the projection $\pi : \widetilde{\mathfrak{G}} \to \mathfrak{G}$ defined with the help of the canonical decomposition of toric grammars as

$$\pi\left(\sum_{e \in \widetilde{\mathscr{E}_c}} \mathscr{G}(e) \otimes e\right) = \sum_{e \in \widetilde{\mathscr{E}_c}} \mathscr{G}(e) \otimes \pi(e),$$

where $\widetilde{\mathscr{E}_c}$ is the set of canonical expressions based on the enlarged dictionary $D \cup B$, and where $\pi(e)$ is obtained by removing from the sequence of symbols e the symbols belonging to the decoration set B (that is the parentheses).

Let us put $\widetilde{\mathscr{G}_0} = \mathscr{G}$ and define $\widetilde{\mathscr{G}_t}$ for $t = 1, \ldots, k$ by induction. We will check on the go that $\pi(\widetilde{\mathscr{G}_t}) = \mathscr{G}_t$. It is obviously true for $\widetilde{\mathscr{G}_0}$, because $\widetilde{\mathscr{G}_0} \in \mathfrak{S}$, so that $\pi(\widetilde{\mathscr{G}_0}) = \widetilde{\mathscr{G}_0} = \mathscr{G}_0$. That said, let us describe the construction of $\widetilde{\mathscr{G}_t}$, assuming that $\widetilde{\mathscr{G}_{t-1}}$ is already defined, and satisfies $\pi(\widetilde{\mathscr{G}_{t-1}}) = \mathscr{G}_{t-1}$. Consider the sequence of symbols a and $b \in S^*$ and the index $i \in \mathbb{N} \setminus \{0\}$ such that

$$\mathscr{G}_t = \mathscr{G}_{t-1} \oplus ab \ominus a\,]_i \ominus [_i\, b.$$

Since $\pi(\widetilde{\mathscr{G}_{t-1}}) = \mathscr{G}_{t-1}$, and since $a\,]_i \oplus [_i\, b \leqslant \mathscr{G}_{t-1}$, there are $\widetilde{a} \in \widetilde{S}^*$ and $\widetilde{b} \in \widetilde{S}^*$ such that $\pi(\widetilde{a}) = a$, $\pi(\widetilde{b}) = b$, and $\widetilde{a}\,]_i \oplus [_i\, \widetilde{b} \leqslant \widetilde{\mathscr{G}_{t-1}}$. (The choice of $\widetilde{a}$ and $\widetilde{b}$ may not be unique, in which case we can make any arbitrary choice). Let us define

$$\widetilde{\mathscr{G}_t} = \widetilde{\mathscr{G}_{t-1}} \oplus \widetilde{a}(_i\, \widetilde{b}\,)_i \ominus \widetilde{a}\,]_i \ominus [_i\, \widetilde{b}.$$

Since $\pi\Big(\widetilde{a}(_i\, \widetilde{b}\,)_i\Big) = \pi\Big(\widetilde{a}\widetilde{b}\Big) = ab$,

$$\pi\Big(\widetilde{\mathscr{G}_t}\Big) = \pi\Big(\widetilde{G}_{t-1}\Big) \oplus \pi\Big(\widetilde{a}(_i\, b\,)_i\Big) \ominus \pi\Big(\widetilde{a}]_i\Big) \ominus \pi\Big([_i\widetilde{b}\Big) = \mathscr{G}_{t-1} \oplus ab \ominus a\,]_i \ominus [_i\, b = \mathscr{G}_t,$$

where we have used the obvious fact that π is linear.

We are now going to define another mapping between grammars that allows to recover $\mathscr{G}$ from any $\widetilde{\mathscr{G}_t}$ (obviously the decorations where added to keep track of $\mathscr{G}$). Let us define $\psi : \widetilde{\mathfrak{S}}' \to \mathfrak{S}$ on the set of decorated grammars $\widetilde{\mathfrak{S}}'$ which are supported by expressions where the parentheses $(_i\,)_i$ are matched (at the same level) by the formula

$$\psi\Big(\sum_{e\in\widetilde{\mathscr{E}_c}} \widetilde{\mathscr{G}}(e)\otimes e\Big) = \sum_{e\in\widetilde{\mathscr{E}_c}} \widetilde{\mathscr{G}}(e)\psi(e),$$

where $\psi(e)$ is defined by the rules

$$\psi(e) = \begin{cases} \psi\big([_i a\,]_j c\big) + \psi\big([_j\, b\big), & \text{if } e = [_i\, a(_j\, b)_j c, \text{ with } a,b,c \in \widetilde{S}^* \\ \psi(e) = 1\otimes e, & \text{otherwise.} \end{cases}$$

It is easy to check that this definition is not ambiguous and that

$$\psi(e) = \psi'(e) \oplus \bigoplus_{(_i a)_i \in \mathrm{supp}(e)} [_i \psi'(a),$$

where $\psi'(e)$ is the expression obtained from e by replacing all the sequences between outer parentheses pairs $(_j a)_j$ by $]_j$. This is may be easier to grasp on some example:

$$\psi\big([_0 a(_1 b(_2 c)_2 d)_1 e(_3 f(_4 g)_4)_3 h\big) = [_0 a\,]_1 e\,]_3 h \oplus [_1\, b\,]_2 d \oplus [_2\, c \oplus [_3\, f\,]_4 \oplus [_4\, g.$$

It is easy to check by induction that $\widetilde{\mathscr{G}_t} \in \widetilde{\mathfrak{S}}'$. Moreover, we have that $\psi\big(\widetilde{\mathscr{G}_t}\big) = \mathscr{G}$. Indeed $\psi\big(\widetilde{\mathscr{G}_0}\big) = \psi\big(\mathscr{G}\big) = \mathscr{G}$ and

$$\psi\big(\widetilde{G}_t\big) = \psi\big(\widetilde{G}_{t-1}\big) \oplus \psi\big(\widetilde{a}(_i \widetilde{b})_i\big) \ominus \psi\big(\widetilde{a}\,]_i\big) \ominus \psi\big([_i\, \widetilde{b}\big) = \psi\big(\widetilde{G}_{t-1}\big),$$

since ψ is linear and $\psi\big(\widetilde{a}(_i\, \widetilde{b})_i\big) = \psi\big(\widetilde{a}\,]_i\big) \oplus \psi\big([_i\, b\big)$.

We are now going to define a continuation for the path $\big(\widetilde{\mathscr{G}_t}, 0 \leqslant t \leqslant k\big)$ that will bring us back to $\mathscr{G}$.

We will maintain during our inductive construction two properties:

$$\psi\big(\widetilde{\mathscr{G}_t}\big) = \mathscr{G},$$
$$\text{and} \quad \mathrm{supp}\big(\widetilde{\mathscr{G}_t}\big) \cap \widetilde{\mathscr{E}_l} \subset \mathscr{E}_l,$$

where $\widetilde{\mathscr{E}_l}$ is the set of local decorated expressions, so that

$$\widetilde{\mathscr{E}_l} = \big\{e \in \widetilde{\mathscr{E}} : [_0 \notin \mathrm{supp}(e)\big\}.$$

We already proved that the first property is satisfied by $\widetilde{\mathscr{G}_k}$. As $\pi(\widetilde{\mathscr{G}_k}) = \mathscr{G}_k = \mathscr{T}$, $\mathrm{supp}(\widetilde{\mathscr{G}_k}) \cap \widetilde{\mathscr{E}_l} = \varnothing$, so that the second condition is also satisfied. Let us assume that, for some $t > k$, $\widetilde{\mathscr{G}_{t-1}}$ has been defined and satisfies the two conditions above, and let us proceed to the construction of $\widetilde{\mathscr{G}_t}$.

As long as $\widetilde{\mathscr{G}_{t-1}} \notin \mathfrak{S}$, (and this will be the case for $t < 2k$), find some canonical expression $e \in \widetilde{\mathscr{E}_c} \setminus \mathscr{E}_c$, such that $\widetilde{\mathscr{G}_{t-1}}(e) \geqslant 1$. From our induction hypotheses, we see that necessarily $\lfloor_0 \in \mathrm{supp}(e)$. Our continuation will be such that each such expression has matching parentheses with matching labels, and we will check this on the go while building it by induction. Among those matching pairs of parentheses, there is necessarily at least one inner pair. We can for instance choose the one starting with the last opening parenthesis $(_j$ of the sequence e. This choice makes it obvious that the subsequence of e enclosed between $(_j$ and $)_j$ contains no further parentheses.

Since ψ is linear and preserves positive measures,

$$\mathscr{G} \ominus \psi(e) = \psi\Big(\widetilde{\mathscr{G}_{t-1}}\Big) \ominus \psi(e) = \psi\Big(\widetilde{\mathscr{G}_{t-1}} \ominus e\Big) \geqslant 0.$$

On the other hand, e has the form $e = \lfloor_0 a (_j b)_j c$, where $\psi(b) = b$ (since $(_j)_j$ is an inner pair of parentheses in e). As $\psi(e) = \psi\Big(\lfloor_0 a \rfloor_j c\Big) + \psi\Big(\lfloor_j b\Big)$ and $\psi\Big(\lfloor_j b\Big) = \lfloor_j b$, this shows that $\lfloor_j b \leqslant \mathscr{G}$, and therefore that $\mathscr{G}(\lfloor_j b) > 0$. Let us now define

$$\widetilde{\mathscr{G}_t} = \widetilde{\mathscr{G}_{t-1}} \ominus e \oplus \lfloor_j b \oplus \lfloor_0 a \rfloor_j c.$$

Applying ψ to $\widetilde{\mathscr{G}_t}$, we see as previously that $\psi(\widetilde{\mathscr{G}_t}) = \psi(\widetilde{\mathscr{G}_{t-1}}) = \mathscr{G}$. As $\widetilde{\mathscr{G}_k}$ contains k pairs of parentheses, and we consume one pair at each step $t > k$, we see that $\widetilde{\mathscr{G}_{2k}}$ contains no more parentheses, so that $\widetilde{\mathscr{G}_{2k}} \in \mathfrak{G}$ and $\widetilde{\mathscr{G}_{2k}} = \psi(\widetilde{\mathscr{G}_{2k}}) = \mathscr{G}$. Let us put now $\mathscr{G}_t = \pi(\widetilde{\mathscr{G}_t})$, for $t = k+1, \ldots, 2k$. We see that

$$\mathscr{G}_t = \mathscr{G}_{t-1} \ominus \lfloor_0 abc \oplus \lfloor_0 a \rfloor_j c \oplus \lfloor_j b,$$

where $\lfloor_0 abc, \lfloor_0 a \rfloor_j c \in \mathscr{E}$ and $\mathscr{G}\Big(\lfloor_j b\Big) > 0$, so that $\mathscr{G}_t \in \beta_n(\mathscr{G}_{t-1}, \mathscr{G})$, therefore $\mathscr{G}_k = \mathscr{T}, \ldots, \mathscr{G}_{2k} = \mathscr{G}$ is a path of positive probability under the parsing process with reference $\mathscr{G}$, leading from $\mathscr{T}$ to $\mathscr{G}$, in other words, $\mathbb{G}_{\mathscr{T},\mathscr{G}}(\mathscr{G}) > 0$ as required.□

1.A.3 Convergence to the expectation of a random toric grammar

We will here prove lemma 1.5.1 on page 38. This result is based on the fact that the operation

$$\Big(\mathscr{G}, \mathscr{G}'\Big) \mapsto \chi\Big(\mathscr{G} \boxplus \mathscr{G}'\Big)$$

is associative.

Let us begin the proof by several definitions and lemmas.

For any grammar $\mathscr{G} \in \mathfrak{G}$ and any pair of indices $p = (p^1, p^2) \in \Big(\mathbb{N} \setminus \{0\}\Big)^2$, we will say that p is $\mathscr{G}$-congruent when there is $a \in S^*$ such that $\mathscr{G}(a\big]_{p^1})\mathscr{G}(a\,\big]_{p^2}) > 0$ or $\mathscr{G}(\big[_{p^1}\, a)\mathscr{G}(\big[_{p^2}\, a) > 0$.

Let us define the label map ξ_p as

$$\xi_p(i) = \begin{cases} i, & \text{when } i \notin \{p^1, p^2\}, \\ \min\{p^1, p^2\}, & \text{when } i \in \{p^1, p^2\}. \end{cases}$$

For any sequence $p_1, \ldots, p_k \in (\mathbb{N} \setminus \{0\})^{2k}$ of pairs of indices, let us define the label map $\xi_{p_1,\ldots,p_k}$ as

$$\xi_{p_1,\ldots,p_k} = \xi_{\xi_{p_1,\ldots,p_{k-1}}(p_k)} \circ \xi_{p_1,\ldots,p_{k-1}},$$

where $f\Big((i,j)\Big) = (f(i), f(j))$, for any $(i,j) \in \Big(\mathbb{N} \setminus \{0\}\Big)^2$.

Let us say that $(p_1, \ldots, p_k) \in \Big(\mathbb{N} \setminus \{0\}\Big)^{2k}$ is $\mathscr{G}$-congruent, if $\xi_{p_1,\ldots,p_{j-1}}(p_j)$ is $\xi_{p_1,\ldots,p_{j-1}}(\mathscr{G})$-congruent for any $j \leqslant k$, and that it is maximal $\mathscr{G}$-congruent is it is $\mathscr{G}$-congruent and any $\mathscr{G}$-congruent sequence of the form $(p_1, \ldots, p_k, p_{k+1})$ is such that

$$\xi_{p_1,\ldots,p_k}(p^1_{k+1}) = \xi_{p_1,\ldots,p_k}(p^2_{k+1}),$$

or equivalently such that $\xi_{p_1,\ldots,p_{k+1}} = \xi_{p_1,\ldots,p_k}$.

Lemma 1.A.1
For any sequence $(p_1, \ldots, p_\ell) \in \Big(\mathbb{N} \setminus \{0\}\Big)^{2\ell}$, *for any* $k < \ell$,

$$\xi_{p_1,\ldots,p_\ell} = \xi_{\xi_{p_1,\ldots,p_k}(p_{k+1},\ldots,p_\ell)} \circ \xi_{p_1,\ldots,p_k}.$$

Proof. By induction on ℓ for k fixed. This is true from the definition for $\ell = k+1$. Assuming we have established the lemma for $\ell - 1$, we can write

$$\begin{aligned} \xi_{p_1,\ldots,p_\ell} &= \xi_{\xi_{p_1,\ldots,p_{\ell-1}}(p_\ell)} \circ \xi_{p_1,\ldots,p_{\ell-1}} \\ &= \xi_{\xi_{\xi_{p_1,\ldots,p_k}(p_{k+1},\ldots,p_{\ell-1})} \circ \xi_{p_1,\ldots,p_k}(p_\ell)} \circ \xi_{\xi_{p_1,\ldots,p_k}(p_{k+1},\ldots,p_{\ell-1})} \circ \xi_{p_1,\ldots,p_k} \\ &= \xi_{\xi_{p_1,\ldots,p_k}(p_{k+1},\ldots,p_\ell)} \circ \xi_{p_1,\ldots,p_k}, \end{aligned}$$ □

Lemma 1.A.2
For any permutation σ *of* $\{1, \ldots, k\}$,

$$\xi_{p_1,\ldots,p_k} \equiv \xi_{p_{\sigma(1)},\ldots,p_{\sigma(k)}}$$

PROOF. Let us consider the smallest equivalence relation containing the set $\{p_1, \dots, p_k\}$. Let $\overline{\pi}_k : \mathbb{N} \setminus \{0\} \to \mathscr{C}$ be the corresponding projection of each label to its component. Let us befine the label map π_k by $\pi_k(0) = 0$ and

$$\pi_k(i) = \min \overline{\pi}_k(i), \qquad i > 0.$$

We are going to prove by induction on k that $\xi_{p_1,\dots,p_k} \equiv \pi_k$. Since π_k is invariant by permutation of the sequence $(p_1, \dots, p_k)$, this will prove the lemma.

Let us remark now that $\xi_{p_1,\dots,p_k} \equiv \pi_k$ if and only if

$$\xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j) \iff \pi_k(i) = \pi_k(j), \qquad i, j > 0.$$

So we are going to prove this equivalence. It is easy to see from the previous lemma that for any integer $m = 1, \dots, k$,

$$\xi_{p_1,\dots,p_k}(p_m^1) = \xi_{p_1,\dots,p_k}(p_m^2). \tag{1.A.1}$$

Indeed,

$$\xi_{p_1,\dots,p_k} = \xi_{\xi_{p_1,\dots,p_m}(p_{m+1},\dots,p_k)} \circ \xi_{\xi_{p_1,\dots,p_{m-1}}(p_m)} \circ \xi_{p_1,\dots,p_{m-1}},$$

so that, changing p_m for $\xi_{p_1,\dots,p_{m-1}}(p_m)$, we are back to proving the result when $m = k = 1$, where it is obvious from the definitions.

Now, eq. (1.A.1) on the current page and the minimality of π_k implies that

$$\pi_k(i) = \pi_k(j) \implies \xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j), \qquad i, j > 0.$$

Let us assume conversely that $\xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j)$ and let

$$m = \min\Big\{\ell : \xi_{p_1,\dots,p_\ell}(i) = \xi_{p_1,\dots,p_\ell}(j)\Big\}.$$

Since $\xi_{p_1,\dots,p_m} = \xi_{\xi_{p_1,\dots,p_{m-1}}(p_m)} \circ \xi_{p_1,\dots,p_{m-1}}$, we see that necessarily

$$\xi_{p_1,\dots,p_{m-1}}\Big(\{i, j\}\Big) = \xi_{p_1,\dots,p_{m-1}}\Big(\{p_m^1, p_m^2\}\Big),$$

and that this set contains two distinct elements. Exchanging the role of i and j if necessary, we can assume without loss of generality that

$$\xi_{p_1,\dots,p_{m-1}}\Big((i, j)\Big) = \xi_{p_1,\dots,p_{m-1}}\Big(p_m\Big).$$

From the induction hypothesis, this implies that $\pi_{m-1}\Big((i, j)\Big) = \pi_{m-1}(p_m)$. Since the equivalence relation defined by π_{m-1} is a subset of the equivalence relation defined by π_k, this implies that $\pi_k\Big((i, j)\Big) = \pi_k(p_m)$. Since moreover we have that $\pi_k(p_m^1) = \pi_k(p_m^2)$, this implies that $\pi_k(i) = \pi_k(j)$. □

Lemma 1.A.3
For any $f \in \mathfrak{F}$, any sequence of pairs of positive labels $p_1, \dots, p_k$, there is a label map $g \in \mathfrak{F}$ such that

$$\xi_{f(p_1,\dots,p_k)} \circ f = g \circ \xi_{p_1,\dots,p_k}.$$

Proof. We have to prove that

$$\xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j) \implies \xi_{f(p_1,\dots,p_k)} \circ f(i) = \xi_{f(p_1,\dots,p_k)} \circ f(j), \qquad i,j > 0.$$

From the proof of the previous lemma, it is enough to check that the right-hand side holds when $(i,j) = p_m$, $m = 1, \dots, k$, which is then obvious. □

Lemma 1.A.4
If $f \in \mathfrak{F}$ and $(p_1, \dots, p_k)$ is $\mathscr{G}$-congruent, then the sequence $(f(p_1), \dots, f(p_k))$ is also $f(\mathscr{G})$-congruent.

Proof. Assume that for some $a \in S^*$

$$\xi_{p_1,\dots,p_{m-1}}(\mathscr{G})\Big(a\,]_{\xi_{p_1,\dots,p_{m-1}}(p^1_m)}\Big) > 0.$$

Then, $\xi_{f(p_1,\dots,p_{m-1})} \circ f = g \circ \xi_{p_1,\dots,p_{m-1}}$, and

$$\begin{aligned}
\xi_{f(p_1,\dots,p_{m-1})} \circ f(\mathscr{G})\Big(g(a)\,]_{\xi_{f(p_1,\dots,p_{m-1})} \circ f(p^1_m)}\Big) & \\
&= g \circ \xi_{p_1,\dots,p_{m-1}}(\mathscr{G})\Big(g(a)\,]_{g \circ \xi_{p_1,\dots,p_{m-1}}(p^1_m)}\Big) \\
&= \xi_{p_1,\dots,p_m}(\mathscr{G})\Big(g^{-1} \circ g\Big(a\,]_{\xi_{p_1,\dots,p_{m-1}}(p^1_m)}\Big)\Big) \\
&\geqslant \xi_{p_1,\dots,p_{m-1}}(\mathscr{G})\Big(a\,]_{\xi_{p_1,\dots,p_{m-1}}(p^1_m)}\Big) > 0.
\end{aligned}$$

The same is true when p^1_m is replaced with p^2_m and when $a\,]_{\xi_{p_1,\dots,p_{m-1}}(p^1_m)}$ is replaced with $a[_{\xi_{p_1,\dots,p_{m-1}}(p^2_m)}$.

The lemma is a straightforward consequence of these remarks and the definition of a congruent sequence. □

Lemma 1.A.5
If $(p_1, \dots, p_k)$ and $(q_1, \dots, q_\ell)$ are both $\mathscr{G}$-congruent, then

$$(p_1, \dots, p_k, q_1, \dots, q_\ell)$$

is $\mathscr{G}$-congruent.

PROOF. According to the previous lemma, we know that $\xi_{p_1,\dots,p_k}(q_1,\dots,q_\ell)$ is $\xi_{p_1,\dots,p_k}\Big(\mathscr{G}\Big)$-congruent. Coming back to the definition this proves that

$$\xi_{\xi_{p_1,\dots,p_k}(q_1,\dots,q_{\ell-1})} \circ \xi_{p_1,\dots,p_k}(q_\ell)$$

is

$$\xi_{\xi_{p_1,\dots,p_k}(q_1,\dots,q_{\ell-1})} \circ \xi_{p_1,\dots,p_k}\Big(\mathscr{G}\Big)\text{-congruent.}$$

In lemma 1.A.1 on page 55 we have moreover proved that

$$\xi_{p_1,\dots,p_k,q_1,\dots,q_{\ell-1}} = \xi_{\xi_{p_1,\dots,p_k}(q_1,\dots,q_{\ell-1})} \circ \xi_{p_1,\dots,p_k}.$$

This identity applied to the above statement shows that $(p_1,\dots,p_k,q_1,\dots,q_\ell)$ satisfies the definition of a $\mathscr{G}$-congruent sequence. □

Proposition 1.A.6
If $(p_1,\dots,p_k)$ and $(q_1,\dots,q_\ell)$ are both maximal $\mathscr{G}$-congruent, then

$$\xi_{p_1,\dots,p_k}\Big(\mathscr{G}\Big) \equiv \xi_{q_1,\dots,q_\ell}\Big(\mathscr{G}\Big) \equiv \chi\Big(\mathscr{G}\Big).$$

PROOF. From the previous lemma, $(p_1,\dots,p_k,q_1,\dots,q_\ell)$ is $\mathscr{G}$-congruent. Since p is maximal, $\xi_{p_1,\dots,p_k,q_1,\dots,q_\ell} = \xi_{p_1,\dots,p_k}$. In the same way $\xi_{q_1,\dots,q_\ell,p_1,\dots,p_k} = \xi_{q_1,\dots,q_\ell}$. We have seen moreover in a previous lemma that

$$\xi_{p_1,\dots,p_k,q_1,\dots,q_\ell} \equiv \xi_{q_1,\dots,q_\ell,p_1,\dots,p_k}.$$

This proves that $\xi_{p_1,\dots,p_k} \equiv \xi_{q_1,\dots,q_\ell}$.

We see from the definition of χ (see definition 1.3.10 on page 34) that there is some maximal $\mathscr{G}$-congruent sequence $r_1,\dots,r_m$ such that $\chi(\mathscr{G}) = \xi_{r_1,\dots,r_m}(\mathscr{G})$. Therefore

$$\chi\Big(\mathscr{G}\Big) \equiv \xi_{p_1,\dots,p_k}\Big(\mathscr{G}\Big) \equiv \xi_{q_1,\dots,q_\ell}\Big(\mathscr{G}\Big).$$ □

Proposition 1.A.7
For any $\mathscr{G},\mathscr{G}' \in \mathfrak{G}$,

$$\chi\Big(\chi(\mathscr{G}) \boxplus \mathscr{G}'\Big) = \chi\Big(\mathscr{G} \boxplus \mathscr{G}'\Big).$$

Consequently, for any $\mathscr{G},\mathscr{G}',\mathscr{G}'' \in \mathfrak{G}$,

$$\chi\Big(\chi\Big(\mathscr{G} \boxplus \mathscr{G}'\Big) \boxplus \mathscr{G}''\Big) = \chi\Big(\mathscr{G} \boxplus \mathscr{G}' \boxplus \mathscr{G}''\Big).$$

Proof. Let us assume that $\mathscr{G}$, $\mathscr{G}'$ and $\chi(\mathscr{G})$ use disjoint label sets, so that

$$\chi\Big(\chi\Big(\mathscr{G}\Big) \boxplus \mathscr{G}'\Big) \equiv \chi\Big(\chi(\mathscr{G}) + \mathscr{G}'\Big),$$
$$\chi\Big(\mathscr{G} \boxplus \mathscr{G}'\Big) \equiv \chi\Big(\mathscr{G} + \mathscr{G}'\Big).$$

Let $p_1, \dots, p_k$ be some maximal $\mathscr{G}$-congruent sequence. It is then obviously also $(\mathscr{G} + \mathscr{G}')$-congruent, and since label sets are disjoint,

$$\xi_{p_1,\dots,p_k}\Big(\mathscr{G}\Big) + \mathscr{G}' = \xi_{p_1,\dots,p_k}\Big(\mathscr{G} + \mathscr{G}'\Big).$$

Let us continue the sequence $p_1, \dots, p_k$ to form a maximal $\Big(\mathscr{G} + \mathscr{G}'\Big)$-congruent sequence $p_1, \dots, p_\ell$. Let $(q_{k+1}, \dots, q_\ell)$ be defined as

$$q_m = \xi_{p_1,\dots,p_k}(p_{k+m}).$$

It now follows from the definitions that the sequence $(q_{k+1}, \dots, q_\ell)$ is a maximal $\xi_{p_1,\dots,p_k}(\mathscr{G} + \mathscr{G}')$-congruent sequence, and a maximal $\Big(\xi_{1,\dots,p_k}(\mathscr{G}) + \mathscr{G}'\Big)$-congruent sequence by consequence. Consequently

$$\begin{aligned}\chi\Big(\chi\Big(\mathscr{G}\Big) + \mathscr{G}'\Big) &\equiv \xi_{q_{k+1},\dots,q_\ell}\Big(\xi_{p_1,\dots,p_k}\Big(\mathscr{G}\Big) + \mathscr{G}'\Big) \\ &= \xi_{q_{k+1},\dots,q_\ell} \circ \xi_{p_1,\dots,p_k}\Big(\mathscr{G} + \mathscr{G}'\Big) = \xi_{\xi_{p_1,\dots,p_k}(p_{k+1},\dots,p_\ell)} \circ \xi_{p_1,\dots,p_k}\Big(\mathscr{G} + \mathscr{G}'\Big) \\ &= \xi_{p_1,\dots,p_\ell}\Big(\mathscr{G} + \mathscr{G}'\Big) \equiv \chi\Big(\mathscr{G} + \mathscr{G}'\Big),\end{aligned}$$

proving the proposition. □

Proof of lemma 1.5.1 on page 38. Let π be the projection of $\mathfrak{G}$ on $\mathfrak{G}/\!\equiv$.

From the law of large numbers, we have that, for all $\mathscr{G} \in \mathfrak{G}$,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(G_i \equiv \mathscr{G}) \underset{n\to\infty}{\longrightarrow} \mathbb{G}(\pi(\mathscr{G})).$$

Let us now remark that $\boxplus_{i=1}^{n} n^{-1}G_i = \boxplus_{\overline{\mathscr{G}} \in \mathfrak{G}/\equiv} \boxplus_{\substack{i \\ G_i \in \overline{\mathscr{G}}}} n^{-1}G_i$. Thus

$$\begin{aligned}\frac{1}{n}\chi\left(\boxplus_{i=1}^{n} G_i\right) &= \chi\left(\boxplus_{\overline{\mathscr{G}} \in \mathfrak{G}/\equiv} \chi\left(\boxplus_{\substack{i \\ G_i \in \overline{\mathscr{G}}}} n^{-1}G_i\right)\right) \\ &= \chi\left(\boxplus_{\overline{\mathscr{G}} \in \mathfrak{G}/\equiv} \left(\sum_{i=1}^{n} n^{-1}\mathbb{1}(G_i \in \overline{\mathscr{G}})\right)\chi(\overline{\mathscr{G}})\right) \\ &= \chi\left(\boxplus_{\overline{\mathscr{G}} \in \mathfrak{G}/\equiv} \left(\sum_{i=1}^{n} n^{-1}\mathbb{1}(G_i \in \overline{\mathscr{G}})\right)\overline{\mathscr{G}}\right).\end{aligned}$$

We used here proposition 1.A.7 on page 58, together with the fact that for any numbers $a, b \in \mathbb{R}_+$,

$$\chi\Big[(a\,\mathscr{G}) \boxplus (b\,\mathscr{G})\Big] = (a+b)\,\chi(\mathscr{G}),$$

which comes from the following reasoning: Suppose that

$$\{1, \dots, d\} = \{i\,;\ \mathscr{G}([_i S^*) > 0\},$$

and let $p_i = (2i, 2i-1)$. Since each p_i is $(a\,\mathscr{G}) \boxplus (b\,\mathscr{G})$-congruent, $(p_1, \dots, p_d)$ is also $(a\,\mathscr{G}) \boxplus (b\,\mathscr{G})$-congruent, from lemma 1.A.5 on page 57. It is quite straightforward to see that

$$\xi_{p_1,\dots,p_d}\Big[(a\,\mathscr{G}) \boxplus (b\,\mathscr{G})\Big] \equiv (a+b)\,\mathscr{G}.$$

This implies that

$$\chi\Big[(a\,\mathscr{G}) \boxplus (b\,\mathscr{G})\Big] = \chi \circ \xi_{p_1,\dots,p_d}\Big[(a\,\mathscr{G}) \boxplus (b\,\mathscr{G})\Big] = \chi\Big[(a+b)\,\mathscr{G}\Big] = (a+b)\,\chi(\mathscr{G}).$$

To take the limit inside χ, we need to prove that χ is continuous in a suitable sense. Actually, $\mathcal{G} \mapsto \chi(\mathcal{G})$ is continuous on sets of fixed support, and this is what is required to conclude.

Indeed, for any sequence $(\mathscr{G}_i)$ with fixed support for n large enough, there is a fixed label map f (depending on the support) such that for n large enough $\chi(\mathscr{G}_i) = f(\mathscr{G}_i)$, and the result follows from the fact that $\mathscr{G} \mapsto f(\mathscr{G})$ is continuous; since $f(\mathscr{G})(A) = \mathscr{G}(f^{-1}(A))$.

Consequently

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{n}\,\chi\left(\boxplus_{i=1}^{n} G_i\right) &= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/_{\equiv}} \lim_{n\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\big(G_i \in \overline{\mathscr{G}}\big)\right)\overline{\mathscr{G}}\right) \\
&= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/_{\equiv}} \mathbb{G}(\overline{\mathscr{G}})\overline{\mathscr{G}}\right) = \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/_{\equiv}} \mathbb{G}(\overline{\mathscr{G}})\chi(\overline{\mathscr{G}})\right) \\
&= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/_{\equiv}} \boxplus_{\mathscr{G}\in\overline{\mathscr{G}}} \mathbb{G}(\mathscr{G})\chi(\mathscr{G})\right) = \chi\left(\boxplus_{\mathscr{G}\in\mathfrak{G}} \mathbb{G}(\mathscr{G})\chi(\mathscr{G})\right) \\
&= \chi\left(\boxplus_{\mathscr{G}\in\mathfrak{G}} \mathbb{G}(\mathscr{G})\mathscr{G}\right) = \oint \mathscr{G}\,\mathrm{d}\mathbb{G}(\mathscr{G}). \qquad \square
\end{aligned}$$

1.B Language produced by a toric grammar

In this appendix, we make a deterministic study of the language produced by a toric grammar $\mathscr{G} \in \beta^*(\mathfrak{T})$. More precisely, we are interested in the support of the distribution $\mathbb{T}_{\mathscr{G}}$ of the final state of the production process.

Lemma 1.B.1
Let $\mathscr{T} \in \mathfrak{T}$ be some text and $\mathscr{G} \in \beta^*(\mathscr{T})$ be some grammar obtained by splitting this text a finite number of times. The number of splits performed can be read in $\mathscr{G}$ and is equal to

$$n = \sum_{i=1}^{+\infty} \mathscr{G}\Big(\rfloor_i S^*\Big).$$

Let us put $\overline{\alpha}(\mathscr{G}) = \alpha^n(\mathscr{G})$. Then, $\mathscr{T} \in \overline{\alpha}(\mathscr{G}) \subset \mathfrak{T}$, moreover $\overline{\alpha}(\mathscr{G}) = \mathrm{supp}(\mathbb{T}_{\mathscr{G}})$.

Proof. The grammar $\mathscr{G}$ is obtained by making a succession of splits. Each of those splits add one $\lfloor_i$ and one $\rfloor_i$ to the grammar, whereas in the original text there are no $\lfloor_i$ nor $\rfloor_i$, except for the $\lfloor_0$ at the beginning of each sentence. Since application of an element of $\mathfrak{F}$ does not change the number of such symbols, they may be used to count the number of splits performed.

Let us take then a sequence of toric grammars $\mathscr{T} = \mathscr{G}_0, \dots, \mathscr{G}_n = \mathscr{G}$, such that $\mathscr{G}_k \in \beta(\mathscr{G}_{k-1})$. From lemma 1.3.2 on page 28, there is a sequence $f_1, \dots, f_n \in \mathfrak{F}$ such that $f_k\Big(\mathscr{G}_{k-1}\Big) \in \alpha\Big(\mathscr{G}_k\Big)$. Let us prove by induction that for any $k = 0, \dots, n$,

$$f_k \circ \cdots \circ f_1\Big(\mathscr{T}\Big) \in \alpha^k\Big(\mathscr{G}_k\Big).$$

Indeed, this is true for $k = 0$, since $\mathscr{G}_0 = \mathscr{T}$. Moreover, assuming that the assertion holds for $k-1$, we deduce that

$$f_k \circ \cdots \circ f_1\Big(\mathscr{T}\Big) \in f_k\Big(\alpha^{k-1}\Big(\mathscr{G}_{k-1}\Big)\Big) \subset \alpha^{k-1}\Big(f_k\Big(\mathscr{G}_{k-1}\Big)\Big) \subset \alpha^k\Big(\mathscr{G}_k\Big).$$

showing that if the assertion holds for k, it also holds for $k+1$. For $k = n$, we obtain that

$$f_n \circ \cdots \circ f_1\Big(\mathscr{T}\Big) \in \alpha^n\Big(\mathscr{G}_n\Big).$$

As $f_n \circ \cdots \circ f_1(\mathscr{T}) = \mathscr{T}$, since $\mathscr{T}$ is a text, and $\mathscr{G}_n = \mathscr{G}$, we get that $\mathscr{T} \in \alpha^n(\mathscr{G})$.

Let us consider now $\mathscr{G}' \in \overline{\alpha}(\mathscr{G})$. Let $(\mathscr{G} = \mathscr{G}_0, \dots, \mathscr{G}_n = \mathscr{G}')$ the chain of grammars leading to $\mathscr{G}'$. Then for any $k = 0, \dots, n$,

$$\sum_{i=1}^{+\infty} \mathscr{G}_k\Big(\rfloor_i S^*\Big) = n - k,$$

since $\mathscr{G}_k \in \alpha(\mathscr{G}_{k-1})$ and each merge takes away one $\lfloor_i$ and one $\rfloor_i$. This implies that $\sum_{i=1}^{+\infty} \mathscr{G}'\Big(\rfloor_i S^*\Big) = 0$, and thus $\mathscr{G}' \in \mathfrak{T}$.

Note that, as remarked above, repeated merges may create elements of the type $\lfloor_i a \rfloor_i b$. However, this will not happen if n successful merges can be performed. Indeed in the case when expressions of the form $\lfloor_i a \rfloor_i b$ remain unmatched during the merge process, we will get $\alpha(\mathscr{G}_k) = \varnothing$ for some $k < n$. □

Chapter 2

Markov substitute Sets

2.1 Presentation of the model

2.1.1 Motivation

In the previous chapter, we constructed a split and merge process that rearranged texts according to a known reference grammar. We saw that when this reference grammar is known, the distribution of texts can be estimated without error via a Monte-Carlo simulation.

The question of building such a reference grammar $\mathscr{R}$ along justified statistical principles was left open. The model of toric grammars, as we saw, is based on the assumption that any elements $[_i a$ and $[_i b$ in the grammar can be substituted in any sentence, while keeping grammaticality. The goal of this chapter is to propose some ways to define properly this kind of property, and to test it.

2.1.2 What is a Markov substitute set

We will work for a while with a probability distribution of sentences using the dictionnary of words D, leaving the notion of text aside. We will use the notation $D^* = \bigcup_{j=0}^{\infty} D^j = \{\epsilon\} \cup D^+$, where ϵ stands for the empty string. Let $S \in D^+$ be a random sentence drawn from the language distribution and S_i, $1 \leqslant i \leqslant n$, a statistical sample made of n independent copies of S, our observed sample.

Given a right and left context $x = (x_1, x_2) \in D^* \times D^*$, and an expression $y \in D^+$ (that is any non void finite string of words), it will be useful to introduce the insertion operator α as

$$\alpha(x, y) = x_1 \, y \, x_2 \in D^+,$$

the concatenation of the strings of words x_1, y, and x_2.

Let us remark that for any $x \in (D^*)^2$, the map $y \mapsto \alpha(x, y)$ is one to one, whereas the map $x \mapsto \alpha(x, y)$ is not, since for instance $\alpha((y, y'), y) = \alpha((\varnothing, yy'), y)$.

Definition 2.1.1
We will say that the set $B \subset D^+$ is a Markov substitute set of S when

$$\sum_{x\in(D^*)^2} \mathbb{P}\Big(S = \alpha(x,y)\Big) > 0, \qquad y \in B, \tag{2.1.1}$$

and when there exists a probability measure $q_B \in \mathscr{M}_+^1(B)$ on B, called the substitute measure of B, such that for any $x \in (D^*)^2$ and any $y \in B$,

$$\mathbb{P}_S[\alpha(x,y)] = \mathbb{P}_S[\alpha(x,B)]q_B(y), \tag{2.1.2}$$

where $\alpha(x,B) = \{\alpha(x,y), y \in B\}$.

For any context $x \in (D^*)^2$, such that $\mathbb{P}_S[\alpha(x,B)] > 0$, the substitute measure is equal to

$$q_B(y) = \frac{\mathbb{P}_S[\alpha(x,y)]}{\mathbb{P}_S[\alpha(x,B)]}.$$

Therefore $\mathrm{supp}(q_B) = B$. *An equivalent characterization of the Markov substitute property is that B is a Markov substitute set if and only if it satisfies eq.* (2.1.1) *on this page and for any $y, y' \in B$, any $x, x' \in (D^*)^2$,*

$$\mathbb{P}_S[\alpha(x,y)]\ \mathbb{P}_S[\alpha(x',y')] = \mathbb{P}_S[\alpha(x,y')]\ \mathbb{P}_S[\alpha(x',y)]. \tag{2.1.3}$$

To justify calling this a Markov substitute set, we can make a link with Markov chains. Indeed, in the case when $S = (Z_1, \ldots, Z_L)$ is a finite length time homogeneous Markov chain, where $L > 2$ is a non random integer, and where $\mathrm{supp}\Big(\mathbb{P}_{Z_1}\Big) = D$, for any $w_1, w_3 \in D$, the set

$$B_{w_1,w_3} = \Big\{(w_1,w_2,w_3) \in D^3, \mathbb{P}(Z_2 = w_2, Z_3 = w_3 | Z_1 = w_1) > 0\Big\},$$

is a Markov substitute set. In the same way, let us consider a time homogeneous Markov chain $(Z_n, n \in \mathbb{N} \setminus \{0\})$, a subset $C \subset D$ and let us take for its stopping time $\tau = \inf\{k > 1, Z_k \in C\}$. Let us assume that $\mathbb{P}(\tau < \infty | Z_1 = z) = 1$ for any $z \in D$. In this situation, the chain $S = Z_{1:\tau} \overset{\text{def}}{=} (Z_1, \ldots, Z_\tau)$, stopped at time τ, is a process in D^+. For any $w, w' \in D$, such that

$$\sum_{1\leqslant j<k} \mathbb{P}\Big(Z_j = w, Z_k = w', j \leqslant \tau\Big) > 0$$

the set

$$B_{w,w'} = \Big\{w_{1:k} \in D^k, k > 1, w_1 = w, w_k = w', w_j \in D \setminus C, 1 < j < k, \\ \mathbb{P}_{Z_{2:k}\,|\,Z_1=w_1}(w_{2:k}) > 0\Big\}$$

is a Markov substitute set.

Indeed, in this case, there are $j < k \leqslant t$ such that

$$\mathbb{P}\Big(Z_j = w, Z_k = w', \tau = t\Big) > 0,$$

so that there is a chain $y_{1:t} \in D^t$ such that $y_{2:t-1} \in (D \setminus C)^{t-2}$, $y_j = w$, $y_k = w'$, $y_t \in C$ and

$$\mathbb{P}\Big(Z_{1:\tau} = y_{1:t}\Big) > 0.$$

Since

$$\begin{aligned}\mathbb{P}\Big(Z_{1:\tau} = y_{1:t}\Big) &= \mathbb{P}\Big(Z_{1:t} = y_{1:t}\Big)\\ &= \mathbb{P}\Big(Z_{1:j} = y_{1:j}\Big)\mathbb{P}\Big(Z_{2:k-j+1} = y_{j+1:k} \,|\, Z_1 = w\Big)\mathbb{P}\Big(Z_{2:t-k+1} = y_{k+1:t} \,|\, Z_1 = w'\Big),\end{aligned}$$

this implies that $y_{j:k} \in B_{w,w'}$, and that for any $w_{1:\ell} \in B_{w,w'}$,

$$\begin{aligned}&\mathbb{P}\Big[S = \alpha\Big((y_{1:j-1}, y_{k+1,t}), w_{1,\ell}\Big)\Big]\\ &\quad = \mathbb{P}\Big(Z_{1:j} = y_{1:j}\Big)\mathbb{P}\Big(Z_{2:\ell} = w_{2:\ell} | Z_1 = w\Big)\mathbb{P}\Big(Z_{2:t-k+1} = y_{k+1:t} | Z_1 = w'\Big) > 0,\end{aligned}$$

so that the set $B_{w,w'}$ satisfies eq. (2.1.1) on the preceding page.

Moreover for any $(x_{1:j}, x_{j+1:m}) \in (D^*)^2$,

$$\begin{aligned}\mathbb{P}_S\Big(\alpha(x, w)\Big) = \mathbb{P}\Big(Z_{1:j+1} = \gamma(x_{1:j}, w), \tau > j\Big)&\mathbb{P}\Big(Z_{2:k} = w_{2:\ell} \,|\, Z_1 = w, \tau \geqslant \ell\Big)\\ &\times \mathbb{P}\Big(Z_{2,m-j+1} = x_{j+1,m} \,|\, Z_1 = w', \tau \geqslant m - j + 1\Big).\end{aligned}$$

(where γ is the concatenation operator) so that $B_{w,w'}$ satisfies also eq. (2.1.2) on the facing page, where

$$q_{B_{w,w'}}(w_{1:\ell}) = \frac{\mathbb{P}\Big(Z_{2,k} = w_{2,\ell} \,|\, Z_1 = w, \tau \geqslant \ell\Big)}{\displaystyle\sum_{y \in (D\setminus C)^*} \mathbb{P}\Big(Z_{2:\ell(y)+1} = \gamma(w, y) \,|\, Z_1 = w\Big)}.$$

We gave these examples to show the connections with the usual Markov property and to provide an example of Markov substitute sets containing strings of variable lengths. We will see other examples of Markov substitute models a little later.

2.1.3 Weak Markov substitute sets

In natural language analysis, we may be mainly interested in the support of the distribution of S. If this is the case, we can weaken our definition of Markov substitute set to deal only with support shapes.

Definition 2.1.2
We will say that the set $B \subset D^+$ is a weak Markov substitute set of S when it satisfies eq. (2.1.1) on page 64 and when for any context $x \in (D^*)^2$ such that $\mathbb{P}_S[\alpha(x,B)] > 0$, $\mathbb{P}_S[\alpha(x,y)] > 0$ for any $y \in B$.

2.1.4 Basic properties of Markov substitute sets

Proposition 2.1.1

Any one point set $\{y\}$, where $y \in D^+$ satisfies eq. (2.1.1) on page 64, is a Markov substitute set, whose substitute measure is the Dirac mass at y.

A subset of a Markov substitute set is itself a Markov substitute set.

If B and C are Markov substitute sets such that $B \cap C \neq \varnothing$, then $B \cup C$ is also a Markov substitute set.

As such, a set B is a Markov substitute set if and only if, for any $y, y' \in B$, the pair $\{y, y'\}$ is a Markov substitute pair.

Less restrictively, a set B is a Markov substitute set if and only if there is a connected undirected spanning graph $\mathscr{G} \subset B^2$ such that for any $(y, y') \in \mathscr{G}$, $\{y, y'\}$ is a Markov substitute pair.

This is obvious from the characterization by eq. (2.1.2) on page 64 for the first point and eq. (2.1.3) on page 64 for the others.

These properties lead us to define the relation

$$y \sim_S y' \iff \{y, y'\} \text{ is a Markov substitute pair.}$$

This is an equivalence relation and $D^+/\!\sim_S$ forms a partition of D^+ into maximal Markov substitute sets.

Proposition 2.1.2
If B is a Markov substitute set, then for any $x \in (D^*)^2$, $\alpha(x,B)$ is also a Markov substitute set, as soon as it satisfies

$$\sum_{z \in (D^*)^2} \mathbb{P}\Big[S \in \alpha\Big(z, \alpha(x,B)\Big)\Big] > 0.$$

PROOF. From the definition of the Markov substitute property, we see that for any $z \in (D^*)^2$, any $y \in B$,

$$\mathbb{P}_S\Big[\alpha(z, \alpha(x,y)\Big] = \mathbb{P}_S\Big[\alpha(z_1x_1, x_2z_2, y)\Big] = \mathbb{P}_S\Big[\alpha\Big(z, \alpha(x,B)\Big)\Big] q_B(y),$$

so that $\alpha(x,B)$ is a Markov substitute set and $q_{\alpha(x,B)}\Big(\alpha(x,y)\Big) = q_B(y)$. □

Proposition 2.1.3
If B_j, $1 \leqslant j \leqslant k$ are Markov substitute sets (including possibly some one point sets), then

$$B = \gamma\left(B_1, \dots, B_k\right) \stackrel{\text{def}}{=} \Big\{ y = y_1 \dots y_k, y_j \in B_j, 1 \leqslant j \leqslant \ell \Big\},$$

is also a Markov substitute set, as soon as it satisfies

$$\sum_{x \in (D^*)^2} \mathbb{P}\Big(\alpha(x, B)\Big) > 0.$$

Proof. For any $y_{1:k}$, $y'_{1:k} \in \gamma\left(B_1, \dots, B_k\right)$,

$$\Big\{ y'_{1:j-1} y_{j:k}, y'_{1:j} y_{j+1:k} \Big\} = \alpha\Big(y'_{1:j-1}, y_{j+1,k}, \{y_j, y'_j\}\Big)$$

and is therefore a Markov substitute pair, so that $y_{1:k}$ and $y'_{1:k}$ are connected by the equivalence relation $\sim_S$ defined above and themselves form a Markov substitute pair. □

The same properties are true for weak Markov substitute sets. They show the key role played by Markov substitute pairs.

2.1.5 Interpretation in terms of random parsing

Let us consider some finite subset $B \in D^+$ satisfying eq. (2.1.1) on page 64. We define the set of B splits of s, for any sentence $s \in D^+$ as

$$\mathscr{S}(s, B) = \Big\{ (x, y), x \in (D^*)^2, y \in B, \alpha(x, y) = s \Big\}.$$

Let us consider some conditional probability kernel

$$\Big(\pi(s, x, y), s \in D^+, x \in (D^*)^2, y \in (B \cup \{\epsilon\})\Big)$$

such that $\pi(s, \cdot) \in \mathscr{M}_+^1\Big((D^*)^2 \times D^*\Big)$, and such that

$$\mathscr{S}(s, B) \subset \operatorname{supp}\Big(\pi(s, \cdot)\Big) \subset \mathscr{S}(s, B) \cup \Big\{ \Big((s, \epsilon), \epsilon\Big) \Big\}. \tag{2.1.4}$$

Introducing the split counts

$$c(s) = \Big| \mathscr{S}(s, B) \Big|,$$

we can for instance take

$$\pi(s, x, y) = \begin{cases} c(s)^{-1}, & (x, y) \in \mathscr{S}(s, B), \\ 1 - \displaystyle\sum_{(x', y') \in \mathscr{S}(s, B)} \pi(s, x', y'), & x = (s, \epsilon), y = \epsilon. \end{cases}$$

Let us define the random B-parse X, Y of the random sentence S on the same probability space by its conditional distribution

$$\mathbb{P}_{X,Y|S=s}(x,y) = \begin{cases} \min_{y' \in B} \pi\Big(\alpha(x,y'),x,y'\Big), & (x,y) \in \mathscr{S}(s,B), \\ 1 - \sum_{(x',y') \in \mathscr{S}(s,B)} \mathbb{P}_{X,Y|S=s}(x',y'), & x = (s,\epsilon), y = \epsilon. \end{cases}$$

Let us remark that the random B-parse can be simulated using a rejection method. Indeed, let us put

$$w(x,y) = \min_{y' \in B} \frac{\pi\Big(\alpha(x,y'),x,y'\Big)}{\pi\Big(\alpha(x,y),x,y\Big)}.$$

We can first draw the pair (X', Y') according to the conditional probability distribution $\mathbb{P}_{X',Y'|S} = \pi(S,\cdot)$, and then draw (X,Y) according to the distribution

$$\mathbb{P}_{(X,Y)|S,X',Y'} = w(X',Y')\delta_{(X',Y')} + \Big(1 - w(X',Y')\Big)\delta_{((S,\epsilon),\epsilon)}.$$

Proposition 2.1.4
The set B is a Markov substitute set for S if and only if the random B-parse (X,Y) satisfies one of the three following conditions :

$$\mathbb{P}_{X,Y|Y \in B} = \mathbb{P}_{X|Y \in B} \otimes \mathbb{P}_{Y|Y \in B}, \tag{2.1.5}$$

$$\mathbb{P}_{X|Y=y} = \mathbb{P}_{X|Y=y'}, \qquad y, y' \in B, \tag{2.1.6}$$

$$\mathbb{P}_{Y|X=x,Y \in B} = \mathbb{P}_{Y|Y \in B}, \qquad x \in (D^*)^2,\ \mathbb{P}_{X|Y \in B}(x) > 0. \tag{2.1.7}$$

Moreover, when B is a Markov substitute set, the substitute measure q_B is equal to $\mathbb{P}_{Y|Y \in B}$.

PROOF. The three properties are quite obviously equivalent, and the fact that, for any $x \in (D^*)^2$ and any $y \in B$,

$$\begin{aligned} \mathbb{P}_{X,Y}(x,y) &= \min_{y' \in B} \pi\Big(\alpha(x,y'),x,y'\Big)\mathbb{P}_S[\alpha(x,y)] \\ &= \min_{y' \in B} \pi\Big(\alpha(x,y'),x,y'\Big)\mathbb{P}_S[\alpha(x,B)]q_B(y), \end{aligned}$$

ends the proof. □

2.2 Invariant dynamics

Markov substitute sets can be used to define various reversible (and consequently invariant) dynamics.

Moreover, when $\mathbb{P}_S$ has a sufficient number of Markov substitute sets, we will be able to define an irreducible (in other words ergodic) dynamics that will have a single invariant measure.

The idea is to replace an element from a Markov substitute set by another one. Such a transformation stays necessarily in the support of the language, and, if the substitution weights are properly related to the substitute measure, this dynamics can be invariant, and even reversible. We will see here simple invariant dynamics, both on single sentences and on texts, that will only modify one element of a Markov substitute set at a time. More evolved invariant dynamics will be defined and studied in chapter 3.

2.2.1 Metropolis invariant dynamics on sentences

We will here have a first go at describing invariant dynamics on sentences, related to the Markov substitute property.

Let us consider a family $B_j, 1 \leqslant j \leqslant t$ of Markov substitute sets, and let us assume that we know for each B_i the substitute measure $q_{\overline{B}_i}$ of a Markov substitute set $\overline{B}_i$ containing B_i. We do not need necessarily to know q_{B_i} itself, since we can deduce it from the relation $q_{B_i}(y) = q_{\overline{B}_i}(y)/q_{\overline{B}_i}(B_i)\mathbb{1}(y \in B_i)$.

We will locate a member of a Markov substitute set in the sentence and replace it by another element of the same Markov substitute set, according to the relevant substitute measure. Let us remark that we accordingly will only use the substitute measures $q_{\overline{B}_i}$ such that

$$\sum_{x \in (D^*)^2} \mathbb{E}\Big(\pi(S, x, \overline{B}_i, i)\Big) > 0.$$

Definition 2.2.1
A splitting kernel is a kernel

$$\pi : D^+ \to \mathscr{M}_+^1\Big((D^*)^2 \times D^* \times \{1, \ldots, t\}\Big)$$

such that for any $s \in D^+$,

$$\pi(s, x, y, j) > 0 \implies y \in B_j \cup \{\epsilon\} \text{ (the empty string), and } \alpha(x, y) = s.$$

For now, we will suppose that we have access to such kernels. Actual constructions will be presented in chapter 3.

Definition 2.2.2
Given a splitting kernel π, we can define the substitute dynamics

$$k_\pi(s,s') = \sum_{x\in(D^*)^2,(y,y')\in(D^*)^2,j} \pi(s,x,y,j)\; q_{\overline{B}_j}(y') \left(\frac{\pi(s',x,y',j)}{\pi(s,x,y,j)} \wedge 1\right), \quad s \neq s',$$
$$k_\pi(s,s) = 1 - \sum_{s'\in D^+\setminus\{s\}} k_\pi(s,s'). \tag{2.2.1}$$

The simulation of S' such that $\mathbb{P}_{S'|S=s}(s') = k_\pi(s,s')$ can be implemented in the following way. First draw (X,Y,j) such that $\mathbb{P}_{X,Y,j|S} = \pi$, then draw Y' according to $q_{\overline{B}_j}$ and set

$$S' = \begin{cases} \alpha(X,Y') & \text{with probability } \left(\dfrac{\pi\Big(\alpha(X,Y'),X,Y',j\Big)}{\pi(S,X,Y,j)} \wedge 1\right), \\ S & \text{otherwise.} \end{cases}$$

Proposition 2.2.1
Any substitute dynamics defined as in definition 2.2.2 on this page is reversible (and therefore invariant) with respect to $\mathbb{P}_S$.

PROOF. We can see that k_π is reversible by writing

$$\mathbb{P}_S(s)k(s,s') = \sum_{x\in(D^*)^2,(y,y')\in(D^*)^2,j} \mathbb{P}_S\Big(\alpha(x,\overline{B}_j)\Big) \times q_{\overline{B}_j}(y) q_{\overline{B}_j}(y') \Big[\pi(s,x,y,j) \wedge \pi(s',x,y',j)\Big],$$

which is obviously a symmetric expression in s and s'. □

Remark that, while all these operations are defined on single sentences, it is quite straightforward to extend them to texts, that is, an empirical distribution on sentences, by applying them to each sentence of the text.

2.2.2 Reflecting a reversible dynamics on the boundary of a finite domain

We can reflect a $\mathbb{P}_S$ reversible dynamics k on the boundary of any finite domain $\mathscr{D} \subset D^+$, by defining the reflected dynamics as

$$k_{\mathscr{D}}(s,s') = \begin{cases} k(s,s'), & \text{when } s,s' \in \mathscr{D},\ s \neq s', \\ 0, & \text{when } s \in \mathscr{D},\ s' \notin \mathscr{D}, \\ k(s,s) + \displaystyle\sum_{s''\in D^+\setminus\mathscr{D}} k(s,s''), & \text{when } s = s' \in \mathscr{D}, \\ \delta_s(s'), & s \notin \mathscr{D}. \end{cases}$$

It is immediate to see that $k_{\mathscr{D}}$ is still reversible with respect to $\mathbb{P}_S$.

2.2.3 Compound dynamics

Definition 2.2.3
Let us consider a finite family of $\mathbb{P}_S$ reversible dynamics, $k_1, \ldots, k_t$ and a probability distribution $\xi \in \mathscr{M}_+^1\big(\{1, \ldots, t\}\big)$ on the t first integers. We define the compound substitute dynamics k_ξ as the reversible kernel

$$k_\xi(s, s') = \sum_{j=1}^{t} \xi(j) k_j(s, s').$$

This compound kernel can be easily implemented by drawing at random an index j according to the distribution ξ and then drawing s' according to $k_{B_j}(s, s')$. It is obviously reversible with respect to $\mathbb{P}_S$.

Definition 2.2.4
We can then define the accelerated compound dynamics

$$\overline{k}_\xi(s, s') = \left(\sum_{s'' \in D^+ \setminus \{s\}} k_\xi(s, s'') \vee \sum_{s'' \in D^+ \setminus \{s'\}} k_\xi(s', s'') \right)^{-1} k_\xi(s, s'), \qquad s \neq s' \in D^+,$$
$$\overline{k}_\xi(s, s) = 1 - \sum_{s' \in D^+ \setminus \{s\}} \overline{k}_\xi(s, s'), \qquad s \in D^+.$$

Let us remark that for states $s \in D^+$ such that

$$\sum_{s'' \in D^+ \setminus \{s\}} k_\xi(s, s'') = \max_{s' \in D^+, k(s,s') > 0} \sum_{s'' \in D^+ \setminus \{s'\}} k_\xi(s', s''),$$

the accelerated dynamics jumps with probability one, meaning that $\overline{k}_\xi(s, s) = 0$.

2.2.4 A simple example of recursive structure

In this subsection, we will present a small example, showing that the Markov substitute property is suitable to model recursive structures.

Let us consider on $D = \{a, b\}$ the language distribution

$$\mathbb{P}_S(ab^n) = \left(\frac{1}{2}\right)^{n+1}.$$

It is easy to check that $\{a, ab\}$ is a Markov substitute set for this language, with

$$q_B = \frac{2}{3}\delta_a + \frac{1}{3}\delta_{ab}.$$

A random B-parse as defined in section 2.1.5 on page 67 can be defined in this case as

$$\mathbb{P}(X = (\epsilon, b^n), Y = a | S = ab^n) = \mathbb{P}(X = (\epsilon, b^{n-1}), Y = ab | S = ab^n) = \frac{1}{2}\cdot$$

The invariant dynamics in this setup is to either remove one b with probability $\frac{1}{3}$ or add one with probability $\frac{1}{6}$, as shown in fig. 2.1 on this page.

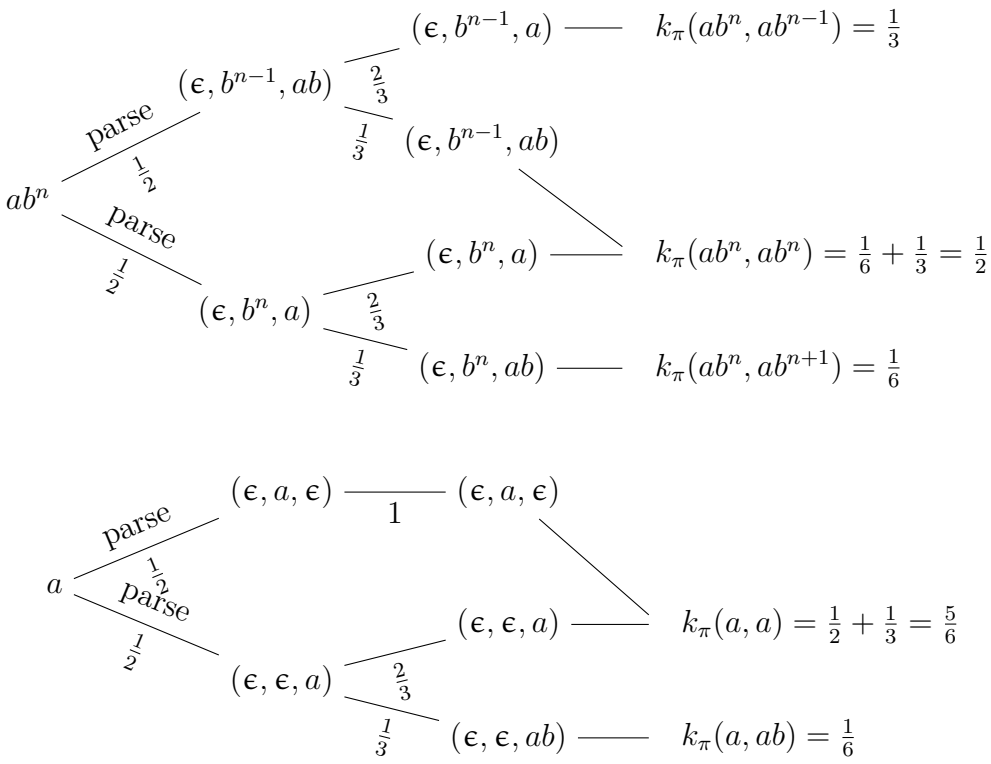


Figure 2.1: Invariant dynamics for ab^n.

It is quite straightforward to see that this dynamics is indeed invariant.

It is important to realize on this basic example that substitute dynamics have the power to generate languages with a recursive structure.

Here, the full language $ab^n, n \in \mathbb{N}$ is generated from any starting point ab^k, because the invariant dynamics is ergodic (since it has a single communicating class). A statistical sample larger than a single sentence would still be necessary to discover that $\{a, ab\}$ is a Markov substitute pair.

On the other hand, we can define invariant dynamics implementing any set of rewriting rules, generating languages with arbitrarily complex recursive structures.

2.2.5 Crossing-over reversible dynamics on texts

The previous method required that we know the substitute measures $q_{\overline{B}_i}$, which must be estimated. We will now present another way to build an invariant dynamics, on texts this time, that does not require to know the substitute measures. We will here see texts as a concatenation of symbols, supposing that each sentence ends with a dedicated symbol (a period, if you prefer), so that the sentences of a text are uniquely defined. This dynamics will simply swap two elements of the same Markov substitute set.

Accordingly, we introduce $T = \gamma(S_1, \ldots, S_n) \in D^+$, obtained by concatenating the sentences of our statistical sample.

To describe crossing-over dynamics, it will be useful to generalize the merge operation to multiple contexts. A two-fold context x being an element of $(D^*)^3$, we define

$$\alpha\Big((x_1, x_2, x_3), y\Big) = (x_1 y x_2, x_3).$$

Quite obviously, all formulas on α may be generalized on two-fold contexts, in particular the definition of Markov substitute sets

$$\mathbb{P}_S\Big(\alpha\Big(\alpha(x, y), z\Big)\Big) = \mathbb{P}_S\Big(\alpha\Big(\alpha(x, B), z\Big)\Big) q_B(y).$$

Definition 2.2.5
A double splitting kernel on a text is an kernel

$$\pi : D^+ \to \mathscr{M}_+^1\Big(\Big((D^*)^3\Big) \times D^* \times D^* \times \{1, \ldots, t\}\Big)$$

such that for any $t \in D^+$,

$$\pi(t, x, y_1, y_2, j) > 0 \implies y_1, y_2 \in B_j \cup \{\epsilon\}, x \in (D^*)^3, \alpha(\alpha(x, y_1), y_2) = t.$$

Definition 2.2.6
Given a double splitting kernel π, we can define the crossing-over dynamics on texts

$$\sigma_\pi(t, t') = \sum_{x \in (D^*)^3, (y_1, y_2) \in (D^*)^2, j} \pi(t, x, y_1, y_2, j) \times \left(\frac{\pi(t', x, y_2, y_1, j)}{\pi(t, x, y_1, y_2, j)} \wedge 1 \right), \qquad t \neq t' \tag{2.2.2}$$

$$\sigma_\pi(t, t) = 1 - \sum_{t' \in D^+ \setminus \{t\}} \sigma_\pi(t, t'). \tag{2.2.3}$$

The simulation of T' such that $\mathbb{P}_{T'|T=t}(t') = \sigma_\pi(t,t')$ can be implemented in the following way. First draw $P = (X, Y_1, Y_2, j)$ such that $\mathbb{P}_{P|T} = \pi$, then set

$$T' = \begin{cases} \alpha\Big(\alpha(X,Y_2),Y_1\Big), & \text{w. p. } \left(\dfrac{\pi(T',X,Y_2,Y_1,j,n_1,n_2)}{\pi(T,X,Y_1,Y_2,j,n_1,n_2)} \wedge 1\right), \\ T & \text{otherwise.} \end{cases}$$

Proposition 2.2.2
Any crossing-over dynamics defined as in definition 2.2.6 on the previous page is reversible with respect to $\mathbb{P}_T = (\mathbb{P}_S)^{\otimes n}$.

PROOF. It is obvious that $\sigma_\pi(t,t') = \sigma_\pi(t',t)$ for any texts t, t', so we only have to prove that $\mathbb{P}_T(t) = \mathbb{P}_T(t')$ as soon as $\sigma_\pi(t,t') > 0$.

In this case, there is a configuration $(x, y_1, y_2, j) \in (D^*)^3 \times B_j^2 \times \{j\}$ such that $t = \alpha\Big(\alpha(x,y_1),y_2\Big)$ and $t' = \alpha\Big(\alpha(x,y_2),y_1\Big)$. We can then remark that

$$\begin{aligned}
\mathbb{P}(T=t) &= q_{B_j}(y_2) \sum_{y \in B_j} \mathbb{P}\Big[T = \alpha\Big(\alpha(x,y_1),y\Big)\Big] \\
&= q_{B_j}(y_1)\, q_{B_j}(y_2) \sum_{y,y' \in B_j} \mathbb{P}\Big[\alpha\Big(\alpha(x,y),y'\Big)\Big] \\
&= q_{B_j}(y_1) \sum_{y \in B_j} \mathbb{P}\Big[T = \alpha\Big(\alpha(x,y_2),y\Big)\Big] \\
&= \mathbb{P}\Big[T = \alpha\Big(\alpha(x,y_2),y_1\Big)\Big] \\
&= \mathbb{P}(T=t').
\end{aligned}$$

□

2.3 Exponential families of Markov substitute processes

As there may be compatibility issues, knowing whether there exists a language distribution with a given family of Markov substitute sets is not obvious. We are going in this section to give a theoretical answer to this question, by which we mean a non constructive description of the Markov substitute processes having a prescribed set of Markov substitute sets. We will show more precisely that the language probability distributions having a prescribed finite collection of Markov substitute sets form a union of exponential families (depending on their support), and that these exponential families are non empty, meaning that the compatibility issue between Markov substitute sets never results in a void model.

Definition 2.3.1
For any given family $\mathscr{B}$ of subsets of D^+, we will say that the random process $S \in D^+$ is a $\mathscr{B}$-Markov substitute process if all the members $B \in \mathscr{B}$ are Markov substitute sets for $\mathbb{P}_S$.

The purpose of this section is to describe the set of $\mathscr{B}$-Markov substitute processes.

We will first introduce the special subfamily of independent Markov substitute processes, that will be useful to show that the set of $\mathscr{B}$-Markov substitute processes is not empty.

Given a strict sub-probability measure $\xi \in \mathscr{M}_+(D)$, let us put

$$r = 1 - \sum_{w \in D} \xi(w) > 0,$$

and let us define the independent process $\widetilde{S}_\xi$ by its distribution

$$\mathbb{P}(\widetilde{S}_\xi = w_{1:k}) = \frac{r}{1-r} \prod_{j=1}^{k} \xi(w_j).$$

Let us remark that it is such that $\mathbb{P}\Big[\ell(\widetilde{S}) = L\Big] = r(1-r)^{L-1}$. It is easy to see from the definition that $\mathrm{supp}(\xi)^+$ is a Markov substitute set for the independent model $\widetilde{S}$ and that its substitute measure is equal to the distribution of $\widetilde{S}_\xi$ itself.

We will start with a description of the possible supports of $\mathscr{B}$-Markov substitute processes.

To do so, let us introduce the equivalence relation $\sim_{\mathscr{B}}$ defined as

$$\begin{aligned} y \sim_{\mathscr{B}} y' \iff & \exists (x_j, y_j, y'_j), x_j \in (D^*)^2, y_j, y'_j \in B_j \in \mathscr{B}, 0 \leqslant j \leqslant J, \\ & y = \alpha(x_0, y_0), y' = \alpha(x_J, y'_J), \alpha(x_{j-1}, y'_{j-1}) = \alpha(x_j, y_j), 0 < j \leqslant J. \end{aligned} \quad (2.3.1)$$

Let us remark that any component of ${}^{D^+}/_{\sim_{\mathscr{B}}}$ is a Markov substitute set for any $\mathscr{B}$-Markov substitute process, due to the basic properties of Markov substitute sets.

For any $C \in {}^{D^+}/_{\sim_{\mathscr{B}}}$, let us define the subset $\mathscr{B}_C$ of $\mathscr{B}$ present in C as

$$\mathscr{B}_C = \Big\{ B \in \mathscr{B}, \sum_{x \in (D^*)^2} \mathbb{1}\Big(\alpha(x, B) \cap C \neq \varnothing\Big) > 0 \Big\}.$$

Proposition 2.3.1
For any family $\mathscr{B}$ of subsets of D^+, for any $\mathscr{B}$-Markov substitute process S, there is a subset $\mathscr{C}_S \subset {}^{D^+}/_{\sim_{\mathscr{B}}}$ such that the support $\mathrm{supp}\Big(\mathbb{P}_S\Big)$ is of the form

$$\mathrm{supp}\Big(\mathbb{P}_S\Big) = \bigcup_{C \in \mathscr{C}_S} C \quad (2.3.2)$$

and such that

$$\mathscr{B} = \bigcup_{C \in \mathscr{C}_S} \mathscr{B}_C. \tag{2.3.3}$$

Conversely, if $\mathscr{C} \subset {}^{D^+}/_{\sim_{\mathscr{B}}}$ *is such that*

$$\mathscr{B} = \bigcup_{C \in \mathscr{C}} \mathscr{B}_C, \tag{2.3.4}$$

given any strict sub-probability measure $\xi \in \mathscr{M}_+(D)$*, such that* $\operatorname{supp}(\xi) = D$*, and any probability measure* $\mu \in \mathscr{M}_+^1(\mathscr{C})$*, such that* $\operatorname{supp}(\mu) = \mathscr{C}$*, the process* S *defined as*

$$\mathbb{P}_S = \sum_{C \in \mathscr{C}} \mu(C)\, \mathbb{P}_{\widetilde{S}_\xi | \widetilde{S}_\xi \in C} \tag{2.3.5}$$

is a $\mathscr{B}$*-Markov substitute process such that*

$$\operatorname{supp}\Big(\mathbb{P}_S\Big) = \bigcup_{C \in \mathscr{C}} C. \tag{2.3.6}$$

PROOF. Any component $C \in {}^{D^+}/_{\sim_{\mathscr{B}}}$ such that $\operatorname{supp}(\mathbb{P}_S) \cap C \neq \varnothing$ is a Markov substitute set. Consequently, $C \subset \operatorname{supp}(\mathbb{P}_S)$, since by considering an empty context x in eq. (2.1.2) on page 64,

$$\mathbb{P}(S = s) = \mathbb{P}(C)\, q_C(s) > 0, \qquad s \in C.$$

Moreover, condition eq. (2.3.3) on the current page is required to ensure that all members of $\mathscr{B}$ satisfies eq. (2.1.1) on page 64.

The support of $\widetilde{S}_\xi$ being D^+, it is clear that the support of S is as described in eq. (2.3.6) on the current page. This implies that any member $B \in \mathscr{B}$ satisfies eq. (2.1.1) on page 64.

Let us consider now any $B \in \mathscr{B}$, any $x \in (D^*)^2$, and any $y \in B$. There is $C \in {}^{D^+}/_{\sim_{\mathscr{B}}}$ such that $\alpha(x, y) \in C$. Consequently, $\alpha(x, B) \in C$, since all members of this set are connected as described in eq. (2.3.1) on the previous page. We can write

$$\begin{aligned}
\mathbb{P}_S\Big(\alpha(x,y)\Big) &= \frac{\mu(C)\, \mathbb{P}\Big(\widetilde{S}_\xi = \alpha(x,y)\Big)}{\mathbb{P}(\widetilde{S}_\xi \in C)} \\
&= \frac{\mu(C)\, \mathbb{P}\Big(\widetilde{S}_\xi = \alpha(x,B)\Big)\mathbb{P}\Big(\widetilde{S}_\xi = y \,|\, \widetilde{S}_\xi \in B\Big)}{\mathbb{P}\Big(\widetilde{S}_\xi \in C\Big)} \\
&= \mathbb{P}_S\Big(\alpha(x,B)\Big)\, \mathbb{P}\Big(\widetilde{S}_\xi = y \,|\, \widetilde{S}_\xi \in B\Big),
\end{aligned}$$

proving that B satisfies eq. (2.1.2) on page 64, with $q_B = \mathbb{P}_{\widetilde{S}_\xi \,|\, \widetilde{S}_\xi \in B}$. Consequently, S is a $\mathscr{B}$-Markov process as claimed in the proposition. In this proof, we have used the fact that the Markov substitute property is stable by conditioning.□

We are now going to show that the $\mathscr{B}$-Markov processes form an exponential family, although we will not provide an efficient algorithm to compute the corresponding energy function (or in other terms sufficient statistics). In this respect, the method can be considered as non constructive.

Let $\mathscr{B}$ be any finite family of finite subsets of D^+.

The $\mathscr{B}$-Markov substitute processes remain the same if we replace $\mathscr{B}$ by any other family of sets such that the equivalence relation $\sim_{\mathscr{B}}$ remains unchanged. Indeed, the fact that S is a $\mathscr{B}$-Markov substitute process is equivalent to the fact that any subset of $D^+/\sim_{\mathscr{B}}$ is a Markov substitute set, since it implies it and the reverse implication is trivial (since each $B \in \mathscr{B}$ is included in one of the components of $D^+/\sim_{\mathscr{B}}$).

This property allows to propose some reorganization of $\mathscr{B}$. First of all, we can split any $B \in \mathscr{B}$ into pairs. Namely, if $B = \{y_j, 1 \leqslant j \leqslant k\}$, we can replace B with the $k-1$ sets $\{y_1, y_j\}$, $1 < j \leqslant k$.

After this transformation, $\mathscr{B}$, being a finite set of pairs, can be considered as an undirected graph on D^+ (with a finite number of edges). We can then remove successively any pair included in a cycle, and prune in this way $\mathscr{B}$ until we obtain an acyclic graph (this will leave the connected components of $\mathscr{B}$, and therefore also $\sim_{\mathscr{B}}$, unchanged). We can go further and remove more pairs as long as this removal does not change $\sim_{\mathscr{B}}$, until we obtain a minimal set of pairs. However, this last removal operation is not constructive since it is not clear that an efficient algorithm could check the required property.

Let us now decompose the distribution of any $\mathscr{B}$-Markov substitute process into

$$\mathbb{P}_S(s) = \sum_{C \in D^+/\sim_{\mathscr{B}}} \mathbb{P}(S \in C)\, \mathbb{P}(S = s | S \in C).$$

Let us choose in each $C \in D^+/\sim_{\mathscr{B}}$ a reference point $s_C \in C$, and let us write explicitly

$$\mathscr{B} = \Big\{\{y_{i,0}, y_{i,1}\}, 1 \leqslant i \leqslant I\Big\},$$

where we have chosen an explicit order or indexation for each pair of $\mathscr{B}$.

Let us choose for any $s \in C \in D^+/\sim_{\mathscr{B}}$, such that $\mathbb{P}(C) > 0$ a path

$$\Big((x_j, i_j, \sigma_j), 0 \leqslant j \leqslant J, x_j \in (D^*)^2, 1 \leqslant i_j \leqslant I, \sigma_j \in \{0,1\}\Big) \tag{2.3.7}$$

such that

$$\begin{aligned} \alpha(x_1, y_{i_0,\sigma_0}) &= s_C, \\ \alpha(x_J, y_{i_J,1-\sigma_J}) &= s, \\ \alpha(x_{j-1}, y_{i_{j-1},1-\sigma_{j-1}}) &= \alpha(x_j, y_{i_j,\sigma_j}), \end{aligned} \tag{2.3.8}$$

and let us put

$$\beta_i = \log\left(\frac{q_{\{y_{i,0},y_{i,1}\}}(y_{i,1})}{q_{\{y_{i,0},y_{i,1}\}}(y_{i,0})}\right), \qquad 1 \leqslant i \leqslant I,$$
$$a_{s,i} = \sum_{j=0}^{J} \mathbb{1}(i_j = i)\Big(\mathbb{1}(\sigma_j = 0) - \mathbb{1}(\sigma_j = 1)\Big), \qquad s \in C, 1 \leqslant i \leqslant I,$$
$$b_s = \log\left(\frac{\mathbb{P}(S = s)}{\mathbb{P}(S = s_C)}\right).$$

The vector b_s defines $\mathbb{P}_{S|S \in C}$, according to the relation

$$\mathbb{P}(S = s|S \in C) = \frac{\exp(b_s)}{\sum_{s' \in C} \exp(b_{s'})}.$$

Moreover, due to the Markov substitute property,

$$b_s = \sum_{i=1}^{I} a_{s,i}\beta_i, \tag{2.3.9}$$

since

$$\frac{\mathbb{P}_S(s)}{\mathbb{P}_S(s_C)} = \prod_{j=0}^{J} \frac{q_{\{y_{j_i,0},y_{j_i,1}\}}(y_{j_i,1-\sigma_j})}{q_{\{y_{j_i,0},y_{j_i,1}\}}(y_{j_i,\sigma_j})}. \tag{2.3.10}$$

Let us remark now that S is a $\mathscr{B}$-Markov substitute process, if and only if

$$\mathscr{B} = \bigcup_{C \in D^+/\sim_{\mathscr{B}}, \mathbb{P}(S \in C) > 0} \mathscr{B}_C, \tag{2.3.11}$$

and eq. (2.3.10) on the current page holds not only for the chosen path, but for any path connecting s_C and s as prescribed above, for some set of substitute measures q_B, $B \in \mathscr{B}$, with full support $\mathrm{supp}(q_B) = B$. Indeed the first condition eq. (2.3.11) on this page is required by eq. (2.1.1) on page 64 and the second condition is obviously necessary too. The two conditions are also sufficient. Indeed, in this case, for any $y \in B = \{y_{i,0}, y_{i,1}\} \in \mathscr{B}$, there is C such that $\mathbb{P}(C) > 0$ and there is $x \in (D^*)^+$ such that $\alpha(x, B) \cap C \neq 0$. Therefore, $\alpha(x, B) \subset C$, because obviously $\alpha(x, y_{i,0}) \sim_{\mathscr{B}} \alpha(x, y_{i,1})$. According to eq. (2.3.10) on this page, $\mathbb{P}(S = s) > 0$ for any $s \in C$, so that $\mathbb{P}\Big(S = \alpha(x, y_{i,0})\Big) > 0$ and $\mathbb{P}\Big(S = \alpha(x, y_{i,1})\Big) > 0$, implying that $\mathbb{P}\Big(S = \alpha(x, y)\Big) > 0$, and therefore that eq. (2.1.1) on page 64 is satisfied. Moreover, given $x \in (D^*)^2$ and $i, 1 \leqslant i \leqslant I$, such that $\mathbb{P}\Big(S = \alpha(x, \{y_{i,0}, y_{i,1}\})\Big) > 0$, considering the component $C \in {}^{D^+}/_{\sim_{\mathscr{B}}}$ such that $\alpha(x, y_{i,0}) \in C$, this implies that

$\alpha(x, y_{i,1}) \in C$ also, and that we can connect s_C with $\alpha(x, y_{i,1})$ adding to the path chosen to connect s_C with $\alpha(x, y_{i,0})$ the step $(x, i, 0)$ at the end of the path. Applying eq. (2.3.10) on the facing page to these two paths proves that

$$\frac{\mathbb{P}_S\Big(\alpha(x, y_{i,1})\Big)}{\mathbb{P}_S\Big(\alpha(x, y_{i,0})\Big)} = \frac{q_{\{y_{i,0}, y_{i,1}\}}(y_{i,1})}{q_{\{y_{i,0}, y_{i,1}\}}(y_{i,0})},$$

and therefore that S is a $\mathscr{B}$-Markov process.

Now we can remark that eq. (2.3.10) on the preceding page will hold for any path, and not only the chosen one, if and only if, for any loop $\ell \in \mathscr{L}$ connecting some s_C, such that $\mathbb{P}(S \in C) > 0$ to itself, that is satisfying eq. (2.3.8) on page 77, with $s = s_C$, the corresponding vector of weights

$$c_{\ell,i} = \sum_{j=0}^{J} \mathbb{1}(i_j = i)\Big(\mathbb{1}(\sigma_j = 0) - \mathbb{1}(\sigma_j = 1)\Big), \qquad 1 \leqslant i \leqslant I,$$

is such that

$$\sum_{i=1}^{I} c_{\ell,i}\, \beta_i = 0. \tag{2.3.12}$$

Let us remark that $\mathscr{L}$ is countable but may be infinite. By reindexing $\mathscr{B}_C$ if necessary, we may assume that the vectors $(c_{\cdot,i}, d < i \leqslant I)$ forms a basis of the vector space generated by the vectors $c_{\cdot,i}$, $1 \leqslant i \leqslant I$, where d is some integer. We can therefore find a real matrix $(e_{j,i}, d < j \leqslant I, 1 \leqslant i \leqslant I)$ such that

$$c_{\ell,i} = \sum_{j=d+1}^{I} c_{\ell,j}\, e_{j,i}, \qquad \ell \in \mathscr{L}, \quad 1 \leqslant i \leqslant I.$$

Let us remark that the submatrix $(e_{j,i}, d < j \leqslant I, d < i \leqslant I)$ is equal to the identity matrix. Substituting in eq. (2.3.12) on the current page, we get

$$\sum_{j=d+1}^{I} c_{\ell,j} \sum_{i=1}^{I} e_{j,i}\, \beta_i = 0.$$

Since by construction $c_{\cdot,i}$, $d < j \leqslant I$ are independent, eq. (2.3.12) on this page is equivalent to

$$\sum_{i=1}^{I} e_{j,i}\, \beta_i = 0, \qquad d < j \leqslant I,$$

which can be rewritten according to the fact that $(e_{j,i}, d < j \leqslant I, d < i \leqslant I)$ is the identity, as

$$\beta_j = -\sum_{i=1}^{d} e_{j,i}\, \beta_i, \qquad d < j \leqslant I.$$

Plugging this condition into eq. (2.3.9) on page 78, we get that S is a $\mathscr{B}$-Markov process if and only if

$$b_s = -\sum_{i=1}^{d} \beta_i \, U_i(s), \qquad s \in \mathrm{supp}(\mathbb{P}_S),$$

where

$$U_i(s) = -\left(a_{s,i} - \sum_{j=d+1}^{I} a_{s,j} \, e_{j,i} \right), \qquad s \in \mathrm{supp}(\mathbb{P}_S), \ 1 \leqslant i \leqslant d.$$

Let us insist on the fact that the energy $U_i(s)$ depends only on $\mathrm{supp}(\mathbb{P}_S)$, and not on the coefficients β_i that defines the substitute measures of the members of $\mathscr{B}$ under $\mathbb{P}_S$.

We conclude that the process S is a $\mathscr{B}$-Markov substitute process if and only if its support satisfies eq. (2.3.11) on page 78 and there is a vector of real coefficients ($\beta_i \in \mathbb{R}, 1 \leqslant i \leqslant d$) such that

$$\mathbb{P}(S = s) = \sum_{C \in D^+/\sim_{\mathscr{B}}} \mathbb{1}(s \in C) \mathbb{P}(S \in C) \frac{\exp\left(-\sum_{i=1}^{d} \beta_i \, U_i(s) \right)}{\sum_{s' \in C} \exp\left(-\sum_{i=1}^{d} \beta_i \, U_i(s') \right)},$$

where the energy function $U_i(s)$, depends only on $\mathrm{supp}(\mathbb{P}_S)$. This can be written as an exponential family on $\mathrm{supp}(\mathbb{P}_S)$. More precisely, for any $\mathscr{C} \in D^+/\sim_{\mathscr{B}}$ such that

$$\mathscr{B} = \bigcup_{C \in \mathscr{C}} \mathscr{B}_C,$$

the set of $\mathscr{B}$-Markov processes whose support is $\bigcup \mathscr{C}$ is the exponential family

$$\left\{ \left(\frac{\exp\left(-\sum_{C \in \mathscr{C} \setminus \{C_0\}} \beta_C \, \mathbb{1}(s \in C) - \sum_{i=1}^{d} \beta_i \, U_i(s) \right)}{\sum_{C \in \mathscr{C}} \sum_{s' \in C} \exp\left(-\sum_{C \in \mathscr{C} \setminus \{C_0\}} \beta_C \, \mathbb{1}(s' \in C) - \sum_{i=1}^{d} \beta_i \, U_i(s') \right)}, s \in \bigcup \mathscr{C} \right), \beta_C, \beta_i \in \mathbb{R} \right\},$$

where β_C and β_i should also be such that the denominator is not infinite.

Moreover, this exponential family contains the independent Markov substitute processes described previously, so that necessarily $d \geqslant 1$, as soon as $\mathscr{B}$ is not trivial, that is contains at least a member that is not a one point set.

Let us summarize what we have proved in the following proposition.

Proposition 2.3.2
Given any finite set $\mathscr{B}$ of finite subsets of D^+, there is a finite set of pairs, $\mathscr{P}$, that we can choose such that each one is included in a member of $\mathscr{B}$, such that the sets of $\mathscr{B}$-Markov and $\mathscr{P}$-Markov substitute processes are the same, and such that $\mathscr{P}$ is minimal for the inclusion relation (removing a pair from $\mathscr{P}$ would break the above property).

For any $\mathscr{B}$-Markov substitute process S, we can decompose its support as a union of components $\mathscr{C} \subset {}^{D^+}/_{\sim_{\mathscr{B}}}$, such that $\mathrm{supp}(\mathbb{P}_S) = \bigcup \mathscr{C}$ *and such that*

$$\mathscr{B} = \bigcup_{C \in \mathscr{C}} \mathscr{B}_C.$$

For any such subset, there is a free, non empty subset of pairs $\mathscr{F} \subset \mathscr{P}$, a matrix $\Big(e_{j,i}, j \in \mathscr{P} \setminus \mathscr{F}, i \in \mathscr{F}\Big)$, an index set $I(\mathscr{C})$ containing $\mathscr{F}$ and an energy function $\Big(U_i(s), i \in I(\mathscr{C}), s \in C\Big)$, such that the set of $\mathscr{B}$-Markov processes whose support is $\bigcup \mathscr{C}$ is the linear exponential family

$$\mathfrak{M}_{\mathscr{C}}(\mathscr{B}) = \left\{ \left(p(s) = \frac{\exp\Big(- \sum_{i \in I(\mathscr{C})} \beta_i \, U_i(s) \Big)}{\sum_{s' \in \bigcup \mathscr{C}} \exp\Big(- \sum_{i \in I(\mathscr{C})} \beta_i \, U_i(s') \Big)}, s \in \bigcup \mathscr{C} \right), \beta \in \mathfrak{B} \subset \mathbb{R}^{I(\mathscr{C})} \right\},$$

where

$$\mathfrak{B} = \left\{ \beta \in \mathbb{R}^{I(\mathscr{C})}, \sum_{s \in \bigcup \mathscr{C}} \exp\Big(- \sum_{i \in I(\mathscr{C})} \beta_i U_i(s) \Big) < \infty \right\},$$

and such that for any member p of this family, the substitute measure of any pair $i = \{y_{i,0}, y'_{i,1}\} \in \mathscr{F}$ (taken in a suitable order compatible with the definition of U), is given by

$$q_i(y_{i,\sigma}) = \frac{\exp(\sigma \beta_i)}{1 + \exp(\beta_i)}, \qquad \sigma \in \{0, 1\},$$

whereas for $j \in \mathscr{P} \setminus \mathscr{F}$,

$$q_j(y_{i,\sigma}) = \frac{\exp(\sigma \beta_j)}{1 + \exp(\beta_j)}, \qquad \sigma \in \{0, 1\},$$

where

$$\beta_j = \sum_{i \in \mathscr{F}(C)} e_{j,i} \, \beta_i.$$

This result implies that the maximum likelihood estimator in $\mathfrak{M}_{\mathscr{C}}(\mathscr{B})$ provides an asymptotically efficient estimator of the parameters β_i, $i \in I(\mathscr{C})$. We will describe in section 3.3 on page 135 exponential families of Markov substitute processes restricted to finite subdomains $\mathscr{D}$ of D^+ and provide a way to compute with the help of a Monte-Carlo simulation the maximum likelihood estimator in this finite setting (where $\mathfrak{B} = \mathbb{R}^{I(\mathscr{C})}$).

Let us give an example showing that it is possible to have $\mathscr{P} \neq \mathscr{F}$ and that $\mathscr{F}$ and therefore the form of the energy may depend on $\mathscr{C}$.

Let us consider the family $\mathscr{B} = \{B_1, B_2\}$, where $B_1 = \{ab, a\}$, $B_2 = \{bc, c\}$, and where $a, b, c \in D$.

Let us consider

$$\begin{aligned} C_1 &= \{ab^n c, n \in \mathbb{N}\}, \\ C_2 &= \{b^m cab^n, m, n \in \mathbb{N}\}, \\ C_3 &= \{b^k cab^m cab^n, k, m, n \in \mathbb{N}\}. \end{aligned}$$

It is easy to check that $C_1, C_2, C_3 \in {}^{D^+}/_{\sim_{\mathscr{B}}}$, and that $\{C_1\}$, $\{C_2\}$ and $\{C_3\}$ are all valid choices for $\mathscr{C}$, since both B_1 and B_2 are present in those three components. On C_1, we have the loop

$$(a)c \to (ab)c \to a(bc) \to a(c),$$

and the induced constraint that $q_{B_1}(ab) = q_{B_2}(bc)$, so that

$$\mathfrak{M}_{\{C_1\}} = \left\{ \Big(p(ab^n c) = r(1-r)^n \Big), r \in]0,1[\right\}.$$

Moreover we can take $\mathscr{F}(\{C_1\}) = \{B_1\}$, and although $\mathscr{B}_{C_1} = \mathscr{B}$, we could remove either B_1 or B_2 from $\mathscr{B}$ while maintaining the fact that $C_1 \in {}^{D^+}/_{\sim_{\mathscr{B}}}$ and without changing $\mathfrak{M}_{\{C_1\}}$.

On C_2, we have no non trivial loop, so that $\mathscr{F}(\{C_2\}) = \mathscr{B}$ and

$$\mathfrak{M}_{\{C_2\}} = \left\{ \left(p\Big(b^m cab^n\Big) = r_1 r_2 (1-r_1)^m (1-r_2)^m \right), r_1, r_2 \in]0,1[\right\}.$$

On C_3, we have the same kind of non trivial loop as on C_1, imposing the constraint $q_{B_1} = q_{B_2}$, so that

$$\mathfrak{M}_{\{C_3\}} = \left\{ \left(p\Big(b^k cab^m cab^n\Big) = r(1-r)^{k+m+n}, k, m, n \in \mathbb{N} \right), r \in]0,1[\right\},$$

but this time, removing B_1 or B_2 from $\mathscr{B}$ would disconnect the component C_3 that would no longer belong to ${}^{D^+}/_{\sim_{\mathscr{B}}}$.

On $\{C_1, C_2\}$, the situation is again different

$$\mathfrak{M}_{\{C_1,C_2\}} = \Big\{ \Big(p\Big(ab^n c\Big) = r_1 r_2 (1-r_2)^n, \\ p\Big(b^m c a b^n\Big) = (1-r_1) r_2 (1-r_2)^{m+n}, m, n \in \mathbb{N} \Big), r_1, r_2 \in]0,1[\Big\}.$$

It is interesting to remark that the trace of $\mathfrak{M}_{\{C_1,C_2\}}$ on C_2 (the set of conditional probabilities on C_2 of distributions in $\mathfrak{M}_{\{C_1,C_2\}}$), is not equal to $\mathfrak{M}_{\{C_2\}}$: the presence of C_1 in the support imposes further constraints on the dependence between substitute measures.

2.4 Testing Markov substitute sets

We will discuss in this section the question of building statistical tests to decide whether sets of expressions are Markov substitute sets.

Let us consider a family of sets of contexts $\Theta \subset 2^{(D^*)^2}$, and assume that it contains all single contexts $\{x\}$, $x \in (D^*)^2$.

Let us assume that we already know that B_1 and B_2 are two Markov substitute sets. If they have a non void intersection, their union is also a Markov substitute set. However, this may be hard to check, so we assume that we do not know whether they are disjoint or not, and want to test whether their union $B = B_1 \cup B_2$ is a Markov substitute set.

To perform this test, we assume that we can check efficiently whether a given $y \in D^+$ belongs to B_1 or to B_2, using a parsing algorithm. We consider the random variables

$$F_{B_1,B_2,\theta}\Big(X_B, Y_B, p\Big) = \Big[\mathbb{1}(Y_B \in B_1) - p\, \mathbb{1}(Y_B \in B)\Big] \mathbb{1}(X_B \in \theta), \\ \theta \in \Theta, p \in [0,1], \quad (2.4.1)$$

where (X_B, Y_B) is the B-parse defined in section 2.1.5 on page 67. It is easy to deduce from the properties stated in this section that B is a Markov substitute set if and only if, there is $p \in [0,1]$ such that for any $\theta \in \Theta$,

$$\mathbb{E}\Big[F_{B_1,B_2,\theta}\Big(X_B, Y_B, p\Big)\Big] = 0.$$

If we do not want to actually simulate the parse process (X_B, Y_B), we can work with $\mathbb{E}\Big[F_{B_1,B_2,\theta}(X_B, Y_B, p)\Big|S\Big]$ instead of working with $F_{B_1,B_2,\theta}(X_B, Y_B, p)$ itself. In

other words, we may consider

$$F_{B_1,B_2,\theta}\big(S,p\big) = \mathbb{E}\Big[\Big(\mathbb{1}(Y_B \in B_1) - p\,\mathbb{1}(Y_B \in B)\Big)\mathbb{1}(X_B \in \theta)\Big|S\Big]$$
$$= \sum_{x\in(D^*)^2}\sum_{y\in B_1\cup B_2} \mathbb{1}(x\in\theta)\min_{y'\in B_1\cup B_2}\pi\big(\alpha(x,y'),x,y'\big)$$
$$\times\Big[\mathbb{1}\big(y\in B_1\big)(1-p) - \mathbb{1}\big(y\in B_2\big)p\Big]\mathbb{1}\big(S=\alpha(x,y)\big). \quad (2.4.2)$$

Using this last formula, we can see that

$$\mathbb{E}\Big[F_{B_1,B_2,\theta}(S,p)\Big] = \sum_{x\in\theta}\min_{y'\in B_1\cup B_2}\pi\big(\alpha(x,y'),x,y'\big)$$
$$\times\Big[\mathbb{P}_S\big(\alpha(x,B_1)\big)(1-p) - p\,\mathbb{P}_S\big(\alpha(x,B_2)\big)\Big].$$

2.4.1 Alternative construction of a test function

For any $\theta\in\Theta$, let us consider some probability measure $\nu_\theta\in\mathscr{M}_+^1((D^*)^2)$, such that $\theta\subset\mathrm{supp}(\nu_\theta)$ (in the case when θ is a finite set, we can for instance consider the uniform measure on θ).

Let X_θ be some random variable distributed according to ν_θ and independent of S. Since $y\mapsto\alpha(x,y)$ is one to one, we can define a random variable $Y_\theta\in D^*$ by the property $\alpha(X_\theta,Y_\theta)=S$ when $S\in\alpha(X_\theta,D^+)$, and $Y_\theta=\epsilon$ otherwise.

Let us define the test functions

$$F_{B_1,B_2,\theta}(S,p) = \sum_{x\in(D^*)^2}\sum_{y\in D^+}\nu_\theta(x)\mathbb{1}\big(x\in\theta\big)\mathbb{1}\big(S=\alpha(x,y)\big)$$
$$\times\Big[\mathbb{1}(y\in B_1) - p\mathbb{1}\big(y\in B_1\cup B_2\big)\Big]$$
$$= \mathbb{E}\Big[\Big(\mathbb{1}\big(Y_\theta\in B_1\big) - p\mathbb{1}\big(Y_\theta\in B_1\cup B_2\big)\Big)\mathbb{1}\big(X_\theta\in\theta\big)\,\Big|\,S\Big]. \quad (2.4.3)$$

We see on this expression that this definition of the test function may not be very efficient when θ is large, because of the factor $\nu_\theta(x)$, but that it will not be affected by the size of $B_1\cup B_2$, that is not even required to be a finite set, whereas the definition given in eq. (2.4.2) on the current page may be inefficient when $B_1\cup B_2$ is large, and does not allow $B_1\cup B_2$ to contain an infinite number of elements.

Given that B_1 and B_2 are two Markov substitute sets, it is easy to see that $B=B_1\cup B_2$ is a Markov substitute set if and only if there is $p\in[0,1]$ such that for any $\theta\in\Theta$,

$$\mathbb{E}\Big[F_{B_1,B_2,\theta}(S,p)\Big] = 0.$$

Indeed, we can write this expectation as

$$\sum_{x\in\theta}\nu_\theta(x)\Big[\mathbb{P}_S\big(\alpha(x,B_1)\big)(1-p) - p\,\mathbb{P}_S\big(\alpha(x,B_2)\big)\Big].$$

If we are ready to loose the interpretation of $F_{B_1,B_2,\theta}(S,p)$ in terms of the random variables X_θ and Y_θ, while retaining the property that $|F_{B_1,B_2,\theta}(S,p)| \leqslant 1$, we can take for some real parameter $\alpha > 1$,

$$\nu_\theta(x) = \nu(x) = \nu(x_1, x_2) = (\alpha - 1)^2 \Big(\ell(x_1) + 2\Big)^{-\alpha} \Big(\ell(x_2) + 2\Big)^{-\alpha},$$

that is such that for any $s \in D^+$,

$$\sum_{x \in (D^*)^2} \sum_{y \in D^+} \nu(x) \mathbb{1}\Big(s = \alpha(x,y)\Big) \leqslant 1.$$

We can also build tests that are more closely related to the invariant dynamics introduced in section 2.2.1 on page 69. Let us assume that B_1 and B_2 are disjoint Markov substitute sets. Let π be a conditional probability kernel, satisfying eq. (2.1.4) on page 67, for $B = B_1 \cup B_2$.

Consider the family of test functions

$$\begin{aligned} F_{B_1,B_2,\theta}(S,p) = \sum_{x \in (D^*)^2} \sum_{y_1 \in B_1, y_2 \in B_2} & \mathbb{1}(x \in \theta) \Big[\pi\Big(\alpha(x,y_1), x, y_1\Big) \wedge \pi\Big(\alpha(x,y_2), x, y_2\Big)\Big] \\ & \times \Big[\mathbb{1}\Big(S = \alpha(x,y_1)\Big) q_{B_2}(y_2)(1-p) - \mathbb{1}\Big(S = \alpha(x,y_2)\Big) q_{B_1}(y_1) p\Big]. \quad (2.4.4) \end{aligned}$$

The advantage, when compared to eq. (2.4.2) on the facing page or eq. (2.4.3) on the preceding page is that we take the infimum on two values of the kernel π only, and that the coefficient $\nu_\theta(x)$ is not present. The drawback is that we have to know, or at least to estimate the substitute measures q_{B_1} and q_{B_2}.

Proposition 2.4.1
Let us assume that B_1 and B_2 are two Markov substitute sets. Let us also assume that for any $(x,y) \in (D^*)^2 \times (B_1 \cup B_2)$, $\pi\Big(\alpha(x,y), x, y\Big) > 0$. Then $B_1 \cup B_2$ is a Markov substitute set if and only if there is $p \in [0,1]$ such that for any $\theta \in \Theta$,

$$\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) = 0,$$

so that we can use the random variables $F_{B_1,B_2,\theta}(S,p)$ to define a test.

To see that the proposition is true, we can remark that

$$\begin{aligned} \mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) = \sum_{x \in (D^*)^2} \Bigg(& \sum_{(y_1,y_2) \in B_1 \times B_2} \mathbb{1}(x \in \theta) q_{B_1}(y_1) q_{B_2}(y_2) \\ & \times \Big[\pi\Big(\alpha(x,y_1), x, y_1\Big) \wedge \pi\Big(\alpha(x,y_2), x, y_2\Big)\Big] \Bigg) \\ & \times \Big[\mathbb{P}_S\Big(\alpha(x,B_1)\Big)(1-p) - \mathbb{P}_S\Big(\alpha(x,B_2)\Big) p\Big]. \end{aligned}$$

If $B_1 \cup B_2$ is a Markov substitute set, then

$$\mathbb{P}_S\Big(\alpha(x, B_1)\Big) = \mathbb{P}_S\Big(\alpha(x, B_1 \cup B_2)\Big) q_{B_1 \cup B_2}(B_1),$$
$$\mathbb{P}_S\Big(\alpha(x, B_2)\Big) = \mathbb{P}_S\Big(\alpha(x, B_1 \cup B_2)\Big) q_{B_1 \cup B_2}(B_2),$$

so that $\mathbb{E}\Big(F_{B_1,B_2,\theta}(S, q_{B_1 \cup B_2}(B_1)\Big) = 0$.

On the other hand, if for any $\theta \in \Theta$, $\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) = 0$, then for any context $x \in (D^*)^2$,

$$\mathbb{P}_S\Big(\alpha(x, B_1)\Big) - p\, \mathbb{P}_S\Big(\alpha(x, B_1 \cup B_2)\Big) = 0,$$

implying that $B_1 \cup B_2$ is a Markov substitute set with $q_{B_1 \cup B_2}(B_1) = p$.

To see how $F_{B_1,B_2,\theta}(S,p)$ can be computed by simulations, we can remark that

$$\begin{aligned} F_{B_1,B_2,\theta}(S,p) = \sum_{x \in (D^*)^2} \sum_{y,y' \in (D^*)^2} & \pi(S,x,y) \mathbb{1}(x \in \theta) \\ & \times \Big[\mathbb{1}(y \in B_1)(1-p) q_{B_2}(y') - \mathbb{1}(y \in B_2) p q_{B_1}(y')\Big] \\ & \times \left(\frac{\pi\Big(\alpha(x,y'),x,y'\Big)}{\pi\Big(\alpha(x,y),x,y\Big)} \wedge 1\right). \end{aligned}$$

Thus, if we draw

$$\mathbb{P}_{(X,Y)|S} = \pi$$

and

$$\mathbb{P}_{Y'|X,Y} = \mathbb{1}(Y \in B_1) q_{B_2} + \mathbb{1}(Y \in B_2) q_{B_1}$$

we obtain that

$$\begin{aligned} F_{B_1,B_2,\theta}(S,p) = \mathbb{E}\Bigg\{ & \mathbb{1}(X \in \theta)\, \Big[\mathbb{1}(Y \in B_1)(1-p) - \mathbb{1}(Y \in B_2) p\Big] \\ & \times \left(\frac{\pi\Big(\alpha(X,Y'),X,Y'\Big)}{\pi\Big(\alpha(X,Y),X,Y\Big)} \wedge 1\right) \;\Big|\; S\Bigg\}. \end{aligned}$$

Let us introduce the weights

$$w(x,y) = \mathbb{E}\left(\left(\frac{\pi\Big(\alpha(X,Y'),X,Y'\Big)}{\pi\Big(\alpha(X,Y),X,Y\Big)} \wedge 1\right)\Big| X = x, Y = y\right),$$

and the random variables (X'', Y'') defined as

$$\mathbb{P}_{X'',Y''|X,Y} = w(X,Y)\delta_{X,Y} + \Big(1 - w(X,Y)\Big)\delta_{\Big(\alpha(X,Y),\,\epsilon\Big),\,\epsilon}.$$

We can rewrite $F_{B_1,B_2,\theta}(S,p)$ as

$$F_{B_1,B_2,\theta}(S,p) = \mathbb{E}\Big\{\mathbb{1}(X'' \in \theta)\Big[\mathbb{1}\Big(Y'' \in B_1\Big) - p\mathbb{1}\Big(Y'' \in B_1 \cup B_2\Big)\Big]\Big|S\Big\}.$$

This shows that we could also simulate (X'', Y'') and base the tests on

$$F_{B_1,B_2,\theta}(X'', Y'', p) = \mathbb{1}(X'' \in \theta)\Big[\mathbb{1}\Big(Y'' \in B_1\Big) - p\,\mathbb{1}\Big(Y'' \in B_1 \cup B_2\Big)\Big]. \qquad (2.4.5)$$

In this way we fall back on the same setting as the previous one, defined by eq. (2.4.1) on page 83.

2.5 Weakening the Markov substitute assumption

Let us remark that if we do not want to assume that $\pi\Big(\alpha(x,y),x,y\Big) > 0$ for all $(x,y) \in (D^*)^2 \times D^*$, we will test a weaker property than the Markov substitute property, as stated in the following proposition.

Proposition 2.5.1
Let us assume that for any $x \in (D^*)^2$, there is $\theta \in \Theta$ such that $\theta = \{x\}$. Let us assume also that the dynamics

$$\begin{aligned} k(s,s') = & \sum_{x\in(D^*)^2,(y,y')\in(D^*)^2} \pi(s,x,y) \\ & \times \Big[\mathbb{1}(y \in B_1)q_{B_1}(y') + \mathbb{1}(y \in B_2)q_{B_2}(y')\Big]\left(\frac{\pi(s',x,y')}{\pi(s,x,y)} \wedge 1\right), \quad s \neq s', \\ k(s,s) = & 1 - \sum_{s'\neq s} k(s,s') \end{aligned}$$

is reversible with respect to $\mathbb{P}_S$. Under these assumptions, the dynamics

$$\begin{aligned} k'(s,s') = & \sum_{x\in(D^*)^2,(y,y')\in(D^*)^2} \pi(s,x,y) \\ & \times \Big[pq_{B_1}(y') + (1-p)q_{B_2}(y')\Big]\left(\frac{\pi(s',x,y')}{\pi(s,x,y)} \wedge 1\right), \quad s \neq s', \\ k'(s,s) = & 1 - \sum_{s'\neq s} k(s,s'). \end{aligned}$$

is reversible with respect to the same invariant probability measure if and only if, for all $\theta \in \Theta$, $\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) = 0$, where the test function is defined as in eq. (2.4.4) on page 85.

This comes from the fact that k is reversible if and only if for any any $i \in \{1,2\}$, any $x \in (D^*)^2, (y,y') \in B_i^2$ such that $\pi\Big(\alpha(x,y),x,y\Big) \wedge \pi\Big(\alpha(x,y'),x,y'\Big) > 0$,

$$\mathbb{P}\Big(S = \alpha(x,y)\Big) q_{B_i}(y') = \mathbb{P}\Big(S = \alpha(x,y')\Big) q_{B_i}(y).$$

Actually, weakening the hypothesis opens perspectives for a more refined modeling. We can for instance implement along this line the masking of some syntactic categories. As an example, in the idiom " a Trojan horse " you would like to prevent the possibility of replacing horse with another noun, whereas you would like to be able to do so in other contexts.

2.6 Testing the Markov substitute property using a parse process

In this section, we will propose a test for the Markov substitute property based on test functions of the type

$$F_\theta(X,Y,p) = \mathbb{1}(X \in \theta)\Big[\mathbb{1}(Y \in B_1) - p\,\mathbb{1}(Y \in B_1 \cup B_2)\Big],$$

where B_1 and B_2 are Markov substitute sets, and the test has to decide whether $B = B_1 \cup B_2$ is also a Markov substitute set. This covers two possible constructions of the parse process, defined either by eq. (2.4.1) on page 83, or by eq. (2.4.5) on the previous page. We make this study as an introduction to tests based on the other type of test functions, where a conditional expectation knowing S is taken. In particular, we will not address here the question of making the test uniform with respect to the range of values that the pair B_1, B_2 may take.

As already explained, in both cases, we want to test whether there is a value of $p \in [0,1]$ such that for any $\theta \in \Theta$, $\mathbb{E}\Big[F_\theta(X,Y,p)\Big] = 0$. Let us introduce the constants

$$p_+ = \sup\Big\{\mathbb{P}\Big(Y \in B_1 \,|\, Y \in B, X = x\Big); x \in (D^*)^2, \mathbb{P}(Y \in B, X = x) > 0\Big\},$$
$$p_- = \inf\Big\{\mathbb{P}\Big(Y \in B_1 \,|\, Y \in B, X = x\Big); x \in (D^*)^2, \mathbb{P}(Y \in B, X = x) > 0\Big\}.$$

Remark that with these notations, $\mathbb{E}\Big[F_\theta(X,Y,p)\Big] = 0$ when $p_+ = p_- = p$, which corresponds exactly to the Markov substitute property to be tested.

Since $p \mapsto F_\theta(X,Y,p)$ is non-increasing,

$$\mathbb{E}\Big[F_\theta(X,Y,p_+) \,|\, X,Y \in B\Big] \leqslant 0,$$
$$\mathbb{E}\Big[F_\theta(X,Y,p_-) \,|\, X,Y \in B\Big] \geqslant 0,$$

and, observing that F_θ is null when $\mathbb{1}(Y \in B) = 0$, we obtain that

$$\begin{aligned} \mathbb{E}\Big[F_\theta(X, Y, p_+) \,|\, X, \mathbb{1}(Y \in B)\Big] &\leqslant 0, \\ \mathbb{E}\Big[F_\theta(X, Y, p_-) \,|\, X, \mathbb{1}(Y \in B)\Big] &\geqslant 0. \end{aligned} \tag{2.6.1}$$

These inequalities will be used to get bounds for p_+ and p_-, based on concentration inequalities concerning the empirical mean of $F_\theta(X, Y, p)$.

The test will be based on a simulation (X_i, Y_i), $1 \leqslant i \leqslant n$ of the random parse process, where each pair (X_i, Y_i) is independently drawn from the conditional probability distribution $\mathbb{P}_{X,\,Y\,|\,S}$. Doing so, we obtain an i.i.d. sample (X_i, Y_i), $1 \leqslant i \leqslant n$.

2.6.1 Probability of false rejection

Proposition 2.6.1
Let $\mu \in \mathscr{M}_+^1$ be a probability measure depending only on $\Big(X_i, \mathbb{1}(Y_i \in B),\, i \in [\![1, n]\!]\Big)$. Let us define

$$F_{\theta,i}(p) = F_\theta(X_i, Y_i, p).$$

Let Λ be a finite subset of $]0, 1[$. With probability at least $1 - 2\varepsilon$,

$$B_-(p_+) \stackrel{\text{def}}{=} \sup_{\rho \in \mathscr{M}_+^1(\Theta), \lambda \in \Lambda} \int \sum_{i=1}^n \log\Big(1 + \lambda F_{\theta,i}(p_+)\Big) \mathrm{d}\rho(\theta) - \mathscr{K}(\rho, \mu) - \log\Big(|\Lambda|/\varepsilon\Big) \leqslant 0,$$

$$B_+(p_-) \stackrel{\text{def}}{=} \sup_{\rho \in \mathscr{M}_+^1(\Theta), \lambda \in \Lambda} \int \sum_{i=1}^n \log\Big(1 - \lambda F_{\theta,i}(p_-)\Big) \mathrm{d}\rho(\theta) - \mathscr{K}(\rho, \mu) - \log\Big(|\Lambda|/\varepsilon\Big) \leqslant 0,$$

where

$$\mathscr{K}(\rho, \mu) = \begin{cases} \displaystyle\int \log\Big(\frac{\mathrm{d}\rho}{\mathrm{d}\mu}\Big)\, \mathrm{d}\rho, & \text{when } \rho \ll \mu, \\ +\infty, & \text{otherwise.} \end{cases}$$

Therefore, if we reject the hypothesis that B is a Markov substitute set when

$$\inf_{p \in [0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0, \tag{2.6.2}$$

the probability of false rejection is at most 2ε.

Remark 2.6.1
We may also strengthen the Markov substitute property, requiring that the substitute measure is not too unbalanced. For any real parameter $\eta \in]0, 1[$, we will say that (B_1, B_2) is an η-Markov substitute pair of sets when $B = B_1 \cup B_2$ is a Markov substitute set such that $q_B(B_1) \in [\eta, 1 - \eta]$. According to the proposition, we can

reject the hypothesis that the pair (B_1, B_2) *is an* η*-Markov substitute pair of sets when*

$$\inf_{p\in[\eta,1-\eta]} \max\Big\{B_-(p), B_+(p)\Big\} > 0, \tag{2.6.3}$$

with a probability of false rejection not greater than 2ε*.*

PROOF. Let us prove the last statement of the proposition first, assuming that the rest is true. In the case when B is a Markov substitute set, $p_- = p_+$, so that with probability at least $1 - 2\varepsilon$,

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} \leqslant \max\Big\{B_-(p_+), B_+(p_+)\Big\} \leqslant 0.$$

Consequently

$$\mathbb{P}\bigg(\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0\bigg) \leqslant 2\varepsilon.$$

If moreover (B_1, B_2) is an η-Markov substitute pair of sets, then $p_+ \in [\eta, 1-\eta]$, so that with probability at least $1 - 2\varepsilon$,

$$\inf_{p\in[\eta,1-\eta]} \max\Big\{B_-(p), B_+(p)\Big\} \leqslant \max\Big\{B_-(p_+), B_+(p_+)\Big\} \leqslant 0,$$

proving that

$$\mathbb{P}\bigg(\inf_{p\in[\eta,1-\eta]} \max\Big\{B_-(p), B_+(p)\Big\} > 0\bigg) \leqslant 2\varepsilon.$$

Let us now proceed to the proof of the first statement of the proposition. Let us put

$$W_\theta(\lambda) = \sum_{i=1}^{n} \log\Big(1 + \lambda F_{\theta,i}(p_+)\Big).$$

This is legitimate, since $|F_{\theta,i}(p)| \leqslant 1$, and $\lambda < 1$. From eq. (2.6.1) on the preceding page, and the independence of $\Big(X_i, \mathbb{1}(Y_i \in B), F_{\theta,i}(p_+)\Big)$,$1 \leqslant i \leqslant n$,

$$\mathbb{E}\Big[\exp\Big(W_\theta(\lambda)\Big) \,\Big|\, X_i, \mathbb{1}(Y_i \in B), 1 \leqslant i \leqslant n\Big] = \prod_{i=1}^{n} \mathbb{E}\Big(1 + \lambda F_{\theta,i}(p_+) \,\Big|\, X_i, \mathbb{1}(Y_i \in B)\Big) \leqslant 1.$$

Since μ depends only on X_i, and $\mathbb{1}(Y_i \in B)$, $1 \leqslant i \leqslant n$, we get that

$$\mathbb{E}\bigg[\int \exp\Big(W_\theta(\lambda)\Big)\, \mathrm{d}\mu(\theta)\bigg] = \mathbb{E}\bigg[\int \mathbb{E}\Big[\exp\Big(W_\theta(\lambda)\Big) \,|\, X_i, \mathbb{1}(Y_i \in B), 1 \leqslant i \leqslant n\Big] \mathrm{d}\mu(\theta)\bigg] \leqslant 1.$$

Together with the remark that for any distribution $\rho \in \mathscr{M}_+^1(\Theta)$,

$$\log\Big(\int \exp(h)\,\mathrm{d}\mu\Big) \geqslant \int h\,\mathrm{d}\rho - \mathscr{K}(\rho,\mu), \tag{2.6.4}$$

we get finally that

$$\begin{aligned}\mathbb{E}\Big[\exp\Big(B_-(p_+)\Big)\Big] &= \mathbb{E}\Big[\exp\Big(\sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda}\int W_\theta(\lambda)\,\mathrm{d}\rho(\theta) - \mathscr{K}(\rho,\mu) - \log\Big(|\Lambda|/\varepsilon\Big)\Big)\Big]\\ &\leqslant \mathbb{E}\Big[\sum_{\lambda\in\Lambda}\int \exp\Big(W_\theta(\lambda) - \log\Big(|\Lambda|/\varepsilon\Big)\Big)\,\mathrm{d}\mu(\theta)\Big] \leqslant \varepsilon,\end{aligned}$$

proving through the Markov inequality that $\mathbb{P}\Big(B_-(p_+) \geqslant 0\Big) \leqslant \varepsilon$. The proof concerning $B_+(p_-)$ is the same, mutatis mutandi (changing λ for $-\lambda$ and $+$ for $-$).□

In practice, we will assume that

$$\sup\Big\{\ \Big|\Big\{\theta\in\Theta : x\in\theta\Big\}\Big| : x\in(D^*)^2, \alpha(x,B)\cap\mathrm{supp}\Big(\mathbb{P}_S\Big)\neq\varnothing\ \Big\} \leqslant m, \tag{2.6.5}$$

and choose for μ the uniform probability measure on

$$\widehat{\Theta} = \Big\{\theta\in\Theta, \{X_i, 1\leqslant i\leqslant n\}\cap\theta\neq\varnothing\Big\}, \tag{2.6.6}$$

a set of size at most mn. In this case $\mathscr{K}(\rho,\mu) \leqslant \log\Big(|\widehat{\Theta}|\Big) \leqslant \log(mn)$, the former inequality being an equality when ρ is a Dirac mass.

2.6.2 Probability of false acceptance of the hypothesis

We have built the test of eq. (2.6.2) on page 89 by controlling its probability of false rejection. We would like now to study its probability of false acceptance. To this purpose, let us establish lower bounds for $B_-(p_+)$ and $B_+(p_-)$. (We need indeed to prove that these quantities cannot be both negative when the hypothesis is false.)

Lemma 2.6.2
Let us define

$$p(\theta) = \mathbb{P}\Big(Y\in B_1 \,|\, X\in\theta, Y\in B\Big).$$

For any $p\in[0,1]$, any $\lambda\in]-1,1[$,

$$\begin{aligned}\mathbb{E}\Big[\exp\Big(-\log\Big[1+\lambda F_\theta(p)\Big]\Big)\Big] \leqslant \exp\Big[\mathbb{E}\Big[\mathbb{1}(X\in\theta, Y\in B)\Big]\Big[\lambda\Big(p-p(\theta)\Big)\\ + \frac{\lambda^2}{1-|\lambda|}\Big(p(\theta)\Big(1-p(\theta)\Big) + \Big(p-p(\theta)\Big)^2\Big)\Big]\Big].\end{aligned}$$

PROOF. Let us remark first that for $-1 < \lambda < 1, -1 \leqslant x \leqslant 1$,

$$-\log(1+\lambda x) = \log\Big(1 - \lambda x + \frac{\lambda^2 x^2}{1+\lambda x}\Big) \leqslant \log\Big(1 - \lambda x + \frac{\lambda^2 x^2}{1-|\lambda|}\Big).$$

As a consequence,

$$\mathbb{E}\bigg[\exp\Big(-\log\big[1+\lambda F_\theta(p)\big]\Big)\bigg] \leqslant 1 - \lambda\mathbb{E}\Big(F_\theta(p)\Big) + \frac{\lambda^2}{1-|\lambda|}\mathbb{E}\Big(F_\theta(p)^2\Big).$$

The rest follows easily from

$$\begin{aligned}
\mathbb{E}\Big(F_\theta(p)\Big) &= \mathbb{E}\Big[\Big(\mathbb{1}(Y \in B_1) - p\Big)\mathbb{1}(X \in \theta, Y \in B)\Big] \\
&= \Big(p(\theta) - p\Big)\mathbb{E}\big[\mathbb{1}(X \in \theta, Y \in B)\big], \\
\mathbb{E}\Big(F_\theta(p)^2\Big) &= \mathbb{E}\Big[\Big(\mathbb{1}(Y \in B_1) - p\Big)^2\mathbb{1}(X \in \theta, Y \in B)\Big] \\
&= \mathbb{E}\Big[\Big(\mathbb{1}(Y \in B_1)(1-2p) + p^2\Big)\mathbb{1}(X \in \theta, Y \in B)\Big] \\
&= \Big(p(\theta)(1-2p) + p^2\Big)\mathbb{P}\Big(X \in \theta, Y \in B\Big) \\
&= \Big[p(\theta)\Big(1 - p(\theta)\Big) + \Big(p - p(\theta)\Big)^2\Big]\mathbb{P}\Big(X \in \theta, Y \in B\Big). \qquad \square
\end{aligned}$$

Proposition 2.6.3
Let us assume that eq. (2.6.5) *on the preceding page holds and that* μ *is the uniform probability measure on the finite set of parameters* $\widehat{\Theta}$ *defined by eq.* (2.6.6) *on the previous page. Let us put*

$$\delta = \frac{\log(mn) + 2\log(\varepsilon^{-1}) + \log(|\Lambda|)}{n},$$

and

$$\chi = \sup_{x \,\in\, [n^{-1/2}, n^{1/2}]} \inf_{\lambda \,\in\, \Lambda} \cosh\Big[\log\Big(\frac{\lambda}{(1-\lambda)x}\Big)\Big].$$

Let us assume that there are θ_+ *and* $\theta_- \in \Theta$ *such that* $\overline{p}_+ = p(\theta_+)$, $\overline{p}_- = p(\theta_-)$, $q_+ = \mathbb{E}\big[\mathbb{1}(Y \in B) f_{\theta_+}(X)\big]$ *and* $q_- = \mathbb{E}\big[\mathbb{1}(Y \in B) f_{\theta_-}(X)\big]$ *satisfy*

$$\begin{aligned}
&\min\{q_-, q_+\} \geqslant 8\chi^2\delta, \\
&\overline{p}_+ - \overline{p}_- \geqslant 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\Big(1 + \frac{4\chi^2\delta}{q_+}\Big) + \frac{(2+\sqrt{2})\delta}{q_+} \\
&\qquad\qquad + 2\chi\sqrt{\frac{\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\Big(1 + \frac{4\chi^2\delta}{q_-}\Big) + \frac{(2+\sqrt{2})\delta}{q_-}. \qquad (2.6.7)
\end{aligned}$$

Then with probability at least $1-2\varepsilon$

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0.$$

Therefore, the test defined by eq. (2.6.2) *on page 89 has in this case a probability of false acceptance at most equal to* 2ε.

If we assume instead of eq. (2.6.7) *on the preceding page that*

$$\overline{p}_+ - 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1+\frac{4\chi^2\delta}{q_+}\right) - \frac{(2+\sqrt{2})\delta}{q_+} > 1-\eta, \tag{2.6.8}$$

or that

$$\overline{p}_- + 2\chi\sqrt{\frac{\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1+\frac{4\chi^2\delta}{q_-}\right) + \frac{(2+\sqrt{2})\delta}{q_-} > \eta, \tag{2.6.9}$$

then with probability at least $1-\varepsilon$,

$$\inf_{p\in[\eta,1-\eta]} \max\Big\{B_-(p), B_+(p)\Big\} > 0.$$

Therefore the test defined by eq. (2.6.3) *on page 90 has a probability of false acceptance not greater than* ε *in this case.*

These two statements are consequences of the following more precise one. Each of the two following inequalities holds with probability at least $1-\varepsilon$ *(so that both hold together with probability at least* $1-2\varepsilon$*):*

$$B_-\left(\overline{p}_+ - 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1+\frac{4\chi^2\delta}{q_+}\right) - \frac{(2+\sqrt{2})\delta}{q_+}\right) > 0 \tag{2.6.10}$$

$$B_+\left(\overline{p}_- + 2\chi\sqrt{\frac{\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1+\frac{4\chi^2\delta}{q_-}\right) + \frac{(2+\sqrt{2})\delta}{q_-}\right) > 0. \tag{2.6.11}$$

Proof. Let us assume for a while the more precise statement at the end of the proposition. Under the hypothesis specified by eq. (2.6.7) on the preceding page, there is $p_* \in [0,1]$, such that

$$\begin{aligned}\overline{p}_- + 2\chi\sqrt{\frac{\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1+\frac{4\chi^2\delta}{q_-}\right) + \frac{(2+\sqrt{2})\delta}{q_-} \\ \leqslant p_* \\ \leqslant \overline{p}_+ - 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1+\frac{4\chi^2\delta}{q_+}\right) - \frac{(2+\sqrt{2})\delta}{q_+}.\end{aligned}$$

As a consequence of the fact that $p \mapsto B_-(p)$ is non-increasing and $p \mapsto B_+(p)$ is non-decreasing, this implies from the precise statement at the end of the proposition that with probability at least $1-2\varepsilon$, $\min\Big\{B_-(p_*), B_+(p_*)\Big\} > 0$, and therefore that

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} \geqslant \min\Big\{\inf_{p\in[0,p_*]} B_-(p), \inf_{p\in[p_*,1]} B_+(p)\Big\} = \min\Big\{B_-(p_*), B_+(p_*)\Big\} > 0.$$

In the same way, if we assume eq. (2.6.8) on the preceding page, we get with probability at least $1-\varepsilon$ that $B_-(1-\eta) > 0$ and if we assume that eq. (2.6.9) on the previous page, we get with probability at least $1-\varepsilon$ that $B_+(\eta) > 0$.

Let us now prove the precise statement at the end of the proposition. Let us put

$$p = \overline{p}_+ - 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{4\chi^2\delta}{q_+}\right) - \frac{(2+\sqrt{2})\delta}{q_+}.$$

Let us choose $\lambda \in \Lambda$ such that

$$\lambda \in \operatorname*{argmin}_{\zeta\in\Lambda} \cosh\left[\log\left(\frac{\zeta}{1-\zeta}\sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}}\,\right)\right].$$

Let us remark that $\overline{p}_+ - p \geqslant \dfrac{\delta}{q_+}$, so that

$$\sqrt{\frac{1}{n}} \leqslant \sqrt{\frac{\delta}{q_+}} \leqslant \sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}} \leqslant \sqrt{\frac{q_+}{\delta}} \leqslant \sqrt{n}.$$

This implies that

$$\cosh\left[\log\left(\frac{\lambda}{1-\lambda}\sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}}\,\right)\right] \leqslant \chi.$$

Let us remark also that

$$B_-(p) \geqslant \sum_{i=1}^{n} \log\Big[1 + \lambda F_{\theta_+,i}(p)\Big] - \log(mn) - \log\Big(|\Lambda|/\varepsilon\Big).$$

According to lemma 2.6.2 on page 91, with probability at least $1-\varepsilon$,

$$\sum_i \log\Big(1 + \lambda F_{\theta_+,i}(p)\Big) \geqslant -\log\left(\varepsilon^{-1}\right) - \lambda(p - \overline{p}_+) - \frac{\lambda^2}{1-\lambda}\Big(\overline{p}_+\big(1-\overline{p}_+\big) + \big(p-\overline{p}_+\big)^2\Big).$$

This implies that with probability at least $1-\varepsilon$,

$$B_-(p) > nq_+\lambda\left((\overline{p}_+ - p) - \frac{\lambda}{1-\lambda}\left[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2\right] - \frac{\delta}{\lambda q_+}\right).$$

Decomposing $\dfrac{\delta}{\lambda q_+} = \dfrac{\delta(1-\lambda)}{\lambda q_+} + \dfrac{\delta}{q_+}$ and using the identity

$$ax + b/x = 2\sqrt{ab}\cosh\left[\log\left(x\sqrt{\frac{a}{b}}\right)\right], \qquad x, a, b \in \mathbb{R}_+^*,$$

we obtain that with probability at least $1-\varepsilon$,

$$\begin{aligned} B_-(p) > nq_+\lambda\Bigg\{(\overline{p}_+ - p) - \frac{\delta}{q_+} - 2\sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]\delta}{q_+}} \\ \times \cosh\left[\log\left(\frac{\lambda}{1-\lambda}\sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}}\,\right)\right]\Bigg\} \\ \geqslant nq_+\lambda\left(\overline{p}_+ - p - \frac{\delta}{q_+} - 2\chi\sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]\delta}{q_+}}\,\right). \end{aligned}$$

Let us put $\rho = \dfrac{4\chi^2\delta}{q_+}$, $v = \overline{p}_+(1-\overline{p}_+)$, $y = \dfrac{\delta}{q_+}$ and $x = \overline{p}_+ - p$. The sign of the right-hand side of the previous inequality is the same as the sign of

$$\begin{aligned} \left(\overline{p}_+ - p - \frac{\delta}{q_+}\right)^2 - \frac{4\chi^2\delta[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]}{q_+} \\ = x^2 - 2yx + y^2 - \rho v - \rho x^2 \\ = (1-\rho)x^2 - 2yx - (\rho v - y^2). \end{aligned}$$

This quantity is non negative when x is not less than

$$\begin{aligned} \frac{y}{1-\rho} + \sqrt{\frac{y^2}{(1-\rho)^2} + \frac{\rho v - y^2}{1-\rho}} = \frac{y}{1-\rho} + \sqrt{\frac{\rho y^2}{(1-\rho)^2} + \frac{\rho v}{1-\rho}} \\ \leqslant \sqrt{\frac{\rho v}{1-\rho}} + \frac{(1+\sqrt{\rho})y}{1-\rho} \leqslant \sqrt{\rho v}\left(1 + \frac{\rho}{2(1-\rho)}\right) + \frac{(1+\sqrt{\rho})y}{1-\rho} \\ \leqslant \sqrt{\rho v}(1+\rho) + (2+\sqrt{2})y = 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{4\chi^2\delta}{q_+}\right) + \frac{(2+\sqrt{2})\delta}{q_+}, \end{aligned}$$

where we have used the inequalities $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$, $\sqrt{\dfrac{1}{1-\rho}} \leqslant 1 + \dfrac{\rho}{2(1-\rho)}$ and $\rho \leqslant 1/2$.

This proves that with probability at least $1-\varepsilon$,

$$B_-\left(\overline{p}_+ - 2\chi\sqrt{\frac{\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1+\frac{4\chi^2\delta}{q_+}\right) - \frac{(2+\sqrt{2})\delta}{q_+}\right) > 0.$$

The proof of eq. (2.6.11) on page 93 is obtained exactly in the same way. □

2.7 Testing Markov substitute sets without simulating the parse process

In this section, we will embrace a broader scope than testing for a single Markov substitute pair of sets.

We will assume that some partition $\mathscr{B}$ of D^+ into Markov substitute set is known, and we will simultaneously test whether any pair $(B_1, B_2) \in \mathscr{B}^2$ is such that $B = B_1 \cup B_2$ is a Markov substitute set.

Instead of simulating random B-parse processes, which add some random fluctuations to the observations, we will consider the following conditional expectations:

$$F_{B_1,B_2,\theta}(s,p) = \mathbb{E}\Big\{\Big[\mathbb{1}\Big(Y_{B_1\cup B_2} \in B_1\Big) - p\,\mathbb{1}\Big(Y_{B_1\cup B_2} \in B_1 \cup B_2\Big)\Big]\mathbb{1}\Big(X_{B_1\cup B_2} \in \theta\Big)\,\Big|\,S = s\Big\}. \quad (2.7.1)$$

Let us notice that similar results could be stated about

$$F_{B_1,B_2,\theta}(s,p) = \mathbb{E}\Big\{\Big[\mathbb{1}\Big(X_{B_1\cup B_2} \in \theta\Big) - p\Big]\mathbb{1}(Y_{B_1\cup B_2} \in B_1)\,\Big|\,S = s\Big\}, \quad (2.7.2)$$

We will not follow this second path, since the first is more closely related to the invariant dynamics we will use to simulate from the model. The results of this section also apply to the alternative choices of test functions described in section 2.4.1 on page 84.

In the case when

$$p = \mathbb{P}\Big(Y_{B_1\cup B_2} \in B_1 \,|\, X_{B_1\cup B_1} \in \theta, Y_{B_1\cup B_2} \in B_1 \cup B_2\Big),$$

it is easy to see that $\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) = 0$, and consequently that

$$\mathbb{E}\Big[\exp\Big(\log\Big(1+\lambda F_{B_1,B_2,\theta}(S,p)\Big)\Big)\Big] = 1, \qquad \lambda \in]-1,1[.$$

Let us introduce $\overline{\Theta} = \mathscr{B}^2 \times \Theta$, the parameter space indexing the random variables $F_{B_1,B_2,\theta}(S,p)$ (apart from p that can be viewed as another parameter).

In order to master the complexity of the problem in a data driven way, we will use a shadow sample, as in the proof of Vapnik-Chervonenkis complexity bounds. However, the actual observation of this shadow sample will not be required, because our generalization bounds will eventually be obtained by taking a conditional expectation with respect to the observed sample $S_{1:n}$.

Lemma 2.7.1

Consider an i.i.d sample S_i, $1 \leqslant i \leqslant kn$ of size kn and an exchangeable prior distribution $\mu_{S_{1:kn}} \in \mathscr{M}_+^1(\overline{\Theta})$, where we have put for short $S_{1:kn} = (S_i, 1 \leqslant i \leqslant kn)$.

For any $\lambda \in]-1,1[$, any $\theta \in \overline{\Theta}$, any $p \in [0,1]$,

$$\mathbb{E}\left(\int \exp\left(\sum_{i=1}^{n} \log\bigl(1+\lambda F_\theta(S_i,p)\bigr) - \frac{\lambda}{k}\sum_{i=1}^{kn} F_\theta(S_i,p)\right) \mathrm{d}\mu(\theta)\right) \leqslant 1.$$

Proof. Let us put $W_i = \lambda F_\theta(S_i,p)$. For any exchangeable function $G(W_{1:kn})$,

$$\begin{aligned}
\mathbb{E}\left(G(W_{1:kn})\prod_{i=1}^{n}\bigl(1+W_i\bigr)\right) &= \mathbb{E}\left(G(W_{1:kn})\sum_{j_1,\ldots,j_n=0}^{k-1}\prod_{i=1}^{n}\frac{\bigl(1+W_{i+j_i n}\bigr)}{k}\right)\\
&= \mathbb{E}\left(G(W_{1:kn})\prod_{i=1}^{n}\left(1+\frac{1}{k}\sum_{j=0}^{k-1}W_{i+jn}\right)\right)\\
&= \mathbb{E}\left(G(W_{1:kn})\exp\left(\sum_{i=1}^{n}\log\left(1+\frac{1}{k}\sum_{j=0}^{k-1}W_{i+jn}\right)\right)\right)\\
&\leqslant \mathbb{E}\left(G(W_{1:kn})\exp\left(n\log\left(1+\frac{1}{nk}\sum_{i=1}^{kn}W_i\right)\right)\right)\\
&\leqslant \mathbb{E}\left(G(W_{1:kn})\exp\left(\frac{1}{k}\sum_{i=1}^{kn}W_i\right)\right).
\end{aligned}$$

If we take for any value of θ

$$G_\theta(W_{1:kn}) = \mu(\theta)\exp\left(-\frac{\lambda}{k}\sum_{i=1}^{kn}F_\theta(S_i,p)\right),$$

we get that

$$\begin{aligned}
&\mathbb{E}\left(\prod_{i=1}^{n}\bigl(1+\lambda F_\theta(S_i,p)\bigr)\exp\left(-\frac{\lambda}{k}\sum_{i=1}^{kn}F_\theta(S_i,p)\right)\mu(\theta)\right)\\
&\qquad\leqslant \mathbb{E}\left(\exp\left(\frac{\lambda}{k}\sum_{i=1}^{kn}F_\theta(S_i,p)\right)\exp\left(-\frac{\lambda}{k}\sum_{i=1}^{kn}F_\theta(S_i,p)\right)\mu(\theta)\right) = \mathbb{E}\bigl(\mu(\theta)\bigr).
\end{aligned}$$

The inequality follows by summing on θ. □

Let us now make a more specific choice for μ.

Let us consider some indicator function $h : \mathscr{B} \times D^+ \to \{0,1\}$, and the exchangeable prior on $\mathscr{B}$ defined by

$$\nu(B) = \frac{1}{kn} \sum_{i=1}^{kn} \left(\sum_{B' \in \mathscr{B}} h(B', S_i) \right)^{-1} h(B, S_i).$$

Let us also consider another indicator function $g : \mathscr{B} \times \Theta \times D^+ \to \{0,1\}$, and define

$$\xi(\theta \,|\, B_1, B_2) = \frac{1}{kn} \sum_{i=1}^{kn} \left(\sum_{\theta' \in \Theta} g\Big(B_1 \cup B_2, \theta', S_i\Big) \right)^{-1} g\Big(B_1 \cup B_2, \theta, S_i\Big).$$

Let us introduce

$$\nu_1(B) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{B' \in \mathscr{B}} h(B', S_i) \right)^{-1} h(B, S_i), \text{ and}$$

$$\xi_1\Big(\theta \,|\, B_1, B_2\Big) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{\theta' \in \Theta} g\Big(B_1 \cup B_2, \theta', S_i\Big) \right)^{-1} g\Big(B_1 \cup B_2, \theta, S_i\Big).$$

We can then choose the prior distribution

$$\mu(B_1, B_2, \theta) = \nu(B_1)\nu(B_2)\xi(\theta \,|\, B_1, B_2), \tag{2.7.3}$$

and decompose it as $\mu = k^{-3}\mu_1 + \Big(1 - k^{-3}\Big)\mu_2$, where

$$\mu_1(B_1, B_2, \theta) = \nu_1(B_1)\nu_1(B_2)\xi_1(\theta \,|\, B_1, B_2)$$

depends only on the observable sample $S_{1:n}$. We can remark moreover that

$$\mathscr{K}(\rho, \mu) \leqslant \mathscr{K}(\rho, \mu_1) + 3\log(k). \tag{2.7.4}$$

Let us remark that by the law of large numbers, when n is large, μ_1 will get close to μ.

We have more specifically in mind the following weight functions

$$h(B, s) = \mathbb{1}\Big(\mathscr{S}(s, B) \neq \varnothing\Big),$$

$$g(B, \theta, s) = \mathbb{1}\Big(\exists\,(x, y) \in \mathscr{S}\Big(s, B\Big); x \in \theta\Big).$$

With this choice, the support of ν_1 is made of all Markov substitute sets $B \in \mathscr{B}$ such that one expression of B is observed in the sample $S_{1:n}$. In the same way, the support of $\mu_1\Big(\cdot|B_1, B_2\Big)$ is made of all the context parameters $\theta \in \Theta$ indexing contexts that appear in the sample $S_{1:n}$.

Let us put

$$\begin{aligned}\overline{h}(s) &= \sum_{B\in\mathscr{B}} h(B,s),\\ \overline{g}(s) &= \sup_{B_1,\,B_2\,\in\,\mathscr{B}} \sum_{\theta\in\Theta} g(B_1\cup B_2,\theta,s).\end{aligned}$$

We can easily see that as soon as $\mu_1\Big(B_1, B_2, \theta\Big) > 0$, then

$$\mu_1(B_1,B_2,\theta) \geqslant n^{-3}\Big(\max_{i=1,\dots,n}\overline{h}(S_i)\Big)^{-2}\Big(\max_{i=1,\dots,n}\overline{g}(S_i)\Big)^{-1}.$$

We can improve this inequality if we choose a conditional probability kernel π satisfying eq. (2.1.4) on page 67 only when $\overline{h}(s) \leqslant C_1$ and $\overline{g}(s) \leqslant C_2$, and such that $\pi\Big(s,(s,\epsilon),\epsilon\Big) = 1$ otherwise. With this choice of kernel, we test the Markov substitute property only for the sentences s satisfying the above constraint. More specifically, we test that

$$\mathbb{P}_S\Big(\alpha(x,y)\Big) = \mathbb{P}_S\Big(\alpha(x,B)\Big)q_B(y), \qquad \inf_{y'\in B}\pi\Big(\alpha(x,y'),x,y'\Big) > 0.$$

On the other hand, we are now sure that

$$\mu_1\Big(B_1,B_2,\theta\Big) \geqslant n^{-3}C_1^{-2}C_2^{-1},$$

as soon as $\mu_1\Big(B_1,B_2,\theta) > 0$,

From lemma 2.7.1 on page 97, we get in this context

Proposition 2.7.2
Consider some finite set $\Lambda \subset]0,1[$, and $-\Lambda = \Big\{-\lambda, \lambda\in\Lambda\Big\}$. With probability at least $1-2\varepsilon$, for any $\lambda \in \Lambda\cup(-\Lambda)$, any $p\in\mathscr{P}\subset]0,1[$, any $\rho\in\mathscr{M}_+^1(\overline{\Theta})$,

$$\begin{aligned}\sum_{i=1}^n \frac{(k-1)\lambda}{k}\int F_\theta(S_i,p)\,\mathrm{d}\rho(\theta) - \frac{\lambda^2}{2(1-|\lambda|)^2}\int F_\theta(S_i,p)^2\,\mathrm{d}\rho(\theta)&\\ \leqslant \int\sum_{i=1}^n\Big[\log\Big(1+\lambda F_\theta(S_i,p)\Big) - \frac{\lambda}{k}F_\theta(S_i,p)\Big]\,\mathrm{d}\rho(\theta)&\\ \leqslant \frac{(k-1)n\lambda}{k}\int\mathbb{E}\Big[F_\theta(S,p)\Big]\,\mathrm{d}\rho(\theta) + \mathscr{K}(\rho,\mu_1) + 3\log(k) + \log\Big(|\Lambda|\,|\mathscr{P}|/\varepsilon\Big).&\end{aligned}$$

Remark 2.7.1
From the shape of this equation, it makes sense to take k of order $k = 10$ or so.

PROOF. The first inequality is a consequence of the fact that

$$\log(1+\lambda x) = \lambda x - \int_0^{\lambda x} \frac{(\lambda x - y)}{(1+y)^2}\,\mathrm{d}y \geqslant \lambda x - \frac{\lambda^2 x^2}{2(1-|\lambda|)^2}, \qquad x \in [-1,1], \lambda \in]-1,1[.$$

In order to prove the second inequality, let us put for short

$$W_1(\theta) = \sum_{i=1}^{n} \log\Big(1 + \lambda F_\theta(S_i, p)\Big) - \frac{\lambda}{k} \sum_{i=1}^{n} F_\theta(S_i, p),$$
$$W_2(\theta) = -\frac{\lambda}{k} \sum_{i=n+1}^{kn} F_\theta(S_i, p).$$

According to the previous lemma,

$$\mathbb{E}\left(\exp\left(\sup_{\rho \in \mathscr{M}(\overline{\Theta})} \int \Big[W_1(\theta) + W_2(\theta)\Big]\,\mathrm{d}\rho(\theta) - \mathscr{K}(\rho, \mu_1) - 3\log(k)\right)\right) \leqslant 1.$$

Using

$$\mathbb{E}\Big(\exp(\cdot)\Big) \geqslant \mathbb{E}\Big[\exp\Big(\mathbb{E}(\cdot|S_{1:n})\Big)\Big],$$
$$\mathbb{E}\Big(\sup_\rho(\cdot)|S_{1:n}\Big) \geqslant \sup_\rho \mathbb{E}(\cdot|S_{1:n}),$$
$$\mathbb{E}(W_1(\theta)|S_{1:n}) = W_1(\theta),$$
$$\mathbb{E}(W_2(\theta)|S_{1:n}) = \mathbb{E}\Big(W_2(\theta)\Big),$$
$$\text{and } \mathbb{E}\Big(\mathscr{K}(\rho, \mu_1)|S_{1:n}\Big) = \mathscr{K}(\rho, \mu_1),$$

we obtain that

$$\mathbb{E}\left(\exp\left(\sup_{\rho \in \mathscr{M}_+^1(\overline{\Theta})} \int W_1(\theta) + \mathbb{E}\Big(W_2(\theta)\Big)\,\mathrm{d}\rho(\theta) - \mathscr{K}(\rho, \mu_1) - 3\log(k)\right)\right) \leqslant 1,$$

from which the proposition is an easy consequence. □

2.7.1 Definition of the test and probability of false rejection

This leads us to define an alternative to the test proposed in section 2.6 on page 88. Let $p(B_1, B_2, \theta)$ be the value of p such that

$$\mathbb{E}\Big(F_{B_1,B_2,\theta}(S, p)\Big) = 0,$$

so that

$$p\Big(B_1, B_2, \theta\Big) = \mathbb{P}\Big(Y_{B_1\cup B_2} \in B_1 \,|\, X_{B_1\cup B_2} \in \theta, Y_{B_1\cup B_2} \in B_1 \cup B_2\Big),$$

where $F_{B_1,B_2,\theta}(S,p)$ is defined by eq. (2.7.1) on page 96.

Let us define

$$p_+(B_1,B_2) = \sup\Big\{p(B_1,B_2,\theta) : \theta \in \Theta, \mathbb{P}\Big(X_{B_1\cup B_2} \in \theta, Y_{B_1\cup B_2} \in B_1 \cup B_2\Big) > 0\Big\},$$

and let $p_-(B_1,B_2)$ be the infimum of the same set.

Let us introduce the function

$$\psi(z) = \log(1+z) - k^{-1}z.$$

Proposition 2.7.3
Let Λ be a finite subset of $]0,1[$.

With probability at least $1-2\varepsilon$, for any pair $(B_1,B_2) \in \mathscr{B}^2$,

$$\begin{aligned} B_-\Big(p_+(B_1,B_2)\Big) \stackrel{\text{def}}{=} \sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda} \int \sum_{i=1}^n \psi\Big(\lambda F_{B_1,B_2,\theta}\Big(S_i, p_+(B_1,B_2)\Big)\Big)\,\mathrm{d}\rho(\theta) \\ - \mathscr{K}(\rho,\mu_1) - 3\log(k) - \log\left(\frac{|\Lambda|}{\varepsilon\,\nu_1(B_1)\,\nu_1(B_2)}\right) \leqslant 0 \\ B_+\Big(p_-(B_1,B_2)\Big) \stackrel{\text{def}}{=} \sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda} \int \sum_{i=1}^n \psi\Big(-\lambda F_{B_1,B_2,\theta}\Big(S_i, p_-(B_1,B_2)\Big)\Big)\,\mathrm{d}\rho(\theta) \\ - \mathscr{K}(\rho,\mu_1) - 3\log(k) - \log\left(\frac{|\Lambda|}{\varepsilon\nu_1(B_1)\nu_1(B_2)}\right) \leqslant 0 \end{aligned}$$

Therefore, if we reject the hypothesis that $B_1\cup B_2$ is a Markov substitute set when

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0, \tag{2.7.5}$$

the probability of making a false rejection (after testing all pairs in $\mathscr{B}^2$) is at most 2ε.

In the same way we can reject the hypothesis that $(B_1,B_2) \in \mathscr{B}^2$ is an η-Markov substitute pair of sets when

$$\inf_{p\in[\eta,1-\eta]} \max\Big\{B_-(p), B_+(p)\Big\} > 0,$$

with a probability of rejecting one of the true η-Markov pairs (after testing all pairs in $\mathscr{B}^2$), not greater than 2ε.

Remark 2.7.2
To our knowledge the observable entropy term $\mathscr{K}(\rho, \mu_1)$ is a novel addition to PAC-Bayes theory that was not proposed before in the literature.

This proposition is a direct consequence of the previous one.

We can remark at this point that this result stays true for any test functions $F_\theta(S, p)$ such that $|F_\theta(S, p)| \leqslant 1$ and $\mathbb{E}(F_\theta(S, p)) = 0$ for $p = p(B_1, B_2, \theta)$, so this kind of test can be used for a wide range of test functions. However, this proposition gives no indication on the probability on false acceptance (which is admittedly the more important part). Section 2.7.5 on page 111 will deal with this issue, but will require a more restrictive form of the test function.

2.7.2 Testing for the weak Markov substitute property

We may be interested in situations where the target language does not have a sufficient number of strong Markov substitute sets, but nevertheless has a bigger number of weak Markov substitute sets. In this case, we will be able to generate the support of the language, running an ill specified Markov substitute set model whose Markov substitute sets are only weak Markov substitute sets for the true language distribution.

This is how we propose to modify the test. To get the right support, we want to select pairs which satisfy $0 < p_-(B_1, B_2) \leqslant p_+(B_1, B_2) < 1$. Since it will not be possible to test that $p_-(B_1, B_2) = 0$ or $p_+(B_1, B_2) = 1$, we will take some margin and focus on pairs for which

$$\eta \leqslant p_-(B_1, B_2) \leqslant p_+(B_1, B_2) \leqslant 1 - \eta,$$

where $\eta \in]0, 1/2[$ is a fixed parameter. We will also relax the property that $p_-(B_1, B_2) = p_+(B_1, B_2)$ to $p_+(B_1, B_2) - p_-(B_1, B_2) \leqslant \gamma$, where $\gamma \in [0, 1 - 2\eta]$ is a second parameter. Pairs satisfying those two conditions will be called γ-weak η-Markov substitute pairs of sets. Let us remark that when $\gamma = 1 - 2\eta$, the second condition is void (because it is implied by the first one). Let us also remark that the first condition says that the ratio between the probabilities of observing $y \in B_1$ and $y \in B_2$ in the same context belongs to the interval $\Big[\eta/(1-\eta), (1-\eta)/\eta\Big]$.

Corollary 2.7.4
With probability at least $1 - 2\varepsilon$, for any γ-weak η-Markov substitute pair,

$$\inf_{p \in [\eta, 1-\eta-\gamma]} \max\Big\{B_-(p+\gamma), B_+(p)\Big\} \leqslant 0,$$

so that if we accept as γ-weak η-Markov pairs all pairs satisfying this condition, the probability of false rejection, that is, the probability that some truly γ-weak η-Markov pair in $\mathscr{B}^2$ may have been unduly rejected is not greater than 2ε.

2.7.3 Computation of the test

Let us remark first that a more explicit formula is available for the test function. Indeed, the solution of the optimization with respect to ρ is explicit, according to the general formula

$$\sup_{\rho\in\mathscr{M}_+^1(\Theta)} \int h(\theta)\,\mathrm{d}\rho(\theta) - \mathscr{K}(\rho,\mu_1) = \log\Big[\int \exp\Big(h(\theta)\Big)\,\mathrm{d}\mu_1(\theta)\Big],$$

the supremum being achieved when

$$\frac{\mathrm{d}\rho}{\mathrm{d}\mu_1} = \frac{\exp(h)}{\displaystyle\int \exp(h)\,\mathrm{d}\mu_1}\cdot$$

According to this general formula

$$B_-(p) = \log\left[\int \exp\left[\sum_{i=1}^n \psi\Big(\lambda F_{B_1,B_2,\theta}\big(S_i,p\big)\Big)\right]\mathrm{d}\mu_1(\theta)\right] - \log\left(\frac{k^3|\Lambda|}{\varepsilon\nu_1(B_1)\nu_1(B_2)}\right), \tag{2.7.6}$$

$$B_+(p) = \log\left[\int \exp\left[\sum_{i=1}^n \psi\Big(-\lambda F_{B_1,B_2,\theta}\big(S_i,p\big)\Big)\right]\mathrm{d}\mu_1(\theta)\right] - \log\left(\frac{k^3|\Lambda|}{\varepsilon\nu_1(B_1)\nu_1(B_2)}\right). \tag{2.7.7}$$

These formulas are indeed very compact and pleasing from this point of view, but they do not show the structure of the test function. For example they do not show whether or how the test function involves some variance and complexity estimates.

To show this structure, we are going in the following lines to compute a more explicit approximation, in the form of a close empirical lower bound for B_- and B_+. Let us put

$$\begin{aligned} a(\theta,i) &= \mathbb{P}\Big(X_{B_1\cup B_2}\in\theta, Y_{B_1\cup B_2}\in B_1 \,|\, S=S_i\Big),\\ b(\theta,i) &= \mathbb{P}\Big(X_{B_1\cup B_2}\in\theta, Y_{B_1\cup B_2}\in B_1\cup B_2 \,|\, S=S_i\Big),\\ \delta(\rho) &= \mathscr{K}(\rho,\mu_1) + \log\left(\frac{k^3|\Lambda|}{\varepsilon\nu_1(B_1)\nu_2(B_2)}\right)\cdot \end{aligned}$$

so that

$$F_{B_1,B_2,\theta}(S_i,p) = a(\theta,i) - p\,b(\theta,i).$$

According to proposition 2.7.2 on page 99,

$$B_-(p) \geqslant \sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda} \int \left(\sum_{i=1}^n \frac{(k-1)\lambda}{k} \Big[a(\theta,i) - p\, b(\theta,i) \Big] \right.$$
$$\left. - \frac{\lambda^2}{2(1-\lambda)^2} \Big[a(\theta,i) - p\, b(\theta,i) \Big]^2 \right) \mathrm{d}\rho(\theta) - \delta(\rho).$$

Let us put, for $k = 1, 2$,

$$\begin{aligned}
a_k(\rho) &= \int \sum_{i=1}^n a(\theta,i)^k \,\mathrm{d}\rho(\theta), \\
b_k(\rho) &= \int \sum_{i=1}^n b(\theta,i)^k \,\mathrm{d}\rho(\theta), \\
\widehat{p}_1(\rho) &= \frac{a_1(\rho)}{b_1(\rho)}, \\
\widehat{p}_2(\rho) &= b_2(\rho)^{-1} \int \sum_{i=1}^n a(\theta,i)\, b(\theta,i) \,\mathrm{d}\rho(\theta), \\
\widehat{v}(\rho) &= b_2(\rho)^{-1} \int \sum_{i=1}^n \Big(a(\theta,i) - \widehat{p}_2\, b(\theta,i) \Big)^2 \,\mathrm{d}\rho(\theta), \\
\alpha &= \frac{k-1}{k}\cdot
\end{aligned}$$

We get that

$$B_-(p) \geqslant \sup_{\rho\in\mathscr{M}_+^1(\Theta),\lambda\in\Lambda} \alpha b_1(\rho)\lambda\Big(\widehat{p}_1(\rho) - p\Big) - \frac{\lambda^2}{2(1-\lambda)^2} b_2(\rho) \Big[\widehat{v}(\rho) + \Big(\widehat{p}_2(\rho) - p\Big)^2 \Big] - \delta(\rho).$$

We deduce that with probability at least $1-\varepsilon$, $p_+(y_1, y_2)$ is solution in p of the inequalities (indexed by $\lambda \in \Lambda$ and $\rho \in \mathscr{M}_+^1(\Theta)$)

$$\widehat{p}_1(\rho) - p - \frac{\lambda b_2(\rho)}{2(1-\lambda)^2 \alpha b_1(\rho)} \Big[\widehat{v}(\rho) + \Big(\widehat{p}_2(\rho) - p\Big)^2 \Big] - \frac{\delta(\rho)}{\lambda \alpha b_1(\rho)} \leqslant 0.$$

Using the inequality $(1-\lambda)^{-2} \leqslant (1-2\lambda)^{-1}$, and optimizing the value of λ, we get

$$\widehat{p}_1(\rho) - p - \chi \sqrt{\frac{2b_2(\rho)\Big[\widehat{v}(\rho) + \Big(\widehat{p}_2(\rho) - p\Big)^2\Big]\delta(\rho)}{\alpha^2 b_1(\rho)^2}} - \frac{2\delta(\rho)}{\alpha b_1(\rho)} \leqslant 0, \tag{2.7.8}$$

where

$$\chi = \inf_{\lambda\in\Lambda} \cosh\left[\log\left(\frac{\lambda}{1-2\lambda} \sqrt{\frac{b_2(\rho)[\widehat{v}(\rho) + (\widehat{p}_2(\rho) - p)^2]}{2\delta(\rho)}} \right) \right].$$

We are now going to solve this inequality in p. Although this is straightforward, it will get a litte technical, so this may be the best place to give an interpratation of the bound. We have an empirical estimate $\widehat{p}_1(\rho)$ of some mixture of conditional probabilities (the mixture being related to ρ). In this empirical estimate, $b_1(\rho)$ plays the role of an empirical "effective" sample dimension. This empirical estimate is corrected by a deviation term, to provide a lower bound for $p_+(B_1, B_2)$ with confidence level $1-\varepsilon$. We also get an upper bound for $p_-(B_1, B_2)$ with confidence level $1-\varepsilon$,

$$p - \widehat{p}_1(\rho) - \chi\sqrt{\frac{2b_2(\rho)\left[\widehat{v}(\rho) + \left(\widehat{p}_2(\rho) - p\right)^2\right]\delta(\rho)}{\alpha^2 b_1(\rho)^2}} - \frac{2\delta(\rho)}{\alpha b_1(\rho)} \leqslant 0,$$

resulting in a final joint statement holding at confidence level $1-2\varepsilon$. The deviation term involves a variance estimate given by $\widehat{v}(\rho)$ and a complexity (or "effective dimension") estimate given by $\delta(\rho)$. The confidence level is uniform in ρ, which includes the case where ρ is a Dirac mass at parameter value θ, but a better compromise may be found for a distribution ρ with more spread (as shown by eq. (2.7.6) on page 103, the optimal ρ is indeed a Gibbs distribution and not a Dirac mass, although a Dirac mass may be a close winner in this discrete setting). Optimizing the bound in ρ will give a tighter lower bound for $p_+(B_1, B_2)$, which is itself defined as an upper bound in θ, and for $p_-(B_1, B_2)$ which is defined as a lower bound in θ.

Let us now solve explicitly the quadratic inequality eq. (2.7.8) on the facing page. Let us put

$$\begin{aligned}
x &= \widehat{p}_2(\rho) - p_+(B_1, B_2),\\
y &= \widehat{p}_2(\rho) - \widehat{p}_1(\rho) + \frac{2\delta(\rho)}{\alpha b_1(\rho)},\\
z &= \frac{2\chi^2 b_2(\rho)\delta(\rho)}{\alpha^2 b_1(\rho)^2},\\
v &= \widehat{v}(\rho).
\end{aligned}$$

We deduce from the previous discussion that with probability at least $1-\varepsilon$,

$$(x-y)^2 - z(v+x^2) \leqslant 0,$$

which can also be written as

$$(1-z)x^2 - 2yx - (zv - y^2) \leqslant 0,$$

implying that

$$\begin{aligned} x &\leqslant \frac{y}{1-z} + \sqrt{\frac{y^2}{(1-z)^2} + \frac{zv-y^2}{1-z}} \\ &= \frac{y}{1-z} + \sqrt{\frac{zy^2}{(1-z)^2} + \frac{zv}{1-z}} \\ &\leqslant \frac{1+\sqrt{z}}{1-z} y + \sqrt{zv}\left(1 + \frac{z}{2(1-z)}\right). \end{aligned}$$

Thus, if we put

$$\widehat{p}_+(\rho) = \widehat{p}_2(\rho) - \frac{1+\sqrt{z}}{1-z}\, y - \sqrt{zv}\left(1 + \frac{z}{2(1-z)}\right),$$

and in the same way (reasoning with $B_+(p)$)

$$\widehat{p}_-(\rho) = \widehat{p}_2(\rho) + \frac{1+\sqrt{z}}{1-z}\Big[y + 2\Big(\widehat{p}_1(\rho) - \widehat{p}_2(\rho)\Big)\Big] + \sqrt{zv}\left(1 + \frac{z}{2(1-z)}\right),$$

we obtain

Proposition 2.7.5
With the previous notations, with probability at least $1-2\varepsilon$, for any $\rho \in \mathscr{M}_+^1(\Theta)$,

$$\begin{aligned} p_-(B_1, B_2) &< \inf_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_-(\rho), \\ \sup_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_+(\rho) &< p_+(B_1, B_2). \end{aligned}$$

Therefore, we can reject the hypothesis that $B_1 \cup B_2$ is a Markov substitute set with a probability of false rejection not greater than 2ε whenever

$$\inf_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_-(\rho) \leqslant \sup_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_+(\rho).$$

In the same way, we can reject the hypothesis that the pair (B_1, B_2) is a γ-weak η-Markov substitute pair of sets with a probability of false rejection not greater than 2ε whenever

$$\begin{aligned} &\sup_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_+(\rho) \geqslant 1-\eta, \inf_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_-(\rho) \leqslant \eta \\ &\textit{or} \sup_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_+(\rho) - \inf_{\rho \in \mathscr{M}_+^1(\Theta)} \widehat{p}_-(\rho) \geqslant \gamma. \end{aligned}$$

We hope that these more explicit approximations of the previous test will help the reader getting a better understanding of the structure of the test. The main thing to understand is that hidden in the tighter test of proposition 2.7.3 on page 101 are lower bound estimators for $p_+(B_1, B_2)$ and upper bound estimators for $p_-(B_1, B_2)$, and that these confidence bounds hold uniformly for any $(B_1, B_2) \in \mathscr{B}^2$, at the given confidence level.

We would also like to point out that the estimates $\widehat{p}_+(\rho)$ and $\widehat{p}_-(\rho)$ contain an explicit empirical variance estimate, that was somehow hidden in the properties of the influence function ψ in the previous section.

2.7.4 Some numerical examples and considerations on the accuracy of the tests

We will here give some numerical examples of the computations presented in the previous subsections. Let us assume for simplicity that $B_1 = \{y_1\}$, $B_2 = \{y_2\}$, and $\pi\Big(\alpha(x, y), x, y\Big) = 1$ for any

$$(x, y) \in \bigcup_{i=1}^{n} \mathscr{S}\Big(S_i, \{y_1, y_2\}\Big),$$

meaning that there is at most one way to $\{y_1, y_2\}$-parse any given sentence of the observed sample. In this simple case $(X_{\{y_1,y_2\},i}, Y_{\{y_1,y_2\},i})$ is a deterministic function of S_i, so that $a(\theta, i), b(\theta, i) \in \{0, 1\}$. As a consequence, $a_2(\rho) = a_1(\rho) \stackrel{\text{def}}{=} a(\rho)$, $b_2(\rho) = b_1(\rho) \stackrel{\text{def}}{=} b(\rho)$, $\widehat{p}_2(\rho) = \widehat{p}_1(\rho) \stackrel{\text{def}}{=} \widehat{p}(\rho)$ and $\widehat{v}(\rho) = \widehat{p}(\rho)\Big(1 - \widehat{p}(\rho)\Big)$. Making as few approximations as possible, we can take

$$\widehat{p}_+(\rho) = \widehat{p}(\rho) - \frac{y}{1-z} - \sqrt{\frac{zy^2}{(1-z)^2} + \frac{z\widehat{v}(\rho)}{1-z}},$$
$$\widehat{p}_-(\rho) = \widehat{p}(\rho) + \frac{y}{1-z} + \sqrt{\frac{zy^2}{(1-z)^2} + \frac{z\widehat{v}(\rho)}{1-z}}.$$

If we choose μ as mentioned in eq. (2.7.3) on page 98, we get that

$$\delta(\rho) \leqslant \log\Big(k^3\Big(\max_{1\leqslant i\leqslant n} \overline{h}(S_i)\Big)^2\Big(\max_{1\leqslant i\leqslant n} \overline{g}(S_i)\Big)n^3|\Lambda|\varepsilon^{-1}\Big).$$

If we want to be very conservative, taking for instance values of $n = 10^4$, $k = 10$, $\max_{1 \leqslant i \leqslant n} \overline{h}(S_i) = 20$, $\max_{1 \leqslant i \leqslant n} \overline{g}(S_i) = 50$, $|\Lambda| = 10 \log(n)$, we get

$$\begin{aligned} \delta(\rho) &\leqslant \log\Big(10^3 \times 50^3 \times 10^{15} \times \log(10^3)\Big) < 56, \\ \chi &\leqslant \cosh(0.1) \leqslant 1.006, \\ y &\leqslant \frac{2 \times 10 \times 56}{9 \times b(\rho)} \leqslant \frac{125}{b(\rho)}, \\ z &\leqslant \frac{2 \times (1.006)^2 \times 56 \times 10^2}{9^2 \times b(\rho)} \leqslant \frac{140}{b(\rho)}. \end{aligned}$$

Let us compute the minimal gap we have to find between $\widehat{p}_+(\rho_+) = \sup \widehat{p}_+(\rho)$ and $\widehat{p}_-(\rho_-) = \inf \widehat{p}_-(\rho)$ to reject the Markov substitute hypothesis when, let us say, $b(\rho_+) = b(\rho_-) = 1000$. We may bound $\widehat{v}(\rho)$ by $1/4$. We get that we need

$$\widehat{p}(\rho_-) \leqslant \widehat{p}(\rho_+) - 0.71$$

to reject the hypothesis that the pair is a Markov substitute pair, of in other term, we need $a(\rho_-) \leqslant a(\rho_+) + 710$. That is not a very tight bound.

However, if we want to be more optimistic, we may think that the union bound precaution leading to the mathematically correct value of $\delta(\rho)$ computed above is too shy, and that we should get rid of it and keep only $\delta(\rho) = \log(\varepsilon^{-1})$ to get more realistic numerical values. If we are ready to take this kind of liberty with the theory, and choose as before $\varepsilon = 10^{-2}$, we get $\delta(\rho) \leqslant 5$, and consequently

$$\begin{aligned} y &\leqslant \frac{12}{b(\rho)}, \\ z &\leqslant \frac{13}{b(\rho)}. \end{aligned}$$

With this more optimistic guess, using again for comparison $b(\rho_+) = b(\rho_-) = 1000$, $\widehat{v} \leqslant 1/4$, we obtain a required gap of

$$\widehat{p}(\rho_-) \leqslant \widehat{p}(\rho_+) - 0.14.$$

We may also ask in this context what number of observations we need to reject the hypothesis when we can find two contexts (say with the same number of observations as above) such that $\widehat{p}(\rho_+) = 1$ and $\widehat{p}(\rho_-) = 0$ (so that in both cases $\widehat{v} = 0$). We get that we need to observe at least 50 times y_1 and y_2 in disjoint contexts before deciding at a confidence level of 98/100 that $\{y_1, y_2\}$ is not a Markov substitute pair (still ignoring the union bound factors).

This is still conservative, since we can make a direct computation in this simple situation, showing that if $\{y_1, y_2\}$ is a Markov substitute pair, then the probability

of observing 7 times y_1 in one context and 7 times y_2 in another context is not greater than $2 \times 2^{-7} \leqslant 0.016$ (the worst case being when the substitute measure is a Bernoulli with parameter 1/2).

In this context, it is interesting to see whether the non approximated test of proposition 2.7.3 on page 101 is more satisfactory for small sample sizes. In the simplified framework described in this section the test of proposition 2.7.3 on page 101 can be written as

$$B_-(p) = \sup_{\rho \in \mathscr{M}_+^1(\Theta), \lambda \in \Lambda} a(\rho)\psi\Big(\lambda(1-p)\Big) + \Big(b(\rho) - a(\rho)\Big)\psi\Big(-\lambda p\Big) + \delta(\rho), \tag{2.7.9}$$

$$B_+(p) = \sup_{\rho \in \mathscr{M}_+^1(\Theta), \lambda \in \Lambda} a(\rho)\psi\Big(-\lambda(1-p)\Big) + \Big(b(\rho) - a(\rho)\Big)\psi\Big(\lambda p\Big) + \delta(\rho). \tag{2.7.10}$$

The test will in particular reject the hypothesis when

$$\min\Big\{B_-(1/2), B_+(1/2)\Big\} > 0. \tag{2.7.11}$$

Let us see what we get if we forget the entropy and union bound factors. Take as suggested before $\delta(\rho) = \log(\varepsilon^{-1}) = \log(100) \simeq 4.6$, $b(\rho_+) = b(\rho_-) = a(\rho_+) = b$, $a(\rho_-) = 0$, and look for the minimum value of b for which eq. 2.7.11 holds. Since $p = 1/2$, $F_{\{y_1\},\{y_2\},\theta}(S, 1/2) \in [-1/2, 1/2]$, so in this case we can use up to $\lambda = 2$. With this choice we obtain that

$$\min\Big\{B_+(1/2), B_-(1/2)\Big\} \geqslant b\Big(\log(2) - k^{-1}\Big) - \delta(\rho) \geqslant b\Big(\log(2) - 1/10\Big) - \log(100) > 0$$

when $b \geqslant 8$, and even $b \geqslant 7$, if we forget the k^{-1} also (that was contributing to make the bound uniform). So the conclusion is that in this low sample size simple situation, the test of proposition 2.7.3 on page 101 is in some sense exact. If we do not compromise the mathematical properties of the test, keep k^{-1} and take $\delta(\rho) = 63$, we get $b \geqslant 107$, whereas the approximated test gives $b \geqslant 584$.

So a first conclusion is that for small sample sizes, the non approximated test of proposition 2.7.3 on page 101, although not so intuitive, is noticeably sharper than its more understandable approximation given in the next section.

The above discussion was interested in the difference between $\hat{p}(\rho_-)$ and $\hat{p}(\rho_+)$. We may also want to study the ratio $\hat{p}(\rho)/(1 - \hat{p}(\rho))$.

Let us consider again the non approximated test. Equations (2.7.9) and (2.7.10) on the current page give us that the test will reject the hypothesis as soon as, for

some p, there are two contexts ρ_+ and ρ_- such that

$$\frac{a(\rho_+)}{b(\rho_+)-a(\rho_+)} \geqslant \frac{-\log(1-\lambda p)-\frac{\lambda}{k}p+\frac{\delta}{b(\rho_+)-a(\rho_+)}}{\log(1+\lambda(1-p))-\frac{\lambda}{k}(1-p)},$$
$$\frac{b(\rho_-)-a(\rho_-)}{a(\rho_-)} \geqslant \frac{-\log(1-\lambda(1-p))-\frac{\lambda}{k}(1-p)+\frac{\delta}{b(\rho_-)-a(\rho_-)}}{\log(1+\lambda p)-\frac{\lambda}{k}p}.$$

Once again, we may consider only the case $p=\frac{1}{2}$ for the rejection, which has the advantage of symmetrizing both equations: if we suppose also that $b(\rho_+)=b(\rho_-)$, the hypothesis will be rejected as soon as both ratios are greater than

$$\frac{\hat{p}}{1-\hat{p}} \geqslant \frac{-\log\left(1-\frac{\lambda}{2}\right)-\frac{\lambda}{2k}+\frac{\delta}{b(\rho)}}{\log\left(1+\frac{\lambda}{2}\right)-\frac{\lambda}{2k}}.$$

We are now left with the optimization of

$$f(x)=\frac{-\log(1-x)-x/k}{\log(1+x)-x/k}+\frac{\Delta}{\log(1+x)-x/k},$$

for $x \in]0,\frac{1}{2}[$, where $\Delta=\delta/b(\rho)$. We may first simplify it by observing that, if $k=10$, the x/k term may be neglected (at least for the derivative), so argmin f should not be far from

$$x_m=\operatorname{argmin}\frac{-\log(1-x)}{\log(1+x)}+\frac{\Delta}{\log(1+x)}.$$

At this point, we can remark that the minimum is attained for $x>\frac{1}{2}$ if $\Delta>\frac{1}{2}$ (approximately), so we may take the value on $\lambda=1$ in this case, which gives the bound

$$\frac{\hat{p}}{1-\hat{p}} \geqslant 1.81+2.82\Delta.$$

This bound is always true, but gives suboptimal results when $\Delta<\frac{1}{2}$, which is the case when we have more than $2\delta(\rho)$ observations, which is quite probable (remember that a plausible bound for $\delta(\rho)$ is 56).

Using again $b(\rho_+)=b(\rho_-)=1000$, $\Delta=.065$, and computing the actual minimum of the bound function, we get a bound around 1.65 ($\lambda\approx 0.45$), better than the 2 of the suboptimal bound ($\lambda=1$).

Another practical question is to choose a reasonable entropy term $\delta(\rho)$. One could argue heuristically that $\delta(\rho)$ should be taken of the form $\log(\varepsilon^{-1}N)$, where N is the number of Markov substitute pairs and contexts actually tested. For instance, if we take $N = 10^6$, we get that a pair that has been seen $b = 32$ times in disjoint contexts can be rejected as a Markov substitute pair with a confidence level of 98/100.

Another way to think about tuning the value of the entropy term $\delta(\rho)$, is that if we lower it, we are sure to both increase the false rejection rate and decrease the false acceptance rate. Thus, if we want to put the stress on minimizing the false acceptance rate rather than the false rejection rate, lowering $\delta(\rho)$ will probably strike a more favorable balance than the theoretical value tuned to obtain a false rejection rate less than 2ε.

On the other hand, the approximations made to the test in proposition 2.7.5 on page 106 are sharp in the large sample size limit, since we kept the leading term unchanged at each step and made approximations only on second order terms.

2.7.5 Probability of false acceptance of the test

In the following, we are going to study the probability of false acceptance of the test of proposition 2.7.3 on page 101. Let us introduce

$$q(B_1, B_2, \theta) = \mathbb{E}\Big(X_{B_1 \cup B_2} \in \theta, Y_{B_1 \cup B_2} \in B_1 \cup B_2\Big).$$

Lemma 2.7.6
For any $p \in [0,1]$, any $\lambda \in]-1,1[$, any $B_1, B_2 \in \mathscr{B}$, any $\theta \in \Theta$,

$$\mathbb{E}\Big[\exp\Big(-\psi\big(\lambda F_{B_1,B_2,\theta}(S,p)\big)\Big)\Big] \leqslant \exp\Bigg(q(\theta)\Bigg[\lambda\frac{k-1}{k}\big(p-p(\theta)\big) + \frac{\lambda^2}{1-|\lambda|}\left(\frac{k-1}{k}+\frac{\phi(k^{-1})}{2k^2}\right)\Big(p(\theta)\big(1-p(\theta)\big)+\big(p-p(\theta)\big)^2\Big)\Bigg]\Bigg),$$

where $\phi(z) = 2\,z^{-2}\big(\exp(z)-1-z\big)$ is an increasing function such that $\phi(0)=1$, and where we have put for short $p(B_1,B_2,\theta) = p(\theta)$ and $q(B_1,B_2,\theta) = q(\theta)$. This obviously implies that, with probability at least $1-\varepsilon$,

$$\sum_{i=1}^{n}\psi\big(\lambda F_{B_1,B_2,\theta}(S,p)\big) \geqslant -\log\big(\varepsilon^{-1}\big) - nq(\theta)\Bigg[\lambda\frac{k-1}{k}\big(p-p(\theta)\big) + \frac{\lambda^2}{1-|\lambda|}\left(\frac{k-1}{k}+\frac{\phi(k^{-1})}{2k^2}\right)\Big(p(\theta)\big(1-p(\theta)\big)+\big(p-p(\theta)\big)^2\Big)\Bigg].$$

PROOF. Let us remark first that

$$\exp\Big(-\psi(\lambda x)\Big) = \exp\left(\frac{\lambda}{k}x - \log\big(1+\lambda x\big)\right) \leqslant \frac{1+\frac{\lambda}{k}x + \phi\left(\frac{\lambda}{k}\right)\frac{\lambda^2}{2k^2}x^2}{1+\lambda x}$$
$$= 1 - \frac{k-1}{k}\lambda x + \left(\frac{k-1}{k} + \frac{1}{2k^2}\phi\left(\frac{\lambda}{k}\right)\right)\frac{\lambda^2 x^2}{1+\lambda x}$$
$$\leqslant 1 - \frac{k-1}{k}\lambda x + \left(\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right)\frac{\lambda^2}{1-\lambda}x^2, \quad \lambda \in [0,1[, x \in [-1,1].$$

As a consequence, for any $\lambda \in]-1,1[$,

$$\mathbb{E}\Big[\exp\Big(-\psi\big(\lambda F_{B_1,B_2,\theta}(S,p)\big)\Big)\Big]$$
$$\leqslant 1 - \frac{k-1}{k}\lambda\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) + \left(\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right)\frac{\lambda^2}{1-|\lambda|}\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)^2\Big)$$
$$\leqslant \exp\left(-\frac{k-1}{k}\lambda\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)\Big) + \left(\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right)\frac{\lambda^2}{1-|\lambda|}\mathbb{E}\Big(F_{B_1,B_2,\theta}(S,p)^2\Big)\right).$$

The rest follows easily from

$$\mathbb{E}\Big(F_{B_1,B_2,\theta}(p)\Big) = \big(p(\theta)-p\big)q(\theta)$$
$$\mathbb{E}\Big(F_{B_1,B_2,\theta}(p)^2\Big) = \Big(p(\theta)\big(1-p(\theta)\big) + \big(p-p(\theta)\big)^2\Big)q(\theta). \qquad \square$$

We can now state the following proposition:

Proposition 2.7.7

Let us put

$$\delta = \frac{1}{n}\log\Big[k^3n^3\Big(\max_{1\leqslant i\leqslant n}\overline{h}(S_i)\Big)\Big(\max_{1\leqslant i\leqslant n}\overline{g}(S_i)\Big)|\Lambda|\varepsilon^{-2}\Big],$$
$$\chi = \sup_{x\,\in\,[(2n)^{-1/2},\,(2n)^{1/2}]}\ \inf_{\lambda\,\in\,\Lambda}\cosh\Big[\log\Big(\frac{\lambda x}{(1-\lambda)}\Big)\Big],$$
$$a = \frac{4\chi^2 k}{(k-1)}\left(1+\frac{\varphi(k^{-1})}{2k(k-1)}\right) \leqslant 4.47\chi^2 \text{ when } k = 10,$$
$$b = \frac{\big(2+\sqrt{2}\big)\,k}{k-1} \leqslant 3.8 \text{ when } k = 10.$$

Let us assume that there are two sets $B_1, B_2 \in \mathscr{B}$ *and two contexts* $\theta_+, \theta_- \in \Theta$ *such that* $\overline{p}_\epsilon = p(B_1, B_2, \theta_\epsilon)$ *and* $q_\epsilon = q(B_1, B_2, \theta_\epsilon)$, $\epsilon = \pm 1$, *satisfy*

$$\min\{q_-, q_+\} \geqslant \frac{16k\chi^2\delta}{(k-1)}, \qquad \textit{and}$$

$$\overline{p}_+ - \overline{p}_- \geqslant \sqrt{\frac{a\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{a\delta}{q_+}\right) + \frac{b\delta}{q_+} + \sqrt{\frac{a\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1 + \frac{a\delta}{q_-}\right) + \frac{b\delta}{q_-}. \tag{2.7.12}$$

Then with probability at least $1 - 2\varepsilon$

$$\inf_{p\in[0,1]} \max\Big\{B_-(p), B_+(p)\Big\} > 0.$$

Therefore, the test defined by eq. (2.7.5) *on page 101 has in this case a probability of false acceptance of the pair* $(B_1, B_2) \in \mathscr{B}^2$ *as a Markov substitute pair of sets at most equal to* 2ε.

More precisely, with probability at least $1 - 2\varepsilon$,

$$B_-\left(\overline{p}_+ - \sqrt{\frac{a\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{a\delta}{q_+}\right) - \frac{b\delta}{q_+}\right) > 0, \tag{2.7.13}$$

$$B_+\left(\overline{p}_- + \sqrt{\frac{a\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1 + \frac{a\delta}{q_-}\right) + \frac{b\delta}{q_-}\right) > 0. \tag{2.7.14}$$

If we assume now in place of eq. (2.7.12) *on the current page that*

$$\overline{p}_+ - 1 + \eta \geqslant \sqrt{\frac{a\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{a\delta}{q_+}\right) + \frac{b\delta}{q_+},$$

$$\textit{or that } \eta - \overline{p}_- \geqslant \sqrt{\frac{a\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1 + \frac{a\delta}{q_-}\right) + \frac{b\delta}{q_-},$$

$$\textit{or that } \overline{p}_+ - \overline{p}_- \geqslant \gamma + \sqrt{\frac{a\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{a\delta}{q_+}\right) + \frac{b\delta}{q_+} + \sqrt{\frac{a\overline{p}_-(1-\overline{p}_-)\delta}{q_-}}\left(1 + \frac{a\delta}{q_-}\right) + \frac{b\delta}{q_-}.$$

the false acceptance probability of the test (defined in corollary 2.7.4 on page 102) that $(B_1, B_2) \in \mathscr{B}^2$ *is an* γ*-weak* η*-Markov substitute pair of sets will be not greater than* 2ε.

PROOF. The proof is very similar to the one of proposition 2.6.3 on page 92.

Let us put

$$p = \overline{p}_+ - \sqrt{\frac{a\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1 + \frac{a\delta}{q_+}\right) - \frac{b\delta}{q_+}.$$

We will here choose $\lambda \in \Lambda$ such that

$$\lambda \in \arg\min_{\zeta\in\Lambda} \cosh\left[\log\left(\frac{\zeta}{1-\zeta}\sqrt{\left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}}\right)\right].$$

Once again, since $\overline{p}_+ - p \geqslant \dfrac{\delta}{q_+}$,

$$\sqrt{\frac{1}{n}} \leqslant \sqrt{\frac{\delta}{q_+}} \leqslant \sqrt{\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}} \leqslant \sqrt{\frac{q_+}{\delta}} \leqslant \sqrt{n},$$

which implies, together with the fact that $\dfrac{1}{2} \leqslant \left[\dfrac{k-1}{k} + \dfrac{\phi(k^{-1})}{2k^2}\right] \leqslant 2$, that

$$\cosh\left[\log\left(\frac{\lambda}{1-\lambda}\sqrt{\left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}}\right)\right] \leqslant \chi.$$

Now, we have that

$$B_-(p) \geqslant \sum_{i=1}^{n} \psi\Big(\lambda F_{B_1,B_2,\theta_+}(S_i,p)\Big) - \log\Big[k^3|\Lambda|\Big(\max_{1\leqslant i\leqslant n}\overline{h}(S_i)\Big)^2\Big(\max_{1\leqslant i\leqslant n}\overline{g}(S_i)\Big)n^3\varepsilon^{-1}\Big]$$

Together with lemma 2.7.6 on page 111, we get that, with probability at least $1-\varepsilon$

$$B_-(p) \geqslant n\lambda q_+\left(\frac{k-1}{k}(\overline{p}_+ - p) - \frac{\lambda}{1-\lambda}\left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\Big(\overline{p}_+\big(1-\overline{p}_+\big) + \big(p-\overline{p}_+\big)^2\Big) - \frac{\delta}{\lambda q_+}\right).$$

We obtain then that with probability at least $1-\varepsilon$,

$$\begin{aligned}
B_-(p) > nq_+\lambda\Bigg\{ & \frac{k-1}{k}(\overline{p}_+ - p) - \frac{\delta}{q_+} \\
& - 2\sqrt{\left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]\delta}{q_+}} \\
& \quad \times \cosh\left[\log\left(\frac{\lambda}{1-\lambda}\sqrt{\left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]q_+}{\delta}}\right)\right]\Bigg\} \\
& \geqslant nq_+\lambda\Bigg(\frac{k-1}{k}(\overline{p}_+ - p) - \frac{\delta}{q_+} \\
& \qquad - 2\chi\sqrt{\left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\frac{[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]\delta}{q_+}}\Bigg).
\end{aligned}$$

Let us put $\alpha = \dfrac{k-1}{k}$ and $\rho = \dfrac{4\chi^2\delta}{q_+}\left(\alpha + \dfrac{\phi(k^{-1})}{2k^2}\right)$, $v = \overline{p}_+(1-\overline{p}_+)$, $y = \dfrac{\delta}{q_+}$ and $x = \overline{p}_+ - p$. The sign of the right-hand side of the previous inequality is the same as the sign of

$$\begin{aligned}
\left(\frac{k-1}{k}(\overline{p}_+ - p) - \frac{\delta}{q_+}\right)^2 - \left[\frac{k-1}{k} + \frac{\phi(k^{-1})}{2k^2}\right]\frac{4\chi^2\delta[\overline{p}_+(1-\overline{p}_+) + (\overline{p}_+ - p)^2]}{q_+} \\
= \alpha^2x^2 - 2\alpha yx + y^2 - \rho v - \rho x^2 \\
= (\alpha^2 - \rho)x^2 - 2\alpha yx - (\rho v - y^2).
\end{aligned}$$

This quantity is non negative when x is not less than

$$\begin{aligned}
\frac{\alpha y}{\alpha^2-\rho} + \sqrt{\frac{\alpha^2y^2}{(\alpha^2-\rho)^2} + \frac{\rho v - y^2}{\alpha^2-\rho}} &= \frac{\alpha y}{\alpha^2-\rho} + \sqrt{\frac{\rho y^2}{(\alpha^2-\rho)^2} + \frac{\rho v}{\alpha^2-\rho}} \\
\leqslant \sqrt{\frac{\rho v}{\alpha^2-\rho}} + \frac{(\alpha+\sqrt{\rho})y}{\alpha^2-\rho} &\leqslant \sqrt{\frac{\rho v}{\alpha^2}}\left(1 + \frac{\rho}{2\alpha^2(1-\rho\alpha^{-2})}\right) + \frac{(\alpha+\sqrt{\rho})y}{\alpha^2-\rho} \\
&\leqslant \sqrt{\frac{\rho v}{\alpha^2}}\left(1 + \frac{\rho}{\alpha^2}\right) + \frac{(2+\sqrt{2})y}{\alpha},
\end{aligned}$$

where we have used the inequalities $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$ and $\rho \leqslant \alpha^2/2$.

Thus we obtain with probability at least $1-\varepsilon$ that $B_-(p) > 0$ when

$$\overline{p}_+ - p \geqslant \sqrt{\frac{\rho v}{\alpha^2}}\left(1 + \frac{\rho}{\alpha^2}\right) + \frac{(2+\sqrt{2})\delta}{\alpha q_+}.$$

Since $\frac{\rho}{\alpha^2} = \frac{a\delta}{q_+}$ and $\frac{(2+\sqrt{2})}{\alpha} = b$, this proves in particular that with probability at least $1-\varepsilon$,

$$B_-\left(\overline{p}_+ - \sqrt{\frac{a\overline{p}_+(1-\overline{p}_+)\delta}{q_+}}\left(1+\frac{a\delta}{q_+}\right) - \frac{b\delta}{q_+}\right) > 0,$$

The proof of eq. (2.7.14) on page 113 is obtained exactly in the same way, and the rest of the proposition is a straightforward consequence of eqs. (2.7.13) and (2.7.14) on page 113. □

2.8 Estimation of the substitute measure using the results of the test

The family of tests we defined here may be used to get an estimation of the substitute measure q_B of a set that passed the test. More precisely, we can get an estimation $\widehat{p}$ of p such that $\mathbb{E}(F(X,Y,p)) = 0$. These p are usually linked to the substitute measures, $p = q_{\{y_1,y_2\}}(y_1)$ in a pair test, or $p = q_{B_1\cup B_2}(B_1)$ in the union test, for example. Note that in the second example, the symmetric test giving $q_{B_1\cup B_2}(B_2)$ is required to have the substitute measure (and gives us an indication of the size of the intersection $q_{B_1\cup B_2}(B_1 \cap B_2)$ as a bonus).

Let us remark that all the tests we defined were based on two functions $B_+(p)$ and $B_-(p)$, one non-decreasing and one non-increasing, that could be written as

$$B_\pm(p) = \tilde{B}_\pm(p) + \log(\varepsilon),$$

where $\tilde{B}_\pm(p)$ did not depend on ε. Note that while the functions $B_\pm$ where expressed as a supremum on $\rho(\theta)$ and λ, we can simply take in this case the supremum on λ, and $\rho = \delta_{(D^*)^2}$ to maximize the size of the sample. The analysis told us that if the set tested was a Markov substitute set with parameter p, with probability greater than $1-2\varepsilon$,

$$B_\pm(p) < 0,$$

or equivalently,

$$\tilde{B}_\pm(p) < \log(\varepsilon^{-1}) = \ell.$$

Thus, if we fix ℓ, we have, as before, a test $\max(B_+(p), B_-(p)) > 0$. However, we could see this in another light. Indeed, any q such that $B_-(q) > 0$ is smaller than p, or bigger if $B_+(q) > 0$. Studying the graph of $B_\pm$ gives us then a confidence interval for p, if we fix ℓ. The middle of the confidence interval is a possible estimator. However, there is another one we can consider.

If we do not fix ℓ, we can consider the value $\widehat{p}$ such that $\tilde{B}_-(\widehat{p}) = \tilde{B}_+(\widehat{p})$. If we put $\ell_0 = \tilde{B}_+(\widehat{p})$, we would reject the hypothesis if we wanted a confidence of more than $1 - 2e^{\ell_0}$. As such, the parameter $\widehat{p}$ is the only parameter that is in all confidence intervals, for any confidence level that accepts the pair as a Markov substitute set.

We could also use ℓ_0 to sort different candidates, the lower ℓ_0, the most likely they are to be Markov substitute sets.

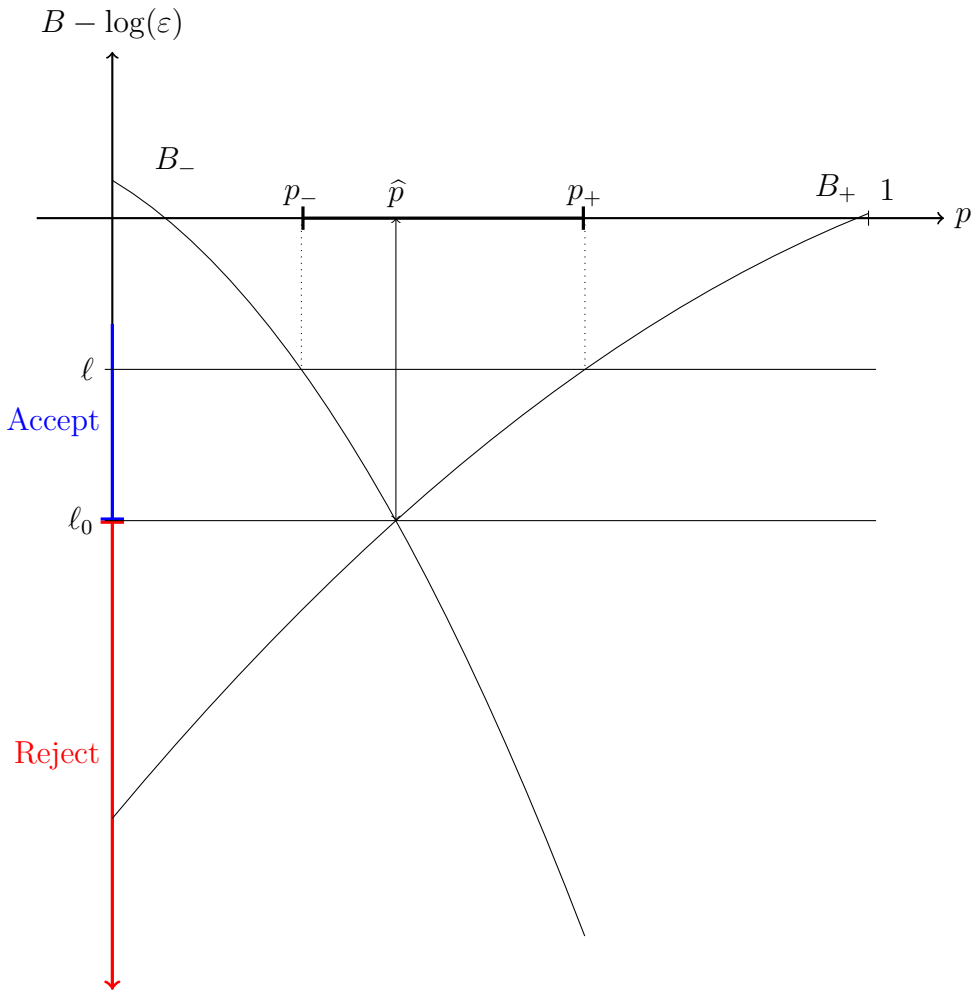


Figure 2.2: Test of the Markov substitute property.

Chapter 3

Markov substitute sets and language

3.1 Production rules and Markov substitute sets

3.1.1 Markov grammars

Consider a set of context-free production rules $\mathscr{R}$ whose terminal symbols belong to an alphabet D and whose non terminal symbols belong to $N = \left\{]_i, i \in \mathbb{N} \setminus \{0\} \right\}$. Let us assume that each rule is of the form $]_i \to e$, where $i \in \mathbb{N} \setminus \{0\}$ and the expression $e \in (D \cup N)^+$, so that no rule uses the empty string.

Production rules alone do not suffice to define a complete context-free grammar, we also need a start symbol. We will here consider multiple start symbols, defining each a different context-free grammar. This approach will enable us to study in a broader sense the properties of our model, and move outside of regular context-free grammars.

Definition 3.1.1
For any set of context-free production rules (ruleset for short) $\mathscr{R}$, and any $i \in \mathbb{N}$, let $\mathscr{R}_i$ be the context-free grammar whose set of rules is $\mathscr{R}$ and whose start symbol is $]_i$.

For any non empty expression $e \in (D \cup N)^+$, let $\mathscr{R}_e$ be the context-free grammar whose start symbol is $]_0$ and whose set of rules is $\mathscr{R} \cup \{]_0 \to e\}$.

For any $i \in \mathbb{N} \setminus \{0\}$, let B_i be the language generated by $\mathscr{R}_i$, and for any expression $e \in (D \cup N)^+$, let B_e be the language generated by $\mathscr{R}_e$. Let us remark that B_i and B_e may be empty, and that this will be the case except for a finite set of values of i and for the expressions e using only the corresponding non terminal symbols. Elements of the set B_i will be called syntagms of type i.

In the following, we will use for the rule $]_i \to e$ the more compact notation $[_i e$, that we already used in chapter 1.

Let us remark that for any $i \in \mathbb{N} \setminus \{0\}$,

$$B_i = B_{]_i} = \bigcup_{e,\, [_i e \in \mathscr{R}} B_e,$$

but that this union is not necessarily a disjoint union.

The context-free property ensures that any elements of a B_i may appear in the same contexts. We will here strengthen this property.

Definition 3.1.2
A ruleset will be called a Markov ruleset if, for any $i \in \mathbb{N} \setminus \{0\}$, B_i is a Markov substitute set.

Lemma 3.1.1
For any Markov ruleset, for any $e \in S^+$, such that $\mathbb{P}\Big(\mathscr{S}(S, B_e) \neq \varnothing\Big) > 0$, B_e is a Markov substitute set.

PROOF. If we write $e = (w_k, 1 \leqslant k \leqslant \ell)$, we can write B_e as

$$B_e = \gamma\left(B_{w_1}, \ldots, B_{w_\ell}\right),$$

and each B_{w_k} is either a singleton (if $w_k \in D$) or $B_{]_i}$, so in any case a Markov substitute set by assumption, so that B_e is also a Markov substitute set, according to proposition 2.1.3 on page 67. □

The substitute measure of a syntagm set B_e is given by the following lemma.

Lemma 3.1.2
Consider any non terminal expression $e = w_{1:k} \in (D \cup N)^k$ of length k, such that

$$\mathbb{P}\Big(\mathscr{S}(S, B_e) \neq \varnothing\Big) > 0.$$

Consider any string $\gamma\left(y_{1:k}\right) \in B_e$, where γ is the concatenation operator, such that $y_j \in B_{w_j}$, $1 \leqslant j \leqslant k$. The substitute measure of B_e can be expressed as

$$q_{B_e}\Big(\gamma\left(y_{1:k}\right)\Big) = A_e \prod_{j=1}^{k} q_{B_{w_j}}(y_j),$$

where the constant A_e is such that for any functions f_j, $1 \leqslant j \leqslant k$,

$$\sum_{y \in B_e} q_{B_e}(y) \sum_{y_j \in B_{w_j}, 1 \leqslant j \leqslant k} \mathbb{1}\Big(y = \gamma\left(y_{1:k}\right)\Big) \prod_{j=1}^{k} f_j(y_j) = A_e \prod_{j=1}^{k} \left(\sum_{y_j \in B_{w_j}} q_{B_{w_j}}(y_j) f_j(y_j) \right).$$

Let us remark that the constant A_e may indeed be different from one, as in the case when $e = {]_1}{]_2}$, where $B_1 = \{a, ab\}$ and $B_2 = \{bc, c\}$. In this case $B_e = \gamma(B_1, B_2) = \{ac, abc, abbc\}$ has only three elements and not four, so that $\sum_{z \in B_e} p(z) = \sum_{z_1 \in B_1, z_2 \in B_2} p(z_1 z_2) - p(abc)$. However, $A_{]_i} = 1$ for all i.

Proof. Let us choose $x \in (D^*)^2$ such that $\mathbb{P}_S\Big(\alpha(x, B_e)\Big) > 0$.

For any $y_{1:k} \in B_{w_1} \times \cdots \times B_{w_k}$,

$$\begin{aligned}\mathbb{P}\Big(S = \alpha\Big(x, \gamma\left(y_{1:k}\right)\Big)\Big) &= \left(\sum_{y'_k \in B_{w_k}} \mathbb{P}_S\Big(\alpha\Big(x, \gamma\left(y_{1:k-1}, y'_k\right)\Big)\Big)\right) q_{B_{w_k}}(y_k) \\ &\vdots \\ &= C_x\, \overline{q}(y_{1:k}).\end{aligned}$$

where

$$C_x = \left(\sum_{y'_{1:k} \in \prod_{j=1}^k B_{w_j}} \mathbb{P}_S\Big(\alpha\Big(x, \gamma\left(y'_{1:k}\right)\Big)\Big)\right) \geqslant \mathbb{P}_S\Big(\alpha(x, B_e)\Big) > 0,$$

$$\text{and} \quad \overline{q}(y_{1:k}) = \left(\prod_{j=1}^k q_{B_{w_j}}(y_j)\right).$$

This shows that $\overline{q}(y_{1:k})$ depends in fact only on $\gamma\left(y_{1:k}\right)$ and that $q_{B_e}\Big(\gamma\left(y_{1:k}\right)\Big)$ is proportional to $\overline{q}(y_{1:k})$, so that for some constant A_e, $q_{B_e} = A_e\, \overline{q}$.

We can then write for any test functions f_j, $1 \leqslant j \leqslant k$,

$$\begin{aligned}&\sum_{y \in B_e} q_{B_e}(y) \sum_{y_j \in B_{w_j}, 1 \leqslant j \leqslant k} \mathbb{1}\Big(y = \gamma\left(y_{1:k}\right)\Big) \prod_{j=1}^k f_j(y_j) \\ &\quad = \sum_{y_{1:k} \in \prod_{j=1}^k B_{w_j}} q_{B_e}\Big(\gamma\left(y_{1:k}\right)\Big) \prod_{j=1}^k f_j(y_j) = \sum_{y_{1:k} \in \prod_{j=1}^k B_{w_j}} A_e \overline{q}(y_{1:k}) \prod_{j=1}^k f_j(y_j) \\ &\quad = A_e \left(\prod_{j=1}^k \sum_{y_j \in B_{w_j}} q_{B_{w_j}}(y_j) f_j(y_j)\right). \qquad \square\end{aligned}$$

This lemma will be used in section 3.2.2 on page 127 to build an estimator of the substitute measures, but we need to introduce first some other notions.

3.1.2 Parse trees

This definition of grammars and language gives production and parsing methods, but the structure of a string is not readily apparent. To manipulate this structure, we will introduce the notion of parse trees.

Definition 3.1.3
For any ruleset $\mathscr{R}$, let $\breve{\mathscr{R}}$ be the ruleset

$$\breve{\mathscr{R}} = \left\{ \left[_i \left(_i e \right)_i, \left[_i e \in \mathscr{R} \right\}. \right.$$

Let $\breve{D} = D \cup \left\{ \left(_i, \right)_i, i \in \mathbb{N} \right\}$ be the related extended set of terminal symbols.

Similarly to definition 3.1.1 on page 119, let $\breve{\mathscr{R}}_i$ be the context-free grammar with start symbol $\left]_i\right.$, terminal symbol set $\breve{D}$ and set of rules $\breve{\mathscr{R}}$. In the same way, let $\breve{\mathscr{R}}_e$ be the grammar with start symbol $\left]_0\right.$, terminal symbol set $\breve{D}$ and set of rules $\breve{\mathscr{R}} \cup \{\left[_0 e\right\}$.

Let $\breve{B}_i$ be the language generated by $\breve{\mathscr{R}}_i$.

We can in the same way define $\breve{B}_e$, the language generated by $\breve{\mathscr{R}}_e$, for any expression $e \in (D \cup N)^+$. Let us remark that, again, $\breve{B}_j = \breve{B}_{\left]_j\right.}$.

Let $\mathscr{T} = \bigcup_{e \in (D \cup N)^+} \breve{B}_e$ the set of parse trees. When $e \neq e'$, $\breve{B}_e$ and $\breve{B}_{e'}$ are disjoint. We can therefore, for any parse tree $t \in \mathscr{T}$ define its surface structure $\varsigma(t) \in (D \cup N)^+$ by the condition $t \in \breve{B}_{\varsigma(t)}$. It is obtained by replacing each outer pair $\left(_j \cdots \right)_j$ of matched parentheses in t and its content by $\left]_j\right.$.

Members of a particular $\breve{B}_i$ will be called parse trees of type i.

The realization of a parse tree is given by the operator $\varphi : \mathscr{T} \to D^+$ which removes parentheses.

We will say that a tree $t \in \mathscr{T}$ is a parse of another t' (noted $t \succcurlyeq t'$), if they have same realizations $\varphi(t) = \varphi(t')$, and all parentheses in t' are also in t.

The realization operator is such that $\varphi\left(\breve{B}_i\right) = B_i$. It is however not one to one, since a given string may have different parses.

These new grammars will be called for short tree grammars, and will be used to describe the parse trees. Parse trees of a given type describe the parses of the syntagms of same type.

Let us remark that for any $t \in \breve{B}_i$, $\varsigma(t) = \left]_i\right.$, and $t = \left(_i y\right)_i$, where $y \in \mathscr{T}$ is such that $\left[_i \varsigma(y) \right. \in \mathscr{R}$.

In the same way, for any $t \in \mathscr{T}$, and any $x \in \left(\breve{D}^*\right)^2$, $y \in \mathscr{T}$ and $j \in \mathbb{N} \setminus \{0\}$, such that $t = \alpha\left(x, \left(_j y\right)_j\right)$, $\left[_j \varsigma(y) \right. \in \mathscr{R}$. This is because the condition $y \in \mathscr{T}$ imposes that in the expression $\left(_j y\right)_j$ the outer parentheses $\left(_j \right)_j$ are matched.

3.1.3 General introduction to parsing

The next step is to construct parsing trees from a string.

Definition 3.1.4
A parsing kernel is a probability kernel $\Pi : D^+ \mapsto \mathscr{T}$ *such that*

$$\mathrm{supp}(\Pi(s, \cdot)) \subset \varphi^{-1}(s).$$

A parsing kernel for a certain syntagm set B_e *is a probability kernel* $\Pi_e : B_e \mapsto \breve{B}_e$ *such that*

$$\mathrm{supp}(\Pi_e(s, \cdot)) = \varphi^{-1}(s) \cup \breve{B}_e.$$

Parsing kernels can be constructed using classical parsing algorithms for context-free grammars such as CYK, as discussed in section 3.B on page 159. They provide a way to build the splitting kernels π used in sections 2.2.1 and 2.2.5 on page 69 and on page 73, by simply drawing at random one or two matching pairs of parentheses in the parsed sentence or text.

A parsing kernel builds a tree from a sentence that is compatible with a ruleset. This operation does not lose any information, and adds a syntactic structure to the sentence. However, the manipulation of natural language corpora may be quite difficult due to its sheer size. For this reason, it may be interesting to find some sort of process that can reduce the size of sentences, while losing as little information as possible. One way to do this is to actually only keep the syntactic structure of a given sentence, by considering the surface structure of a parse of the sentence.

Definition 3.1.5
A parsing reduction kernel is an kernel r *from* D^+ *to* S^+ *such that*

$$r(s, e) > 0 \implies s \in B_e.$$

Given a parsing kernel Π*, we can define a parsing reduction kernel* r *as*

$$r(s, e) = \sum_{t : \varsigma(t) = e} \Pi(s, t).$$

In order to implement the reversible dynamics we have defined, we need to have Markov kernels for which both simulating transitions and computing transition probabilities are fast enough. A detailed description of these two operations is provided in section 3.D on page 165. In the following sections, we will use parsing kernels to define the splitting kernel of our Metropolis reversible dynamics. While simulation and computation are both easy when parsing in a given syntagm set, the general parse is more tricky. We will have to access the history of the parse,

and manipulate strings of syntactic trees. This section introduces the required definitions to achieve this, while the actual parsing schemes, based on the CYK algorithm, are described in section 3.B on page 159.

Let us note b and e the begin and end operators on strings, $b(s_{1:n}) = s_1$, $e(s_{1:n}) = s_n$.

We will build the parse using extension kernels, that add one pair of parentheses at a time.

Definition 3.1.6
A parsing extension is a kernel

$$\Big\{\varpi(t,t'), t, t' \in \mathscr{T}, t' \succcurlyeq t\Big\}.$$

A parsing extension relative to a reference tree T is a kernel

$$\Big\{\rho_T(t,t'), t, t' \in \mathscr{T}, T \succcurlyeq t' \succcurlyeq t\Big\}.$$

We will usually consider bottom-up parsing extensions, that is, that add one pair of outer parentheses at a time,

$$\Big\{\varpi(t,t') : t, t' \in \mathscr{T}, \exists i \in \mathbb{N}, x \in \mathscr{T}, a \in S^+, [_i a \in \mathscr{R}, y \in \breve{B}_a, \\ t = \alpha(x,y), t' = \alpha(x, ({}_i y)_i)\Big\}.$$

Parsing extensions define chains of trees, the end of which is our final parse.

Definition 3.1.7
Any parsing extension ϖ defines a stopped Markov chain T_n on parse trees, its stopping time being

$$\tau = \inf\{n, \mathrm{supp}(\varpi(T_n, \cdot)) = \varnothing\}.$$

A general parsing kernel will then be a kernel $\overline{\Pi}$ from syntactic trees (in practice, strings), to sequences of parsing trees.

$$\overline{\Pi}(t,\tilde{t}) = \mathbb{1}(\tilde{t}_0 = t)\mathbb{1}\Big(\mathrm{supp}(\varpi(e(\tilde{t}), \cdot)) = \varnothing\Big) \prod_{i=0}^{\ell(\tilde{t})-1} \varpi(\tilde{t}_i, \tilde{t}_{i+1}), \qquad \tilde{t} \in \mathscr{T}^+.$$

The "final" parse being, of course, $e(\tilde{t})$.

In the case of parsing extensions relative to a tree, we define the kernels

$$\zeta(t,\tilde{t}) = \mathbb{1}(e(\tilde{t}) = t)\mathbb{1}(b(\tilde{t}) = \varphi(t)) \prod_{i=0}^{\ell(\tilde{t})-1} \rho_t(\tilde{t}_i, \tilde{t}_{i+1}), \qquad \tilde{t} \in \mathscr{T}^+.$$

These parsings will be used to define reversible dynamics as described in section 3.D on page 165. This method supposes that we have, for each kernel π used in the simulation, a "symmetric" kernel π', so that the quotient $\frac{\pi(t,t')}{\pi'(t',t)}$ is as close to 1 as possible. In this case, we start from a string s, and build a sequence of increasingly bigger parses $\tilde{t}_i$ using a parsing extension. Conversely, we want, from a parsing tree t, to build a sequence of smaller parsing trees $\tilde{t}_i$, ending at $\varphi(t)$.

Building "parsing reduction kernels"

$$\Big\{\rho(t,t'), t, t' \in \mathscr{T}, t \succcurlyeq t'\Big\}$$

that remove parentheses from an initial tree would be an intuitive way to achieve this, but we can remark that since we know the end part of the tree sequence $(\varphi(t))$, we can actually begin from the end result, and use regular parsing extensions, the only additional requirement being that we finish at the fixed t. This is the motivation behind the definition of parsing extensions relative to a tree.

Actual construction of these parsing extensions, and discussion of the quotient $\frac{\varpi}{\rho}$ will be found in section 3.B on page 159.

3.2 Substitute measures and reversible dynamics

3.2.1 Parametrization of substitute measures

Of course, Markov substitute sets are mostly interesting in conjunction with their substitute measures q_B. The proof of lemma 3.1.1 on page 120 hints that this requires to compute q_{B_i}. We will now see how to perform this computation from the estimation of a few parameters — one for each production rule — using a parse tree structure.

Lemma 3.2.1
For any $s \in D^+$, and any $t \in \mathscr{T}$ such that $s = \varphi(t)$,

$$q_{B_{\varsigma(t)}}(s) = A_{\varsigma(t)} \prod_{\lfloor_j e \in \mathscr{R}} \Big[A_e q_{B_j}\Big(B_e\Big)\Big]^{\chi(t,j,e)},$$

where

$$\chi(t,j,e) = \sum_{x \in (\check{D}^*)^2} \sum_{y \in \mathscr{T}} \mathbb{1}\Big[t = \alpha\Big(x, ({}_j y)_j\Big)\Big] \mathbb{1}\Big(\varsigma(y) = e\Big). \tag{3.2.1}$$

Proof. The result is equivalent to the fact that for any $t \in \mathscr{T}$

$$q_{B_{\varsigma(t)}}\Big(\varphi(t)\Big) = A_{\varsigma(t)} \prod_{\lfloor_j e \in \mathscr{R}} \Big[A_e q_{B_j}\Big(B_e\Big)\Big]^{\chi(t,j,e)},$$

We will prove it by induction on the length of t. If $\ell(t) = 1$, then $t \in D$, $\varsigma(t) = \varphi(t) = t$, and $q_{B_{\varsigma(t)}}\big(\varphi(t)\big) = q_{\{t\}}(t) = 1$, so that the result holds in this case.

Let $\varsigma(t) = \gamma(w_{1:\ell}) \in (D \cup N)^\ell$. We can write the tree t as $t = \gamma(t_{1:\ell})$, where $t_k \in \breve{B}_{w_k}$. With these notations, $w_k = \varsigma(t_k)$, $\varphi(t) = \gamma(\varphi(t_{1:\ell}))$, and $\varphi(t_k) \in B_{\varsigma(t_k)}$. According to lemma 3.1.2 on page 120 this implies that

$$q_{B_{\varsigma(t)}}\big(\varphi(t)\big) = \prod_{k=1}^{\ell} A_{\varsigma(t)} q_{B_{\varsigma(t_k)}}\big(\varphi(t_k)\big). \tag{3.2.2}$$

If $w_k = \varsigma(t_k) \in D$, then $t_k = w_k$, and as already seen, $q_{B_{\varsigma(t_k)}}(\varphi(t_k)) = 1$.

In the case when $w_k \in N$, $\varsigma(t_k) = \,]_i$ and $t_k = ({}_i y_k)_i$, where $\big[_i\varsigma(y_k) \in \mathscr{R}$. Consequently $\varphi(t_k) = \varphi(y_k) \in B_{\varsigma(y_k)} \subset B_{\varsigma(t_k)}$. Therefore

$$q_{B_{\varsigma(t_k)}}\big(\varphi(t_k)\big) = q_{B_{\varsigma(t_k)}}\big(B_{\varsigma(y_k)}\big) q_{B_{\varsigma(y_k)}}\big(\varphi(y_k)\big).$$

Moreover, since $\varsigma(t_k)$ is of length one, $A_{\varsigma(t_k)} = 1$. Since $\ell(y_k) \leqslant \ell(t) - 2 < \ell(t)$, we can now apply the induction hypothesis to y_k, to get

$$\begin{aligned} q_{B_{\varsigma(t_k)}}\big(\varphi(t_k)\big) &= A_{\varsigma(y_k)} q_{B_{\varsigma(t_k)}}\big(B_{\varsigma(y_k)}\big) \prod_{\big[_j e \in \mathscr{R}} A_e q_{B_j}(B_e)^{\chi(y_k, j, e) + \mathbb{1}(j = i, e = \varsigma(y_k))} \\ &= \prod_{\big[_j e \in \mathscr{R}} A_e q_{B_j}(B_e)^{\chi(t_k, j, e)}. \end{aligned}$$

This proves the result for t_k, $1 \leqslant k \leqslant \ell$, (since $A_{\varsigma(t_k)} = 1$) and implies that

$$q_{B_{\varsigma(t)}}\big(\varphi(t)\big) = A_{\varsigma(t)} \prod_{\big[_j e \in \mathscr{R}} \Big[A_e\, q_{B_j}(B_e)\Big]^{\sum_{k=1}^{\ell} \chi(t_k, j, e)},$$

proving the result for t, since

$$\chi(t, j, e) = \sum_{k=1}^{\ell} \chi(t_k, j, e).$$

□

This lemma means that we only need to know the parameters $q_{B_j}(B_e)$, and A_e, for $\big[_j e \in \mathscr{R}$, to know all substitute measures. These parameters can effectively be stored with the ruleset by adding weights to each element $\big[_j e \in \mathscr{R}$.

Let us remark that the knowledge of these measures reduces the estimation of the distibution $\mathbb{P}_S$ to the estimation of the probability of syntagm sets according to the relation

$$\mathbb{P}_S(s) = \mathbb{P}\Big(S \in B_{\varsigma(t)}\Big) q_{B_{\varsigma(t)}}(s), \qquad s \in \varphi(t).$$

3.2.2 Estimating substitute measures

We saw that estimating the substitute measures q_{B_i} can be performed through the estimation of the family of coefficients $\Big(A_e, q_{B_i}(B_e), i \in \mathbb{N} \setminus \{0\}, [_i e \in \mathscr{R}_i\Big)$.

Let us start with the estimation of the constants A_e, where $e = w_{1:k} \in (D \cup N)^k$. We will estimate separately

$$N_e = \sum_{y \in B_e} q_{B_e}(y) \sum_{y_j \in B_{w_j}, 1 \leqslant j \leqslant k} \mathbb{1}\Big(y = \gamma\left(y_{1:k}\right)\Big) \prod_{j=1}^{k} f(y_j),$$

where $f(y) = r(1-r)^{\ell(y)-1}$, $y \in D^+$, for some parameter $r \in (0,1)$, and each

$$D_{w_j} = \sum_{y_j \in B_{w_j}} q_{B_{w_j}}(y_j) f(y_j),$$

to estimate in the end A_e using the formula

$$A_e = \frac{N_e}{\prod_{j=1}^{k} D_{w_j}}.$$

Let us remark first that for any $y \in B_e$,

$$\begin{aligned}\sum_{y_j \in B_{w_j}, 1 \leqslant j \leqslant k} \mathbb{1}\Big(y = \gamma\left(y_{1:k}\right)\Big) \prod_{j=1}^{k} f(y_j) &= \sum_{y_j \in B_{w_j}, 1 \leqslant j \leqslant k} \mathbb{1}\Big(y = \gamma\left(y_{1:k}\right)\Big) r^k (1-r)^{\ell(y)-k} \\ &\leqslant \binom{\ell(y)-1}{k-1} r^k (1-r)^{\ell(y)-k} \leqslant r.\end{aligned}$$

This inequality motivated our choice of test function.

To estimate N_e, we may consider the test functions

$$\begin{aligned}F_{\theta,e}(s,p) = \sum_{x \in \theta} \sum_{y \in B_e} \mathbb{1}\Big(s = \alpha(x,y)\Big) \nu_\theta(x) \\ \times \left[\sum_{y_j \in B_{w_j}, 1 \leqslant j \leqslant k} \mathbb{1}\Big(y = \gamma\left(y_{1:k}\right)\Big) r^k (1-r)^{\ell(y)-k} - p \right].\end{aligned}$$

They are such that $|F_{\theta,e}(s,p)| \leqslant 1$ and

$$\mathbb{E}\Big[F_{\theta,e}(S,p)\Big] = \sum_{x \in \theta} \nu_\theta(x) \mathbb{P}_S\Big(\alpha(x, B_e)\Big)\Big(N_e - p\Big).$$

Therefore, for any θ such that

$$\sum_{x \in \theta} \mathbb{P}\Big(S \in \alpha(B_e)\Big) > 0,$$

N_e is the only value of $p \in [0,1]$ such that $\mathbb{E}\Big[F_{\theta,e}(S,p)\Big] = 0$.

To estimate each D_{w_j}, we may consider the test functions

$$G_{\theta,w_j}(s,p) = \sum_{x\in\theta}\sum_{y\in B_{w_j}} \mathbb{1}\Big(s = \alpha(x,y)\Big)\nu_\theta(x)\Big[r(1-r)^{\ell(y)-1} - p\Big].$$

They are such that $|G_{\theta,w_j}(s,p)| \leqslant 1$ and

$$\mathbb{E}\Big[G_{\theta,w_j}(S,p)\Big] = \sum_{x\in\theta}\mathbb{P}_S\Big(\alpha(x,B_{w_j})\Big)\Big(D_{w_j} - p\Big).$$

Therefore, for any θ such that

$$\sum_{x\in\theta}\mathbb{P}\Big(S \in \alpha(x,B_{w_j})\Big) > 0,$$

D_{w_j} is the only value of $p \in [0,1]$ such that $\mathbb{E}\Big[G_{\theta,w_j}(S,p)\Big] = 0$.

Now that we have explained how to estimate A_e, let us come to the estimation of $\Big(q_{B_i}(B_e), i \in \mathbb{N}\setminus\{0\}, \lfloor_i e \in \mathscr{R}\Big)$.

Let us choose $\theta \in \Theta$ such that

$$\sum_{x\in\theta}\mathbb{P}\Big(S \in \alpha(x,B_i)\Big) > 0,$$

and consider for any $\lfloor_i e \in \mathscr{G}_i$ the test function defined as

$$F_{i,e,\theta}(s,p) = \sum_{x\in\theta}\sum_{y\in D^+} \mathbb{1}\Big(s = \alpha(x,y)\Big)\nu_\theta(x)\Big[\mathbb{1}(y\in B_e) - p\,\mathbb{1}(y\in B_i)\Big].$$

Then

$$\begin{aligned}\mathbb{E}\Big[F_{i,e,\theta}(S,p)\Big] &= \sum_{x\in\theta}\nu_\theta(x)\Big(\mathbb{P}_S\Big(\alpha(x,B_e)\Big) - p\,\mathbb{P}_S\Big(\alpha(x,B_i)\Big)\Big)\\ &= \sum_{x\in\theta}\nu_\theta(x)\mathbb{P}_S\Big(\alpha(x,B_i)\Big)\Big(q_{B_i}(B_e) - p\Big),\end{aligned}$$

so that $q_{B_i}(B_e)$ is the only value of p such that $\mathbb{E}\Big[F_{i,e,\theta}(S,p)\Big] = 0$. As such, the estimator $\widehat{p}$ such that $F_{i,e,\theta}(s,\widehat{p}) = 0$ is without bias.

Let us also remark that if we assume only that B_e are Markov substitute sets for all expressions e such that $\lfloor_i e \in \mathscr{R}$, then B_i is itself a Markov substitute set if and only if for any $\lfloor_i e \in \mathscr{G}$, there is $p_{i,e} \in [0,1]$ such that for any $\theta \in \Theta$, $\mathbb{E}\Big[F_{i,e,\theta}\Big(S,p_{i,e}\Big)\Big] = 0$. This can serve to build statistical tests to validate the addition of new rules to the set of rules $\mathscr{R}$. But we will study these in more details in section 3.4.1 on page 144.

3.2.3 Simulating substitute measures

The next question is the simulation of the substitute measure q_{B_e}, for some expression $e \in S^+$.

Let us consider a parsing kernel $\Big(\Pi_e(s,t), s \in B_e, t \in \breve{B}_e\Big)$ for the syntagm set B_e, as defined in definition 3.1.4 on page 123.

Let us define for any $[_i y \in \mathscr{R}$

$$p_i(y) = \frac{A_y q_{B_i}(B_y)}{\sum\limits_{[_i z \in \mathscr{R}} A_z q_{B_i}(B_z)} \leqslant q_{B_i}(B_y).$$

Consider the subprobability measure on $\breve{B}_e$ defined as

$$p_e(t) = \mathbb{1}\Big(\varsigma\,(t) = e\Big) \prod_{[_j y \in \mathscr{R}} p_j(y)^{\chi(t,j,y)},$$

where χ is defined as in eq. (3.2.1) on page 125. To see that this is indeed a subprobability measure, see lemma 3.2.2 on the following page.

Starting from the obvious equality

$$\sum_{t \in \breve{B}_e} p_e(t) \frac{q_{B_e}(\varphi\,(t))\Pi_e(s,t)}{p_e(t)} = q_{B_e}(s), \quad s \in B_e,$$

we see that if we draw $T \in \breve{B}_e \cup \{\mathfrak{e}\}$ according to the subprobability p_e completed with $p_e(\mathfrak{e}) = 1 - p_e\Big(\breve{B}_e\Big)$, we obtain that for any bounded function f,

$$\sum_{s \in B_e} q_{B_e}(s) f(s) = \mathbb{E}\Big(w(T) f(T)\Big),$$

where

$$\begin{aligned} w(T) &= \mathbb{1}\Big(T \neq \mathfrak{e}\Big) \frac{q_{B_e}\Big(\varphi\,(T)\Big)\Pi_e\Big(\varphi\,(T), T\Big)}{p_e(T)} \\ &= \mathbb{1}\Big(T \neq \mathfrak{e}\Big)\Pi_e\Big(\varphi\,(T), T\Big) A_e \prod_{j \in \mathbb{N}\setminus\{0\}} \Bigg(\sum_{y, [_j y \in \mathscr{R}} A_y q_{B_j}(B_y)\Bigg)^{\sum\limits_{y, [_j y \in \mathscr{R}} \chi(T,j,y)}. \end{aligned}$$

The simulation of T according to p_e may be implemented in the following way. We start with $e_0 = e$. Given $e_k \in S^+$,

– if $e_k \in \mathscr{T}$, put $T = e_k$,

– otherwise $e_k = \alpha(x,]_i)$, $i \in \mathbb{N}$, $x \in \breve{D}^* \times S^+$. Draw y according to p_i, completed with $p_i(\epsilon) = 1 - p_i(\breve{B}_e)$, and put $e_{k+1} = \alpha(x, ({}_i y)_i)$, when $y \neq \epsilon$ and set $T = \epsilon$ otherwise.

Lemma 3.2.2
For any $k \in \mathbb{N}$, any well-parenthetized expression $t \in (\breve{D} \cup N)^*$, if we call $|t|$ the number of opening parentheses in t,

$$\mathbb{P}(e_k = t) = \mathbb{1}(\varsigma(t) = e)\mathbb{1}(|t| = k) \prod_{[_j y \in \mathscr{R}} p_j(y)^{\chi(t,j,y)}.$$

As such, if we define $\tau = \inf k, e_k \in \mathscr{T}$,

$$\mathbb{P}(e_\tau = t) = p_e(t).$$

PROOF. By induction on k:

– Obviously $\chi(e, j, y) = 0$ for all y, j, and

$$\mathbb{P}(e_0 = t) = \mathbb{1}(t = e) = \mathbb{1}(\varsigma(t) = e)\mathbb{1}(|t| = 0).$$

– For any $t \in (\breve{D} \cup N)^*$, suppose we can write $t = \alpha(x, ({}_i y)_i)$, with $x_2 \in S^+$. Otherwise t has no parentheses and $\mathbb{P}(e_{k+1} = t) = 0$.

$$\begin{aligned}
\mathbb{P}(e_{k+1} = t) &= \mathbb{P}(e_k = \alpha(x,]_i)) \cdot p_i(y) \\
&= \mathbb{1}(\varsigma(\alpha(x,]_i)) = e)\mathbb{1}(|\alpha(x,]_i)| = k) \\
&\qquad \times \prod_{[_j z \in \mathscr{R}} p_j(z)^{\chi(\alpha(x,]_i), j, z) + \mathbb{1}(j=i, z=y)} \\
&= \mathbb{1}(\varsigma(t) = e)\mathbb{1}(|t| = k+1) \prod_{[_j z \in \mathscr{R}} p_j(z)^{\chi(t,j,z)}.
\end{aligned}$$

□

This lemma proves that p_e is in fact a probability distribution if the process defined previously finishes in finite time, in which case we can draw T using this method. The process does not finish in finite times if there are looping rewriting rules with probability above some critical value. This will not happen if the parameters are accurately estimated from a true Markov substitute process.

An alternative to the use of importance sampling is the use a reversible dynamics. Let us introduce the conditional probability kernel $\Big(k_e(s, s'), s, s' \in B_e\Big)$, defined for any $s \neq s'$ as

$$k_e(s, s') = \sum_{t,t' \in \breve{B}_e} \Pi_e(s, t) p_e(t') w(t, t') \mathbb{1}\Big(s' = \varphi(t')\Big),$$

where

$$w(t,t') = \left(1 \wedge \frac{\Pi_e\Big(\varphi\,(t'),t'\Big)}{\Pi_e\Big(\varphi\,(t),t\Big)} \cdot \frac{q_{B_e}(\varphi\,(t'))}{p_e(t')} \cdot \frac{p_e(t)}{q_{B_e}(\varphi\,(t))}\right)$$
$$= \left(1 \wedge \frac{\Pi_e\Big(\varphi\,(t'),t'\Big)}{\Pi_e\Big(\varphi\,(t),t\Big)} \prod_{j\in\mathbb{N}\setminus\{0\}} \left(\sum_{\lfloor_j y\in\mathscr{R}} A_y q_{B_j}(B_y)\right)^{\sum_{\lfloor_j y\in\mathscr{R}} \chi(t',j,y)-\chi(t,j,y)}\right).$$

To draw s' according to $k_e(s,\cdot)$, we first draw t according to $\Pi_e(s,t)$, then we draw (independently !) t' according to $p_e(t')$ (again, we suppose that the process finishes in finite time), and we set $s' = \varphi\,(t')$ with probability $w(t,t')$, and $s' = s$ otherwise.

Proposition 3.2.3
The kernel $k_e(s,s')$ is reversible with respect to q_{B_e} and irreducible on B_e.

Proof. It is irreducible because obviously $\mathrm{supp}\Big(k_e(s,\cdot)\Big) = B_e$ already. To see that it is reversible we can write

$$q_{B_e}(s)k_e(s,s') = \sum_{t,t'\in\check{B}_e} \Big(q_{B_e}(s)\Pi_e(s,t)p_e(t') \wedge q_{B_e}(s')\Pi_e(s',t')p_e(t)\Big).$$

This expression is symmetric in s and s', since it is of the form

$$q_{B_e}(s)k_e(s,s') = \sum_{t,t'} f(s,t,t',s'),$$

where $f(s,t,t',s') = f(s',t',t,s)$. □

3.2.4 Reversible dynamics for the language distribution

The same method can be used to build reversible dynamics for the language distribution on the whole, simply by parsing a sentence in general, allowing any surface structure. Let us consider a general parsing kernel

$$\Big(\overline{\Pi}(s,t), s\in D^+, t\in\mathscr{T}\Big)$$

such that $\mathrm{supp}(\Pi(s,\cdot)) \subset \varphi^{-1}\,(s)$, and a converse kernel ζ from syntactic trees to sequences of parsing trees. Consider now the conditional probability kernel $\Big(k(s,s'), s,s'\in D^+\Big)$ defined for $s\neq s'$ as

$$k(s,s') = \sum_{\substack{\tilde{t},\tilde{t}'\in\mathscr{T}^+\\ t,t'\in\mathscr{T}}} \overline{\Pi}(s,\tilde{t})\mathbb{1}(t=\tilde{t}_{\ell(\tilde{t})})p_{\varsigma(t)}(t')\zeta(t',\tilde{t}')w(\tilde{t},t,t',\tilde{t}')\mathbb{1}\Big(s'=\varphi\,(t')\Big),$$

where

$$w(\tilde{t},t,t',\tilde{t}') = \left(1 \wedge \frac{\overline{\Pi}\Big(\varphi\,(t'),\tilde{t}'\Big)}{\zeta\Big(\tilde{t}',\varphi\,(t')\Big)} \cdot \frac{\zeta\Big(\tilde{t},\varphi\,(t)\Big)}{\overline{\Pi}\Big(\varphi\,(t),\tilde{t}\Big)} \cdot \frac{q_{B_{\varsigma(t')}}(\varphi\,(t'))}{p_{\varsigma(t')}(t')} \cdot \frac{p_{\varsigma(t)}(t)}{q_{B_{\varsigma(t)}}(\varphi\,(t))}\right)$$
$$= \left(1 \wedge \frac{\overline{\Pi}\Big(\varphi\,(t'),\tilde{t}'\Big)}{\zeta\Big(\tilde{t}',\varphi\,(t')\Big)} \cdot \frac{\zeta\Big(\tilde{t},\varphi\,(t)\Big)}{\overline{\Pi}\Big(\varphi\,(t),\tilde{t}\Big)}\right.$$
$$\left. \times \prod_{j\in\mathbb{N}\setminus\{0\}} \left(\sum_{\lfloor_j y\in\mathscr{R}} A_y q_{B_j}(B_y)\right)^{\sum\limits_{\lfloor_j y\in\mathscr{R}} \chi(t',j,y)-\chi(t,j,y)}\right).$$

Proposition 3.2.4
The kernel k is reversible with respect to the language distribution $\mathbb{P}_S$.

PROOF. We can use the identity $\mathbb{P}_S\Big(\varphi\,(t)\Big) = \mathbb{P}(S \in B_{\varsigma(t)}) q_{B_{\varsigma(t)}}\Big(\varphi\,(t)\Big)$ to remark that

$$\mathbb{P}_S(s)k(s,s') = \sum_{\substack{\tilde{t},\tilde{t}'\in\mathscr{T}^+\\ t,t'\in\mathscr{T}}} \mathbb{P}\Big(S \in B_{\varsigma(t)}\Big)\mathbb{1}\Big(\varsigma\,(t) = \varsigma\,(t')\Big)$$
$$\mathbb{1}\Big(s = \varphi\,(t)\Big)\mathbb{1}\Big(s' = \varphi\,(t')\Big)\mathbb{1}(t = \tilde{t}_{\ell(\tilde{t})})\mathbb{1}(t' = \tilde{t}'_{\ell(\tilde{t}')})$$
$$\times \Big(\overline{\Pi}\Big(\varphi\,(t),t\Big)\zeta\Big(\tilde{t}',\varphi\,(t')\Big)p_{\varsigma(t')}(t')q_{B_{\varsigma(t)}}\Big(\varphi\,(t)\Big)$$
$$\wedge\, \overline{\Pi}\Big(\varphi\,(t'),t'\Big)\zeta\Big(\tilde{t},\varphi\,(t)\Big)p_{\varsigma(t)}(t)q_{B_{\varsigma(t')}}\Big(\varphi\,(t')\Big)\Big),$$

where the symmetry in (s,s') can be seen in the same way as for k_e above. □

This construction improves on the reversible dynamics of section 2.2.1 on page 69, inasmuch as it performs more substitutions at each step, allowing to substitute a sentence in B_e with a randomly chosen other one in one shot. It can also be seen as an implementation of the first dynamics using the extended Markov substitute set B_e. However, this method still requires the knowledge of q_{B_i}, and requires that the production process will terminate in finite time, something that will happen if the parameters are accurately estimated from a true Markov substitute process, but may fail if parameter estimation is not precise enough or if the data are off model. We will see later in section 3.3 on page 135 how to deal with those termination problems by restricting to a finite state space and using a different kind of estimator (crossing-over dynamics on a replicated sample).

We can now remark that the invariant dynamics k has the following context free property.

Proposition 3.2.5
Let us put $\overline{B}_e = \mathrm{supp}\Big(q_{B_e} k^\infty\Big)$. *Then* k *is reversible with respect to* $q_{\overline{B}_e}$.

Proof. To prove this, we will first prove the following lemma.

Lemma 3.2.6
For any $s, s' \in D^+$ *such that* $k(s,s') > 0$, $\{s, s'\}$ *is a Markov substitute pair and*

$$q_{\{s,s'\}}(s)k(s,s') = q_{\{s,s'\}}(s')k(s',s).$$

Indeed, when $k(s,s') > 0$, there is an expression e such that $s, s' \in B_e$, given by the surface structure of the parse tree t of s drawn according to $\Pi(s,\cdot)$. Thus $\{s,s'\}$ forms a Markov substitute pair and

$$q_{\{s,s'\}}(s)k(s,s') = q_{\{s,s'\}}(s')k(s',s). \tag{3.2.3}$$

The lemma implies that for any $s \in B_e$, $\mathrm{supp}\Big(\delta_s k^\infty\Big)$ is a Markov substitute set, and, since $s, s' \in B_e$ implies $k(s,s') > 0$, that

$$\bigcup_{s \in B_e} \mathrm{supp}\Big(\delta_s k^\infty\Big) = \mathrm{supp}\Big(q_{B_e} k^\infty\Big) = \overline{B}_e$$

is also a Markov substitute set. It is stable : $k(s, \overline{B}_e) = 1$ for any $s \in \overline{B}_e$. Moreover, for any $s, s' \in \overline{B}_e$,

$$q_{\overline{B}_e}(s'') = q_{\overline{B}_e}(\{s,s'\}) q_{\{s,s'\}}(s''), \qquad s'' \in \{s,s'\},$$

proving (still from the above lemma) that k is reversible with respect to $q_{\overline{B}_e}$. □

This leads us to define a equivalence relation on strings.

Definition 3.2.1
For any pair of strings s *and* s', *we will say that*

$$s \sim s' \iff \sum_n k^n(s,s') > 0.$$

This is an equivalence relation, and the classes will be called syntactic categories.

Remark that the equivalence classes thus defined depend only on the syntagm sets of $\mathscr{R}$, and not on the actual substitute measures, and form Markov substitute sets.

3.2.5 Crossing-over dynamics

In the same vein as the crossing-over dynamics defined in section 2.2.5 on page 73, we will now present a second approach that requires neither knowledge of substitute measures, nor a structure with no infinite loops, but works on whole texts. This new dynamics will again be a way to accelerate the first crossing-over dynamics by allowing multiple swap at the same time.

Let us extend on texts the previously defined parsing kernel Π by defining

$$\overline{\Pi}\Big(s_{1:n}, t_{1:n}\Big) = \prod_{k=1}^{n} \overline{\Pi}(s_k, t_k), \quad s_{1:n} \in \Big(D^+\Big)^n, t_{1:n} \in \mathscr{T}^n.$$

Let us consider some crossing-over tree kernel $\Big(\mathfrak{s}(t,t'), t, t' \in \bar{\mathfrak{T}}\Big)$, such that $\mathfrak{s}(t,t') > 0$ implies that

$$\begin{aligned} &\varsigma\left(t'_k\right) = \varsigma\left(t_k\right), && 1 \leqslant k \leqslant n, \\ &\sum_{k=1}^{n} \chi(t_k, j, e) = \sum_{k=1}^{n} \chi(t'_k, j, e), && \big[_j e \in \mathscr{R}. \end{aligned} \tag{3.2.4}$$

Let us remark that as a consequence of these properties,

$$\mathbb{P}_S^{\otimes n}\Big(\varphi\left(t_k\right), 1 \leqslant k \leqslant n\Big) = \mathbb{P}_S^{\otimes n}\Big(\varphi\left(t'_k\right), 1 \leqslant k \leqslant n\Big). \tag{3.2.5}$$

Such crossing-over tree kernel will be constructed in section 3.C on page 163.

Let us consider the conditional probability kernel K defined for $s_{1:n} \neq s'_{1:n} \in \mathfrak{T}$ as

$$\begin{aligned} K\Big(s_{1:n}, s'_{1:n}\Big) = \sum_{t,t' \in \bar{\mathfrak{T}}^+} \overline{\Pi}\Big(s_{1:n}, t\Big)\mathfrak{s}\Big(e(t), e(t')\Big)\zeta\Big(t', s'_{1:n}\Big) \\ \times \left(\frac{\zeta\Big(t, s_{1:n}\Big)}{\overline{\Pi}\Big(s_{1:n}, t\Big)} \frac{\mathfrak{s}\Big(t, t'\Big)}{\mathfrak{s}\Big(t', t\Big)} \frac{\overline{\Pi}\Big(s'_{1:n}, t'\Big)}{\zeta\Big(t', s'_{1:n}\Big)} \wedge 1\right). \end{aligned}$$

Proposition 3.2.7
The crossing-over sample kernel K is reversible with respect to $\mathbb{P}_S^{\otimes n}$.

PROOF. As a consequence of eq. (3.2.5) on the current page, we have to prove that the kernel K is symmetric. This property can easily be deduced from the following identity

$$K\Big(s_{1:n}, s'_{1:n}\Big) = \sum_{t_{1:n}, t'_{1:n} \in \mathscr{T}^n} \Big(\Pi\Big(s_{1:n}, t_{1:n}\Big)\zeta\Big(t_{1:n}, t'_{1:n}\Big) \wedge \Pi\Big(s'_{1:n}, t'_{1:n}\Big)\zeta\Big(t'_{1:n}, t_{1:n}\Big)\Big).$$

□

3.3 Crossing-over dynamics and the maximum likelihood estimator

Let us consider some given (deterministic) sample $s_{1:n} \in \mathscr{D}^n$, where $\mathscr{D} \in D^+$ is a finite domain (for example the set of sentences of length no larger than some constant). Let us assume that $s_{1:n}$ is not constant (in other words let us assume that $|\{s_i, 1 \leqslant i \leqslant n\}| \geqslant 2$).

Let $\mathscr{B}$ be a finite set of finite subsets of D^+, such that

$$\sum_{x \in (D^*)^2} \sum_{j=1}^{n} \mathbb{1}\Big(s_j = \alpha(x, y)\Big) > 0, \qquad y \in B. \tag{3.3.1}$$

This means that each member of each substitute set in $\mathscr{B}$ is present in the substrings of the sample $s_{1:n}$.

We will say that $p \in \mathscr{M}_+^1(\mathscr{D})$ is a $\mathscr{B}$-Markov substitute process on $\mathscr{D}$ if it satisfies eq. (2.1.1) on page 64 and there are substitute measures $q_B \in \mathscr{M}_+^1(B)$ for all $B \in \mathscr{B}$, such that for any $y, y' \in B$ and any $x \in (D^*)^2$ such that $\Big\{\alpha(x, y), \alpha(x, y')\Big\} \subset \mathscr{D}$,

$$p\Big[\alpha(x, y)\Big] q_B(y') = p\Big[\alpha(x, y')\Big] q_B(y).$$

This definition goes with a modification of the equivalence relation defined by eq. (2.3.1) on page 75, where we impose that the path should belong to the restricted domain $\mathscr{D}$. Namely

$$\begin{aligned} s \sim_{\mathscr{B},\mathscr{D}} s' \iff \exists (x_j, y_j, y'_j), x_j \in (D^*)^2, y_j, y'_j \in B_j \in \mathscr{B}, \\ \alpha(x_j, y_j), \alpha(x_j, y'_j) \in \mathscr{D}, \alpha(x_{j-1}, y'_{j-1}) = \alpha(x_j, y_j), 0 \leqslant j \leqslant J, \\ s = \alpha(x_0, y_0), s' = \alpha(x_J, y'_J). \end{aligned}$$

It is easy to extend proposition 2.3.2 on page 81 to show that the set of $\mathscr{B}$-Markov substitute processes on $\mathscr{D}$ whose support is the minimal possible support including $\{s_i, 1 \leqslant i \leqslant n\}$, is an exponential family of the form

$$\mathfrak{M}_{\mathscr{C}}(\mathscr{D}, \mathscr{B}) = \left\{ \left(p_\beta(s) = Z_\beta^{-1} \exp\left(-\sum_{i=1}^{I} \beta_i U_i(s) \right), s \in \bigcup \mathscr{C} \right), \beta \in \mathbb{R}^I \right\},$$

where

$$\mathscr{C} = \Big\{ C \in {}^{\mathscr{D}}/_{\sim_{\mathscr{B},\mathscr{D}}};\ \sum_{i=1}^{n} \mathbb{1}(s_i \in C) > 0 \Big\}.$$

We would like in this section to solve the question of finding the maximum likelihood estimator of $s_{1:n}$ in the set of distributions $\mathfrak{M}_{\mathscr{C}}(\mathscr{D}, \mathscr{B})$. In other words we would like to solve

$$\inf_{\beta \in \mathbb{R}^I} \sum_{i=1}^{n} -\log[p_\beta(s_i)].$$

This is a convex minimization problem, and by taking derivatives, we can see that when the minimum is reached, it is characterized by the fact that

$$\int_{\mathscr{D}} U_i \, \mathrm{d}p_{\beta_*} = \frac{1}{n} \sum_{j=1}^{n} U_i(s_j), \qquad 1 \leqslant i \leqslant I.$$

We will prove in this section that the minimum is indeed reached when eq. (3.3.1) on the previous page is satisfied. In practice however, obtaining an explicit parametrization of p_β (through an explicit computation of the potential functions U_i), is only possible in very simple situations, so that this result is only of theoretical interest.

We will show on the other hand that a crossing-over dynamics can be used to compute a Monte-Carlo approximation of p_{β_*}.

Let K_m be a symmetric kernel on the product $\mathscr{D}^{nm}$, such that

$$K_m(s_{1:nm}, s'_{1:nm}) > 0 \implies \left(\sum_{j=1}^{nm} U_i(s_j) - U_i(s'_j) = 0, \quad 1 \leqslant i \leqslant I \right).$$

Note that the dynamics K defined in section 3.2.5 on page 134 verifies this property. Let us assume moreover that K_m is permutation invariant in the sense that for any permutation σ,

$$K_m(s_{1:nm} \circ \sigma, s'_{1:nm} \circ \sigma) = K_m(s_{1:nm}, s'_{1:nm}), \qquad s_{1:nm} \in \mathscr{D}^{nm}, s'_{1:nm} \in \mathscr{D}^{nm},$$

where $s_{1:nm} \circ \sigma = \Big(s_{\sigma(i)}, 1 \leqslant i \leqslant nm\Big)$. Let us assume also that

$$K_m\Big[\Big(\alpha(x,y), \alpha(x',y'), s_{3:nm}\Big), \Big(\alpha(x,y'), \alpha(x',y), s_{3:nm}\Big)\Big] > 0,$$
$$x, x' \in (D^*)^2, y, y' \in B \in \mathscr{B}, \alpha(x,y), \alpha(x,y'), \alpha(x',y), \alpha(x',y') \in \mathscr{D}.$$

This means that K_m performs individual swaps with a positive probability.

It is quite easy to modify the dynamics of section 3.2.5 on page 134 to ensure these properties, by applying a random permutation to the sentences for the first one, and setting the probability of each crossing over to be strictly less than one for the second.

We will show that K_m can be used to sample (approximately) from p_{β_*}. Let $S^m \in (D^+)^{mn}$ be defined as

$$S^m_{\sigma(kn+i)} = s_i, \qquad 0 \leqslant k < m, \ 1 \leqslant i \leqslant n,$$

where σ is a uniform random permutation of $\{1,\dots,nm\}$ independent of everything else, so that S^m is a random uniform shuffle of m copies of s.

Let us consider for each m the distribution on $\mathscr{D}^{nm}$ defined as

$$p_m = \lim_{t\to\infty} \frac{1}{t} \sum_{u=1}^{t} \mathbb{P}_{S^m} K_m^u.$$

Since the initial distribution $\mathbb{P}_{S^m}$ is exchangeable, and since K_m is permutation invariant, p_m is an exchangeable probability measure. Moreover, since K_m is a symmetric kernel, and since the starting distribution is a uniform shuffle of a single configuration, p_m is uniform on its support, its support being permutation invariant. Let us consider a random variable $\big(\overline{S}_{m,i}, 1 \leqslant i \leqslant nm\big)$ distributed according to p_m, and let us put

$$N_m = \sum_{i=1}^{nm} \delta_{\overline{S}_{m,i}} \in \mathscr{M}_+(\mathscr{D}).$$

This is a random measure with integer weights. According to what we said above about p_m,

$$\mathbb{P}\big(N_m = \nu\big) = \frac{(nm)!}{Z_m \prod_{s\in D^+} \nu(s)!}, \qquad \nu \in \mathrm{supp}\big(\mathbb{P}_{N_m}\big),$$

where the partition function Z is defined as

$$Z_m = \sum_{\nu\in\mathrm{supp}(\mathbb{P}_{N_m})} \frac{(nm)!}{\prod_{s\in D^+} \nu(s)!},$$

and where we set by convention that $0! = 1$. This comes from the fact that the map $\Phi(s_{1:nm}) = \sum_{i=1}^{nm} \delta_{s_i}$ is such that $\Phi^{-1}(\nu)$ is made of $\dfrac{(nm)!}{\prod_{s\in D^+} \nu(s)!}$ distinct permutations of the same sequence of sentences, that are equiprobable under p_m, as soon as $\nu \in \mathrm{supp}(\mathbb{P}_{N_m})$.

Let us recall Stirling's approximation formula (see for example [Fel68], p.54)

$$\exp\left(\frac{1}{12n+1}\right) < \frac{n!}{\sqrt{2\pi} n^{n+1/2} \exp(-n)} < \exp\left(\frac{1}{12n}\right), \quad n \geqslant 1.$$

Since $\Big|\mathrm{supp}(\mathbb{P}_{N_m})\Big| \leqslant (nm+1)^{|\mathscr{D}|}$, the following lemma is a consequence of Stirling's approximation formula.

Lemma 3.3.1
Let $\mu_m \in \mathrm{supp}\Big(\mathbb{P}_{N_m/(nm)}\Big)$ *be such that*

$$H(\mu_m) = \max\Big\{H(\nu),\ \nu \in \mathrm{supp}\Big(\mathbb{P}_{N_m/(nm)}\Big)\Big\},$$

where

$$H(\mu) = -\sum_{s\in\mathscr{D}} \mu(s)\log\Big[\mu(s)\Big], \quad \mu \in \mathscr{M}_+^1(\mathscr{D}),$$

is the Shannon entropy.

There is a real positive constant c *(depending only on* $|\mathscr{D}|$*) such that*

$$(nm+1)^{-c} \leqslant \frac{\mathbb{P}(N_m = \nu)}{\exp\Big\{nm\Big[H\Big(\nu/(nm)\Big) - H\Big(\mu_m\Big)\Big]\Big\}} \leqslant (nm+1)^{c}. \tag{3.3.2}$$

PROOF. Let us consider some $\nu \in \mathrm{supp}\Big(\mathbb{P}_{N_m}\Big)$, and let us put $\mathscr{D}_\nu = \mathrm{supp}(\nu)$, $d_\nu = |\mathscr{D}_\nu|$ and $d = |\mathscr{D}|$. From Stirling formula, we get

$$\begin{aligned}
&\frac{(nm)!}{\prod\limits_{s\in\mathscr{D}_\nu}\nu(s)!} \\
&\quad\leqslant (2\pi)^{-(d_\nu-1)/2}\frac{(nm)^{nm}}{\prod\limits_{s\in\mathscr{D}_\nu}\nu(s)^{\nu(s)}}\left(\frac{nm}{\prod\limits_{s\in\mathscr{D}_\nu}\nu(s)}\right)^{1/2}\exp\left(\frac{1}{12nm} - \sum_{s\in\mathscr{D}_\nu}\frac{1}{12(\nu(s)+1)}\right) \\
&\qquad\qquad\leqslant (nm)^{1/2}\exp\Big[nmH\Big(\nu/(nm)\Big)\Big].
\end{aligned}$$

On the other hand

$$\begin{aligned}
&\frac{(nm)!}{\prod\limits_{s\in\mathscr{D}_\nu}\nu(s)!} \\
&\quad\geqslant (2\pi)^{-(d_\nu-1)/2}\frac{(nm)^{nm}}{\prod\limits_{s\in\mathscr{D}_\nu}\nu(s)^{\nu(s)}}\left(\frac{nm}{\prod\limits_{s\in\mathscr{D}_\nu}\nu(s)}\right)^{1/2}\exp\left(\frac{1}{12(nm+1)} - \sum_{s\in\mathscr{D}_\nu}\frac{1}{12\nu(s)}\right).
\end{aligned}$$

Let us then remark that, introducing the uniform probability measure $U_{\mathscr{D}_\nu}$ on $\mathscr{D}_\nu$,

$$\begin{aligned}
\prod_{s\in\mathscr{D}_\nu}\nu(s) &= \exp\biggl(\sum_{s\in\mathscr{D}_\nu}\log\bigl(\nu(s)\bigr)\biggr)\\
&= \exp\biggl[-d_\nu\frac{1}{d_\nu}\sum_{s\in\mathscr{D}_\nu}\log\biggl(\frac{nm}{d_\nu\nu(s)}\biggr)+d_\nu\log\biggl(\frac{nm}{d_\nu}\biggr)\biggr]\\
&= \exp\biggl[d_\nu\biggl(\log\biggl(\frac{nm}{d_\nu}\biggr)-\mathscr{K}\bigl(U_{\mathscr{D}_\nu},\nu/(nm)\bigr)\biggr)\biggr]\\
&\leqslant \exp\biggl[d_\nu\log\biggl(\frac{nm}{d_\nu}\biggr)\biggr]\\
&\leqslant \exp\biggl[d\max\biggl\{1,\log\biggl(\frac{nm}{d}\biggr)\biggr\}\biggr]\\
&\leqslant \exp\biggl[d\log\biggl(\frac{enm}{d}\biggr)\biggr]\\
&\leqslant \biggl(\frac{nm}{d}\biggr)^d\exp(d).
\end{aligned}$$

Combining the last two equations, we get

$$\begin{aligned}
\frac{(nm)!}{\prod_{s\in\mathscr{D}_\nu}\nu(s)!} &\geqslant d^{1/2}\biggl(\frac{2\pi nm}{d}\biggr)^{-(d-1)/2}\exp\bigl(-7d/12\bigr)\exp\bigl[nmH\bigl(\nu/(nm)\bigr)\bigr]\\
&\geqslant \exp\biggl(\frac{d}{2}\bigl(\log(d)-\log(2\pi)-7/12\bigr)+\frac{1}{2}\log(2\pi)\biggr)(nm)^{-(d-1)/2}\exp\bigl[nmH\bigl(\nu/(nm)\bigr)\bigr]\\
&\geqslant (nm)^{-(\max\{d,10\}-1)/2}\exp\bigl[nmH\bigl(\nu/(nm)\bigr)\bigr].
\end{aligned}$$

Accordingly, there is a constant $a>0$, depending only on $|\mathscr{D}|$, such that

$$(nm)^{-a}\leqslant\frac{(nm)!}{\prod_{s\in\mathscr{D}_\nu}\nu(s)!}\exp\bigl[-nmH\bigl(\nu/(nm)\bigr)\bigr]\leqslant(nm)^{1/2}. \tag{3.3.3}$$

Let us remark now that

$$\bigl|\mathrm{supp}\bigl(\mathbb{P}_{N_m}\bigr)\bigr|\leqslant(nm+1)^d.$$

This implies that

$$Z_m\leqslant(nm+1)^d(nm)^{1/2}\exp\bigl(nmH(\mu_m)\bigr).$$

On the other hand

$$Z_m \geqslant (nm+1)^{-a} \exp\Big(nmH(\mu_m)\Big),$$

so that

$$(nm+1)^{-(c+d+1/2)} \leqslant \frac{\mathbb{P}(N_m = \nu)}{\exp\Big(nm\Big[H\Big(\nu/(nm)\Big) - H(\mu_m)\Big]\Big)} \leqslant (nm+1)^{a+1/2},$$

ending the proof of the above lemma. □

To use this lemma, it remains to prove that $\mathrm{supp}(N_m)$ is essentially defined by the constraints

$$\mathrm{supp}\Big(\mathbb{P}_{N_m/(mn)}\Big) \simeq \Big\{\nu \in \mathscr{M}_+^1\Big(\bigcup \mathscr{C}\Big), \int U_i(s)\,\mathrm{d}\nu(s) = \frac{1}{n}\sum_{j=1}^{n} U_i(s_j), 1 \leqslant i \leqslant I\Big\},$$

This fact is not so easy to prove directly. To come to a conclusion, we are going to describe a non random path in $\mathrm{supp}\Big(\mathbb{P}_{N_m/(mn)}\Big)$, starting at the initial probability measure

$$\overline{p} = \frac{1}{n}\sum_{i=1}^{n} \delta_{s_i},$$

and converging to some $p_\beta \in \mathfrak{M}_{\mathscr{C}}(\mathscr{D}, \mathscr{B})$.

Let us consider the set of vectors

$$\mathscr{Q} = \Big\{\delta_{\alpha(x,y')} + \delta_{\alpha(x',y)} - \delta_{\alpha(x,y)} - \delta_{\alpha(x',y')},\ x, x' \in (D^*)^2, (y, y') \in B^2 \in \mathscr{B},\\ \alpha(x,y), \alpha(x,y'), \alpha(x',y), \alpha(x',y') \in \mathscr{D}\Big\}.$$

Since $\mathscr{D}$ is a finite subset of D^+, $\mathscr{Q}$ is also a finite set. Let us consider the convex set

$$A = \Big\{\overline{p} + \sum_{v \in \mathscr{Q}} \gamma(v)v, \gamma \in \mathbb{R}^{\mathscr{Q}}\Big\} \cap \mathscr{M}_+^1(\mathscr{D}).$$

Let us prove first that we can find a positive constant $\zeta > 0$, a finite sequence $v_j \in \mathscr{Q}$, $1 \leqslant j \leqslant J$, such that for any large enough value of m,

$$\mu_k = \overline{p} + \sum_{j=1}^{k} \frac{\lceil nm\zeta \rceil}{nm} v_j \in A, \quad 0 \leqslant k \leqslant J,$$

and $\Big\{s \in \mathscr{D}, \mu_{J_1}(s) \geqslant \zeta\Big\} = \bigcup \mathscr{C}$.

Indeed, for any $s \in C \in \mathscr{C}$, there is i, $1 \leqslant i \leqslant n$ and a sequence

$$(x_k, y_k, y'_k), 0 \leqslant k \leqslant J, x_k \in (D^*)^2, (y_k, y'_k) \in B^2 \in \mathscr{B}^2,\\ \alpha(x_k, y_k) \in \mathscr{D}, \alpha(x_{k-1}, y'_{k-1}) = \alpha(x_k, y_k), 1 \leqslant k \leqslant J,\\ \alpha(x_0, y_0) = s_i, \alpha(x_j, y'_J) = s.$$

For any k, $0 \leqslant k \leqslant J$, there is i_k, $1 \leqslant i_k \leqslant n$ and $x'_k \in (D^*)^2$ such that $s_{i_k} = \alpha(x'_k, y'_k)$. Considering

$$v_k = \delta_{\alpha(x_k, y'_k)} + \delta_{\alpha(x'_k, y_k)} - \delta_{\alpha(x_k, y_k)} - \delta_{s_{i_k}},$$

it is easy to check that for ζ small enough and m large enough,

$$\mu_k = \overline{p} + \frac{\lceil nm\zeta \rceil}{nm} \sum_{j=0}^{k} v_j \in \mathscr{M}_+^1(\mathscr{D}),$$

(and belongs to A,) and that

$$\mu_J(s_i) \geqslant \zeta, \text{ and } \mu_J(s) \geqslant \zeta.$$

Indeed, to create μ_J, we remove some mass from $\{s_i, 1 \leqslant i \leqslant n\}$ to carry a fraction not less than ζ of it to s while spilling the rest around. As the amount of mass that is moved around goes to zero when ζ decreases and m increases, it is possible to keep some positive mass at each s_i, $1 \leqslant i \leqslant n$.

We can repeat this process, starting from the end point μ_J, and increasing m if necessary, to extend progressively the support to the whole $\bigcup \mathscr{C}$, along a sequence $\mu_k, 1 \leqslant k \leqslant J$. We build in this way a probability measure

$$\nu_m = \mu_J = \overline{p} + \sum_{j=1}^{J} \frac{\lceil nm\zeta \rceil}{nm} v_j \in \operatorname{supp}\Big(\mathbb{P}_{(nm)^{-1}N_m}\Big),$$

such that $\nu_m(s) \geqslant \zeta, s \in \bigcup \mathscr{C}$, and such that

$$\lim_{m\to\infty} \nu_m = \nu_\infty = \overline{p} + \sum_{j=1}^{J} \zeta v_j \in A,$$

with $\nu_\infty(s) \geqslant \zeta, s \in \bigcup \mathscr{C}$.

Now let us consider $\mu_* \in A$ such that

$$H(\mu_*) = \sup\Big\{H(\mu), \mu \in A\Big\}.$$

Since the Shannon entropy H is strictly convex, it reaches its maximum on the compact convex set A at a single point, so that μ_* is uniquely defined by this property.

Let us prove that $\operatorname{supp}(\mu_*) = \bigcup \mathscr{C}$. Since $H(\mu_*)$ is maximum,

$$\lim_{\substack{\lambda\to 0 \\ \lambda>0}} \sum_{s\in\bigcup\mathscr{C}} \Big(\mu_*(s) - \nu_\infty(s)\Big) \log\Big[\lambda\nu_\infty(s) + (1-\lambda)\mu_*(s)\Big] = \lim_{\substack{\lambda\to 0 \\ \lambda>0}} \frac{\partial}{\partial\lambda} H\Big(\lambda\nu_\infty + (1-\lambda)\mu_*\Big) \leqslant 0.$$

Since the left-hand side of this equation is equal to $+\infty$ as soon as $\mu_*(s) = 0$ for some $s \in \bigcup \mathscr{C}$, this proves our claim (here we need the fact that $\nu_\infty(s) > 0$, that we proved before). Let us remark that we could have used any ν_m with m large enough instead of ν_∞ in the previous equation.

Lemma 3.3.2
For any $x, x' \in (D^*)^2$, and $(y, y') \in B^2 \in \mathscr{B}^2$, such that

$$\alpha(x,y), \alpha(x,y'), \alpha(x',y), \alpha(x',y') \in \bigcup \mathscr{C},$$

$$\frac{\mu_*\Big[\alpha(x,y')\Big]}{\mu_*\Big[\alpha(x,y)\Big]} = \frac{\mu_*\Big[\alpha(x',y')\Big]}{\mu_*\Big[\alpha(x',y)\Big]}.$$

PROOF. Consider $v = \delta_{\alpha(x,y)} + \delta_{\alpha(x',y')} - \delta_{\alpha(x,y')} - \delta_{\alpha(x',y)}$. For any small enough positive or negative value of γ, $\mu_* + \gamma v \in A$, therefore

$$\begin{aligned} 0 &= \left.\frac{\partial}{\partial\gamma}\right|_{\gamma=0} H(\mu_* + \gamma v) \\ &= \log\Big[\mu\Big(\alpha(x,y')\Big)\Big] + \log\Big[\mu\Big(\alpha(x',y)\Big)\Big] - \log\Big[\mu\Big(\alpha(x,y)\Big)\Big] - \log\Big[\mu\Big(\alpha(x',y')\Big)\Big]. \ \square \end{aligned}$$

Since $\{s_i, 1 \leqslant i \leqslant n\} \subset \mathrm{supp}(\mu_*)$, μ_* satisfies eq. (2.1.1) on page 64, therefore the above lemma implies that $\mu_* \in \mathfrak{M}_{\mathscr{C}}(\mathscr{D}, \mathscr{B})$. Moreover, as it is the case for any probability measure in A,

$$\sum_{s\in\mathscr{D}} \mu_*(s)\, U_i(s) = \sum_{s\in\mathscr{D}} \overline{p}(s)\, U_i(s),$$

proving that β_* exists and that $\mu_* = p_{\beta_*}$.

Since $\nu, \mu_* \in A$ and since $\mathscr{Q}$ is symmetric, we can write μ_* as

$$\mu_* = \nu_\infty + \sum_{j=1}^{J} \gamma_j v_j, v_j \in \mathscr{Q}, \gamma_j > 0, 1 \leqslant j \leqslant J, \tag{3.3.4}$$

where v_j are independent vectors. Consider the lattice

$$L_m = \left(\nu_m + \frac{1}{nm} \sum_{j=1}^{J} \mathbb{Z}\, v_j\right) \cap A.$$

Since $\mathrm{supp}(\nu_\infty) = \mathrm{supp}(\mu_*) = \bigcup \mathscr{C}$, it is easy to check that for m large enough, there is a connected path in this lattice joining ν_m to the point of the lattice nearest to μ_*. This a consequence of the fact that the segment joining ν_∞ to μ_* is in the

interior of $A \cap \left(\nu_\infty + \sum_{j=1}^{J} \mathbb{R}v_j\right)$ viewed as a subset of $\nu_\infty + \sum_{j=1}^{J} \mathbb{R}v_j$. Since the nearest point to μ_* in L_m converges to μ_* according to eq. (3.3.4) on the preceding page, and since $L_m \in \operatorname{supp}\Big(\mathbb{P}_{N_m/(nm)}\Big)$, we have proved that

$$\lim_{m\to\infty} d\Big(\operatorname{supp}\Big(\mathbb{P}_{N_m/(nm)}\Big), \mu_*\Big) = 0,$$

where d is (for instance) the Euclidean distance.

Proposition 3.3.3
Under the hypotheses described at the beginning of this section, the maximum likelihood

$$\sup_{\beta\in\mathbb{R}^I} \sum_{j=1}^{n} \log\Big[p_\beta(s_i)\Big]$$

is reached at some $\beta_* \in \mathbb{R}^I$. For any $\varepsilon > 0$, there are $\eta > 0$ and $M > 0$ such that for any $m \geqslant M$,

$$\mathbb{P}\Big[d\Big(N_m, p_{\beta_*}\Big) \geqslant \varepsilon\Big] \leqslant \exp(-nm\eta),$$

where $d(\mu,\nu) = \sqrt{\sum_{s\in\mathscr{D}} \Big(\mu(s) - \nu(s)\Big)^2}$. Moreover,

$$\lim_{m\to\infty} \mathbb{E}\Big(N_m/(nm)\Big) = p_{\beta_*}.$$

Proof. For any $\varepsilon > 0$, there is $\eta > 0$ such that

$$\Big\{\mu \in A, d(\mu, p_{\beta_*}) \geqslant \varepsilon\Big\} \subset \Big\{\mu \in A, H(\mu) \leqslant H(p_{\beta_*}) - 3\eta\Big\},$$

because H is strictly concave.

For m large enough, there is a measure $\mu \in \operatorname{supp}\Big(\mathbb{P}_{N_m/(nm)}\Big)$ whose entropy is such that $H(\mu) \geqslant H(p_{\beta_*}) - \eta$. Thus

$$\begin{aligned}
\mathbb{P}\Big[d\Big(N_m/(nm), p_{\beta_*}\Big) \geqslant \varepsilon\Big] &\leqslant \mathbb{P}\Big[H\Big(N_m/(nm)\Big) \leqslant H(p_{\beta_*}) - 3\eta\Big] \\
&\leqslant \mathbb{P}\Big[H\Big(N_m/(nm)\Big) \leqslant H(\mu) - 2\eta\Big] \\
&\leqslant \exp(-nm\eta).
\end{aligned}$$

The last inequality is a consequence of eq. (3.3.2) on page 138 and of the fact that

$$\Big|\operatorname{supp}\Big(\mathbb{P}_{N_m/(nm)}\Big)\Big| \leqslant (nm+1)^{|\mathscr{D}|}$$

is polynomial and not exponential in mn. To study the convergence of the expectation of $N_m/(nm)$, we can remark that the Euclidean distance between two probability distributions cannot be greater than 2, so that, for m large enough,

$$d\Big[\mathbb{E}\Big(N_{mn}/(nm)\Big), p_{\beta_*}\Big] \leqslant \mathbb{E}\Big[d\Big(N_m/(nm), p_{\beta_*}\Big)\Big] \leqslant 2^{1/2}\varepsilon + \exp(-nm\eta),$$

proving that

$$\lim_{m\to\infty} \mathbb{E}\Big(N_m/(nm)\Big) = p_{\beta_*}. \qquad \square$$

3.4 Building a Markov ruleset

Let us now discuss the question of building a Markov ruleset. If we index the dictionnary $D = (w_i)_{i=1}^M$, the trivial set of rules

$$\mathscr{R}_0 = \bigoplus_i 1 \otimes [_i w_i$$

is of course Markov, for any distribution. The question now is, given a Markov grammar $\mathscr{R}$, how to make it grow.

3.4.1 Adding new rules

If we are given a Markov grammar $\mathscr{R}$, with the corresponding substitute measures, we already know that B_e is a Markov substitute set for any expression e. We could now want to add new rules $[_i e$ to the grammar, so that the new grammar

$$\mathscr{R}' = \mathscr{R} \bigoplus [_i e$$

is still Markov. Let us note

$$I(\mathscr{R}) = \Big\{i \in \mathbb{N}, \exists e \in S^+, [_i e \in \mathscr{R}\Big\}.$$

As a first step, we will only add rules $[_i e$ where $i \notin I(\mathscr{R})$. In this case, $B'_i = B_i$, $i \in I(\mathscr{R})$ will still be Markov substitute sets, and we only have to check the property for the new syntagm sets. In practice, this means that we want to find expressions e_1, e_2 so that $B'_i = B_{e_1} \cup B_{e_2}$ is still a Markov substitute set, in which case we can add $[_i e_1 \oplus [_i e_2$ to the grammar.

Note that if we have at some point added a rule $[_i]_j$, we should simply identify the labels i and j instead.

Now, suppose we have a splitting kernel

$$\pi : D^+ \mapsto \mathscr{M}_+^1\Big((D^*)^2 \times D^*\Big)$$

such that

$$\pi(s,x,y) > 0 \text{ and } y \neq \epsilon \implies \exists \breve{x} \in (\breve{D}^*)^2, \breve{y} \in \mathscr{T} : \varsigma(\breve{y}) \in \{e_1, e_2\},$$

and such that

$$\mathscr{S}(s, B'_i) \subset \text{supp}(\pi(s,\cdot)). \tag{3.4.1}$$

If we introduce for B'_i the test functions

$$\begin{aligned} F_\theta(S,p) = \sum_{x\in(D^*)^2} \sum_{y_1,y_2\in D^+} & \mathbb{1}(x\in\theta)\Big[\pi\Big(\alpha(x,y_1),x,y_1\Big) \wedge \pi\Big(\alpha(x,y_2),x,y_2\Big)\Big] \\ & \times \Big[(1-p)\mathbb{1}\Big(S=\alpha(x,y_1)\Big)\mathbb{1}\Big(y_1\in B_{e_1}\Big)q_{B'_i\setminus B_{e_1}}(y_2) \\ & \qquad - p\mathbb{1}\Big(S=\alpha(x,y_2)\Big)\mathbb{1}\Big(y_2\in B'_i\setminus B_{e_1}\Big)q_{B_{e_1}}(y_1)\Big], \end{aligned}$$

we have that B'_i is a Markov substitute set if and only if, for some value of $p\in]0,1[$,

$$\forall\theta\in\Theta,\ \mathbb{E}\Big(F_\theta(S,p)\Big) = 0.$$

When this is the case, we have

$$q_{B'_i}(B_{e_1}) = p. \tag{3.4.2}$$

This mean that we can use the random variables $F_\theta(S,p)$ to define a test.

Remark at this point that eq. (3.4.1) on the current page is only necessary to test if the new set is a "true" Markov substitute set. Whether the condition is verified or not, it will not change the fact that if $\mathbb{E}(F_\theta(S,p)) = 0$, the invariant dynamics defined with this splitting kernel will still be invariant. Roughly speaking, the set would be a Markov substitute set only in the contexts weighted by π.

The test function presented in section 3.2.2 on page 127 may also be used to test these new rules, however, the introduction of the measure ν_θ can make the test less efficient when θ is large.

3.4.2 Reducing the context space

The test we described here relies on the fact that $B = C_1 \cup C_2$ is a Markov substitute set if and only if for some $p\in]0,1[$,

$$\forall\theta\in\Theta,\ \mathbb{E}\Big(F_\theta(S,p)\Big) = 0,$$

assuming that the context set Θ contains all singletons $\{x\}, x\in D^{*2}$.

However, we can use here the reduction kernel r to reduce the context space, by observing that for any context (x_1, x_2), any pair of expressions (e_1, e_2) such that $x_i \in B_{e_i}$, any string y,

$$\mathbb{P}_S\Big(\gamma(x_1, y, x_2)\Big) = \Bigg(\sum_{\substack{x_1' \in B_{e_1},\\ x_2' \in B_{e_2}}} \mathbb{P}_S\Big(\gamma(x_1', y, x_2')\Big)\Bigg) q_{B_{e_1}}(x_1) q_{B_{e_2}}(x_2). \tag{3.4.3}$$

Let us assume that we have a collection of expressions $(e_i)_i, i \in I$, such that $D^* = \bigcup_{i\in I} B_{e_i}$ and $(\{B_{e_i}, i \in I\})^2 \subset \Theta$, and do not assume any more that Θ contains all one point sets.

Proposition 3.4.1
In this case, the union $B = C_1 \cup C_2$ is a Markov substitute set, if and only if, for some $p \in]0,1[$,

$$\forall \theta \in \Theta,\ \mathbb{E}\Big(F_\theta(S,p)\Big) = 0,$$

PROOF. We see from the definition of $F_\theta(S,p)$ that $\mathbb{E}\Big(F_\theta(S,p)\Big) = 0$ if and only if

$$\sum_{x\in\theta}\sum_{y\in C_1} \mathbb{P}_S\Big(\alpha(x,y)\Big) = p \sum_{x\in\theta}\sum_{y\in C_1\cup C_2} \mathbb{P}_S\Big(\alpha(x,y)\Big), \tag{3.4.4}$$

whereas $C_1 \cup C_2$ is a Markov substitute set if and only if for any $x \in (D^*)^2$,

$$\sum_{y\in C_1} \mathbb{P}_S\Big(\alpha(x,y)\Big) = p \sum_{y\in C_1\cup C_2} \mathbb{P}_S\Big(\alpha(x,y)\Big). \tag{3.4.5}$$

Now, for any $x = (x_1, x_2) \in (D^*)^2$, there are e_1, e_2 such that $x_1 \in B_{e_1}, x_2 \in B_{e_2}$, and we see that the combination of eq. (3.4.4) on the current page applied to $\theta = B_{e_1} \times B_{e_2}$ and of eq. (3.4.3) on this page implies eq. (3.4.5) on the current page. □

To implement this, we can put the whole of $\{B_e, \exists s \in D^+, r(s,e) > 0\}^2$ in Θ, since we can make Θ as big as we like, knowing that we only have to test on the contexts that we actually see, and compute $\mathbb{1}\Big(x \in B_{e_1} \times B_{e_2}\Big)$ as

$$\mathbb{1}\Big(r(x_i, e_i) > 0, i = 1, 2\Big).$$

3.4.3 Saturation

We actually can go further than simply adding rules to the ruleset. Consider for a given ruleset $\mathscr{R}$, the collection of subsets $\mathscr{B} = \{B_i, i \in \mathbb{N}\}$, and, as constructed in section 2.3 on page 74, the graph in D^+ so that $(\alpha(x,y), \alpha(x,y'))$ is an edge

if, and only if, y and y' are in the same B_i. This graph defines an equivalence relation $\sim_{\mathscr{R}}$, which is obviously the same as the one defined in definition 3.2.1 on page 133.

Now we can remark that we can change the actual graph (that is, the ruleset) without changing the actual equivalence relation. An "sparse" graph, with few edges and small B_i sets, was considered in section 2.3 on page 74, where the goal was to simplify as much as possible the graph to ease computation. A "fuller" graph with many edges and big sets, on the other hand, accelerates the invariant dynamics, since they only jump along the edges of the graph. We will here attempt to build, from a given ruleset, another one, with the same equivalence relation, but with richer syntagm sets.

Let us first consider a small example to see how this works. Say we have in the rules

$$\begin{aligned} &\lfloor_1 ab \\ \oplus &\lfloor_2 b \oplus \lfloor_2 c. \end{aligned}$$

Such a set of rules will not be able to recognise B_{ac} as a syntagm set, while it is a Markov substitute set, easily obtained from these rules. Indeed, the set of rules

$$\begin{aligned} &\lfloor_1 a \rfloor_2 \\ \oplus &\lfloor_2 b \oplus \lfloor_2 c, \end{aligned}$$

is still Markov (if the first one was, obviously), and does recognise ac. The missing part is $\lfloor_1 a \rfloor_2$, which may be obtained by parsing $\lfloor_1 ab$ by $\lfloor_2 b$.

The question whether we should keep the original $\lfloor_1 ab$ is open. It clutters a bit the reference grammar, and any ab may still be parsed by $\lfloor_1 a \rfloor_2$ and $\lfloor_2 b$, but it may speed up the parses, and allows more leeway in the order in which the different elements are parsed. Moreover, it may be smart to keep these elements to maximize the number of parses we can find.

For example, consider the (supposed Markov) set of rules

$$\begin{aligned} &\lfloor_1 abc \oplus \lfloor_1 d \\ \oplus &\lfloor_2 ab \oplus \lfloor_2 e \\ \oplus &\lfloor_3 bc \oplus \lfloor_3 f. \end{aligned}$$

The following set would still be Markov:

$$\begin{aligned} &\lfloor_1 \rfloor_2 c \oplus \lfloor_1 a \rfloor_3 \oplus \lfloor_1 d(\oplus \lfloor_1 abc) \\ \oplus &\lfloor_2 ab \oplus \lfloor_2 e \\ \oplus &\lfloor_3 bc \oplus \lfloor_3 f. \end{aligned}$$

It should be clear here that both $[_1]_2c$ and $[_1a]_3$ come from the same $[_1abc$. If we did not keep the original $[_1abc$, we would only get, say, $[_1]_2c$, and af would not be recognised as a syntagm. Running several classic parses (and deleting parsed elements), and merging the results, will give some of the missed elements, but not all (for example, those needing both $[_1]_2c$ and $[_1a]_3$ to parse them, say $[_4]_2ca]_3$).

Remark however that the redundant elements can always be removed afterwards, by removing all elements that may be parsed by the grammar.

As mentioned, we do not want to modify the rules so that they generalize more, as in the previous section. Rather, we want to keep the same syntactic classes. This leads us to the following definitions.

Definition 3.4.1
A ruleset $\mathscr{R}$ is weakly smaller than another ruleset $\mathscr{R}'$ when, for any syntagm type i, the set B_i of syntagms for $\mathscr{R}$ is contained in a set of syntagms B'_j of $\mathscr{R}'$.

$$\mathscr{R} \lesssim \mathscr{R}' \iff \forall i\, \exists j;\ B_i \subset B'_j.$$

Weak equality is of course the case when there is a bijection $f : \mathbb{N} \mapsto \mathbb{N}$ such that

$$\forall i;\ B_i = B'_{f(i)}.$$

The weak order on ruleset is obviously a reflexive partial order.

Definition 3.4.2
A ruleset $\mathscr{R}$ is compatible with another $\mathscr{R}'$, when, the set of syntagms of any type is included in a equivalence class of $\sim_{\mathscr{R}'}$:

$$\mathscr{R} \Leftarrow \mathscr{R}' \Leftrightarrow \forall i\, \forall a, b \in B_i;\ a \sim_{\mathscr{R}'} b.$$

It is quite obvious that a ruleset weakly smaller than a compatible one is still compatible, and that the relation $\Leftarrow$ is transitive.

It follows from this definition and from lemma 3.2.6 on page 133 that any ruleset compatible with a Markov ruleset, is also Markov.

Proposition 3.4.2
Let us take two rulesets $\mathscr{R}, \mathscr{R}'$. Then we have the following

$$\mathscr{R} \lesssim \mathscr{R}' \implies \left(\forall s, s' \in D^+, s \sim_{\mathscr{R}} s' \implies s \sim_{\mathscr{R}'} s'\right)$$
$$\mathscr{R} \Leftarrow \mathscr{R}' \implies \left(\forall s, s' \in D^+, s \sim_{\mathscr{R}} s' \implies s \sim_{\mathscr{R}'} s'\right).$$

This means that a weakly bigger, compatible ruleset will define the same equivalence relation.

Proof. Suppose $\mathscr{R} \lesssim \mathscr{R}'$, and $s \sim_{\mathscr{R}} s'$. By definition, we have a chain (x_j, y_j, y'_j), $y_j, y'_j \in B_{k_j}$, so that $s = \alpha(x_0, y_0), \alpha(x_J, y'_J) = s'$ and $\alpha(x_{j-1}, y'_{j-1}) = \alpha(x_j, y_j)$. By hypothesis, there are k'_j so that $B_{k_j} \subset B'_{k'_j}$, and $y_j, y'_j \in B_{k_j}$. This means that $s \sim_{\mathscr{R}'} s'$.

Suppose now that $\mathscr{R} \lessapprox \mathscr{R}'$, and $s \sim_{\mathscr{R}} s'$. Again, we have a chain (x_j, y_j, y'_j), $y_j, y'_j \in B_{k_j}$, so that $s = \alpha(x_0, y_0), \alpha(x_J, y'_J) = s'$ and $\alpha(x_{j-1}, y'_{j-1}) = \alpha(x_j, y_j)$. This time, we know that $y_j \sim_{\mathscr{R}'} y'_j$, so that again $s \sim_{\mathscr{R}'} s'$. □

The goal now will be to build, from a given ruleset, a bigger, compatible one. This can be done in the following way.

Proposition 3.4.3
Let $\mathscr{R}$ be a Markov ruleset. Consider the sequence of grammars $(\mathscr{R}_i)_i$:

- $\mathscr{R}_0 = \mathscr{R}$,
- *$\mathscr{R}_{i+1}$ is a splitting of $\cup_{j \leqslant i} \mathscr{R}_j$ with reference grammar $\cup_{j \leqslant i} \mathscr{R}_j$.*

Then $\cup_i \mathscr{R}_i$ is weakly bigger and compatible, and will be called the saturated ruleset, noted $\overline{\mathscr{R}}$.

Proof. Let us begin with the fact that $\overline{\mathscr{R}} \gtrsim \mathscr{R}$. This comes from the fact that $\mathscr{R}_{i+1} \gtrsim \mathscr{R}_i$, which in turns comes from the fact that, if we put B_i the syntagm sets of $\mathscr{R}_i$, and B'_i those of $\mathscr{R}_{i+1}$, $B_i \subset B'_i$.

Now let us consider the compatibility. We will prove the weaker result:

Lemma 3.4.4
For any ruleset $\mathscr{R}$, for any $[_i\alpha(x, y), [_j y \in \mathscr{R}$, the ruleset $\mathscr{R}' = \mathscr{R} \oplus [_i\alpha(x,]_j)$ is compatible with $\mathscr{R}$.

Proof. Let us note B_i the syntagms sets for $\mathscr{R}$, and B'_i the syntagms sets for $\mathscr{R}'$. Let us begin by taking a syntagm $s \in B_y$, and a parse $\breve{s}$, $\Pi_y(s, \breve{s}) > 0$.

Take any syntagm $a \in B'_i$, and a parse $\breve{a}$, $\Pi_i(a, \breve{a}) > 0$.

Consider now all c, t such that $\breve{a} = \alpha(c, (_j t)_j)$ and $[_j\varsigma(t) = [_i\alpha(a,]_j)$. We will prove by induction on the number of such splits that a is equivalent to a syntagm of type i for $\mathscr{G}$.

If there is none, then for all c, t such that $\breve{a} = \alpha(c, (_j t)_j)$, $[_j\varsigma(t) \in \mathscr{R}$, and $a \in B_i$. This initializes the induction.

If not, let us consider one of the deepest such x, y. This means that for all x', y', if $t = \alpha\Big(x', (_k y')_k\Big)$, $[_k\varsigma(y') \in \mathscr{R}$, so $\varphi(t) \in B_k$. Moreover, since $\varsigma(t) \notin \mathscr{R}$, $\varsigma(t) = \alpha(x,]_j)$, and we can take (x', y') such that $t = \alpha(x', y')$, $\varsigma(x') = \varsigma(x)$, and $\varphi(y') \in B_y$. Obviously $\varphi(y') \sim s$, so that, if $\breve{a}' = \alpha\Big(c, (_j\alpha(x', \breve{s}))_j\Big)$, $\breve{a}'$ is still a syntagm for $\mathscr{R}'$, and $a = \varphi\left(\alpha\Big(c, (_j\alpha(x', y'))_j\Big)\right) \sim \varphi(\breve{a}')$. Since $\breve{a}'$ has one

less (c, t), its realization is equivalent by hypothesis to a syntagm for $\mathscr{R}$, and so is a.

Thus any $\mathscr{R}'$-syntagm of type i is equivalent to a $\mathscr{R}$-syntagm of type i. This ends the proof. □

Since the ruleset $\overline{\mathscr{R}}$ is build by repetitive application of splits as in the lemma, the result is compatible with the first ruleset. □

Proposition 3.4.5
The process defined in proposition 3.4.3 on the preceding page is stationary in finite time.

PROOF. The saturated ruleset is obviously finite, since the lengths of its elements are bounded by the maximum length of the elements of the original ruleset $\mathscr{R}_0$. The dictionnary is finite since no new indices for brackets are created.

Since each step adds at least a new element to the saturated ruleset (or the process ends), the process finishes a.s. in finite times. We could even compute an explicit bound using this argument, but chances are it is much larger than the best possible one. □

3.4.4 Identification of labels

Nowhere in this study did we identify labels, except in the obvious case $[_i]_j$. However, if, at any point, we discover an element $s \in B_i \cap B_j$, while parsing, or simulating, for example, we can at once indentify the labels i and j.

Proposition 3.4.6
For any ruleset $\mathscr{R}$, if $B_i \cap B_j \neq \varnothing$ for some $i \neq j$, then if we define the ruleset $\mathscr{R}'$ as the ruleset where all instances of j in $\mathscr{R}$ was replaced by i, $\mathscr{R}' \Leftarrow \mathscr{R}$.

PROOF. Let us take $s \in B_i \cap B_j$. For any pair of parse trees $({}_k a)_k, ({}_k b)_k \in \breve{B}_k$, we will show, by induction on their maximal depth, that $\varphi(a) \sim \varphi(b)$.

- If their depth is 1, that is, $a, b \in D^+$, either $k \neq i$ and $\varphi(a), \varphi(b) \in B_k$, hence the result, or $k = i$. In this case, by hypothesis, $\varphi(a) \sim s \sim \varphi(b)$.

- Suppose their depth is less than n. Say $a = \alpha(x, ({}_j y)_j)$. By hypothesis, there is $y' \in \breve{B}_j$ such that $\varphi(a) \sim \varphi(\alpha(x, ({}_j y')_j))$. This means that there is $t \in \breve{B}'_k$, $\varphi(a) \sim \varphi(t)$, and for any x, y, $t = \alpha(x, ({}_j y)_j) \implies y \in \breve{B}_k$. The same holds true for b, with a parse t'.

  Now, if $k \neq i$, $t, t' \in \breve{B}_k$, and $\varphi(a) \sim \varphi(t) \sim \varphi(t') \sim \varphi(b)$. If $k = i$, then $t, t' \in \breve{B}_i \cup \breve{B}_j$, and $\varphi(a) \sim \varphi(t) \sim s \sim \varphi(t') \sim \varphi(b)$. □

3.5 Toric grammars

The formalism we used to describe rulesets corresponds exactly to the definition of toric grammars introduced in chapter 1, more precisely, reference grammars (since we only consider local expressions), minus permutations. We will, in this section, use the term reference grammar in place of ruleset.

The construction described in section 3.4 on page 144 may be used to obtain the reference grammar needed for the communication model. Let us now study the properties of this model.

3.5.1 Split and merge processes using parsing

The parsing framework we defined in section 3.1.3 on page 123 can be used to define probabilities for the splitting part of a split-and-merge process, provided the parsing extension kernels are well chosen.

Definition 3.5.1
For any bottom-up parsing extension kernel ϖ that depends only on the surface structures, that is, for any $x, x' \in \breve{B}_e \times \breve{B}_{e'}, ({}_i y)_i, ({}_i y')_i \in B_i$,

$$\varpi\Big(\alpha(x,y), \alpha(x,({}_i y)_i)\Big) = \varpi\Big(\alpha(x',y'), \alpha(x',({}_i y')_i)\Big), \tag{3.5.1}$$

we can define a splitting rule

$$\begin{aligned}\beta_\varpi(\mathscr{G}) =& \Big\{\mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus \alpha\Big((e,e'),]_i\Big) \oplus [{}_i a \ominus \alpha\Big((e,e'),a\Big),\\ & [{}_i a \in \mathscr{R}, \exists x \in \breve{B}_e \times \breve{B}_{e'}, y \in B_a, \varpi\Big(\alpha(x,y), \alpha(x,({}_i y)_i)\Big) > 0\Big\} \subset \beta_n(\mathscr{G},\mathscr{R}).\end{aligned}$$

We can even define a distribution on $\beta_\varpi(\mathscr{G})$ as the unique value of ϖ,

$$\mathbb{P}\Big(\mathscr{G} \oplus \alpha(e,]_i) \oplus [{}_i a \ominus \alpha(e,a)\Big) = \varpi\Big(\alpha(x,y), \alpha(x,({}_i y)_i)\Big).$$

The split process thus defined will be noted $\mathcal{S}$.

We can complete the definition of split-and-merge process by using the uniform distribution on the merges $\alpha(\mathscr{G})$.

Using this definition, there is a one-to-one correspondence between split processes and a general parsing kernel.

Lemma 3.5.1
Given any string s, and a split $\mathscr{G} \in \beta_n^*(s,\mathscr{R})$, there is a bijection

$$T : \beta_n^*(s,\mathscr{R}) \longrightarrow \Big\{t \in \mathscr{T}, s \in B_t\Big\}.$$

PROOF. Let us take a sequence of splits $s = \mathscr{G}_0, \ldots, \mathscr{G}_n$, $\mathscr{G}_i \in \beta_n(\mathscr{G}_{i-1}, \mathscr{R})$. We can reverse this process by building a sequence $(\breve{\mathscr{G}}_i)$, such that $\varphi\left(\breve{\mathscr{G}}_i\right) = \mathscr{G}_i$. We will first let $\breve{\mathscr{G}}_n = \mathscr{G}_n$, and, given $\breve{\mathscr{G}}_i$, if

$$\mathscr{G}_i = \mathscr{G}_{i-1} \oplus \alpha(e,]_i) \oplus [_i a \ominus \alpha(e, a),$$

put

$$\breve{\mathscr{G}}_{i-1} = \breve{\mathscr{G}}_i \ominus \alpha(e,]_i) \ominus [_i \breve{a} \oplus \alpha(e, ({}_i\breve{a})_i),$$

where $\breve{a}$ is the expression corresponding to a in $\breve{\mathscr{G}}_i$.

This construction verifies the condition $\varphi\left(\breve{\mathscr{G}}_i\right) = \mathscr{G}_i$, so in particular we have that $t = \varphi\left(\breve{\mathscr{G}}_0\right) \in \mathscr{T}$ and $\varphi(t) = s$.

We will now prove that this t does not depend on the actual sequence of splits. This comes from the fact that the surface structure $\varsigma(y)$ of any substring y of t must be in $\mathscr{G}_n$. This means that we can define $T(\mathscr{G}) = t$. All that is left to prove is that this is a bijection.

In order to do so, consider the converse construction. Take a parse tree t. There is a sequence of parse trees $s = t_0, \ldots, t_n = t$, such that there is, for any i, a context $x \in \mathscr{T}^2$, an expression a, $[_i a \in \mathscr{R}$ and a tree $y \in B_a$, with $t_i = \alpha(x, y)$, $t_{i+1} = \alpha(x, ({}_i y)_i)$. This sequence defines a sequence of splits, with the rule

$$\mathscr{G}_{i+1} = \mathscr{G}_i \oplus \alpha(x,]_i) \oplus [_i y \ominus \alpha(x, y),$$

as mentioned before, and the end result $\mathscr{G}$ does not depend on the actual sequence, since

$$\mathscr{G}(t) = \varsigma(t) \oplus \bigoplus_{[_j e \in \mathscr{R}} \chi(t, j, e) \otimes [_j e \tag{3.5.2}$$

This obviously defines the inverse of T. □

Remark that this lemma states also that, as soon as we have a parsing kernel such that

$$\Pi(s, t) > 0 \Leftrightarrow \beta_n(\mathscr{G}(t), \mathscr{R}) = \varnothing, t \succcurlyeq s, \tag{3.5.3}$$

we can simply use this parsing kernel to simulate a split process.

3.5.2 Reversible split and merge process

The results above showed that the (generalized) split and merge process could be used to explore the communication classes of k. However, such a process will probably distort the distribution, and can only be used to study the support of the language. We will here propose a method to build a reversible crossing-over dynamics, using the general Metropolis method described in section 3.D on page 165, using split and merge processes.

We will here work on texts, that is, on sums of Dirac masses at possibly repeated sentences. As such, we will note $\mathfrak{T}$ the set of texts of strings, $\overline{\mathfrak{T}}$ that of texts of parse trees, $\mathfrak{G}$ for texts of expressions, $\bar{\mathfrak{G}}$ for texts of well-parenthesized expressions. The length of texts, that is, their mass, is fixed at n.

We begin with a text $S_{1:n} \in \mathfrak{T}$, and a reference grammar $\mathscr{R} \in \mathfrak{G}$. The basic idea is to split the text according to a parse of its sentences, and then to merge it back, effectively crossing over constituents.

Let us then consider a general parsing kernel $\overline{\Pi}$. We also have the splitting kernel from parsing trees to toric grammars $t \mapsto \mathscr{G}(t)$ defined in eq. (3.5.2) on the facing page. We now want to build a production process that will merge back the elements of $\mathscr{G}$ into a text.

Remark first that any text that can be obtained by regular merges (where we merge any $\lfloor_i \alpha(x, \rfloor_j)$ with any $\lfloor_j y$) can also be obtained by merging only $\lfloor_0 \alpha(x, \rfloor_j)$ with $\lfloor_j y$, in left-to-right order. Indeed, given a merge result, we can simply follow the order of merges given by taking outer parentheses first, in left-to-right order. If we moreover index the global brackets $\lfloor_0$ of the grammar, the order of parentheses are fixed.

Consider now the merge transformations on grammars

$$\Big\{\mathcal{M}_k(\mathscr{G}, \mathscr{G}');\, \mathscr{G}' = \mathscr{G} \ominus \lfloor_0^k \alpha(x, \rfloor_i) \ominus \lfloor_i y \oplus \lfloor_0^k \alpha(x, (_i y)_i)\Big\}.$$

We can take, for example,

$$\mathcal{M}_k\Big(\mathscr{G}, \mathscr{G} \ominus \lfloor_0^k \alpha(x, \rfloor_i) \ominus \lfloor_i y \oplus \lfloor_0^k \alpha(x, (_i y)_i)\Big) = \frac{\mathbb{1}(x_1 \in \breve{D}^*)}{\mathscr{G}(\lfloor_i S^*)},$$

which corresponds to taking the closing brackets from left-to-right order in the kth sentence of $\mathscr{G}$ (with weights), and a matching $\lfloor_i y$.

If we put, for short, $\mathcal{M}(\mathscr{G}) = \mathcal{M}_k(\mathscr{G})$, where the merge is on the first incomplete sentence, that is, $k = \min\{m, \mathscr{G}(\lfloor_0^m \breve{D}^*) = 0\}$, the Markov chain defined by the kernel $\mathcal{M}$ is stationary in finite time, at most $\sum_{i \in \mathbb{N}\setminus\{0\}} \mathscr{G}(\lfloor_i S^*)$, since each steps removes one $\lfloor_i$. We can then define the production process as

$$\mathcal{P} = \mathcal{M}^\infty.$$

The trick of indexing the sentences allows us to have a deterministic order of the merges, at least for the closing brackets, which means that for any two grammars $\mathscr{G}, \mathscr{G}'$, if $\mathcal{P}(\mathscr{G}, \mathscr{G}') > 0$,

$$\mathcal{P}(\mathscr{G}, \mathscr{G}') = \prod_i \frac{\mathscr{G}'(\lfloor_i S^*)!}{\mathscr{G}(\lfloor_i S^*)!}.$$

Remark that the result of $\mathcal{P}$ is not necessarily a text (that is, a grammar with no $\lfloor_i$, $i > 0$ left). We will not explore how we could recover from this failure, and

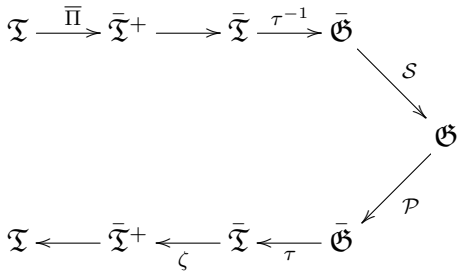
merge back the remaining elements, since we think this would be unnecessarily complicated. We prefer to simply reject the result if $\mathscr{G}'([_+D^+) \neq 0$ and repeat the merge until success. Failures will only happen when the grammar is recursive, and be actually rare unless the grammar is strongly recursive. As we mentioned in section 1.3.2 on page 28, this was indeed a rare occurrence in our experiments.

Since the Metropolis algorithm used to build the reversible dynamics already uses some sort of rejection process, we can even integrate it to the chain using the projection kernel

$$\begin{array}{rccl} \tau: & \bar{\mathfrak{G}} & \longrightarrow & \mathscr{M}_1(\bar{\mathfrak{T}}) \\ & \mathscr{G} & \longmapsto & \begin{cases} \delta_{\mathscr{G}} & \text{if } \mathscr{G} \in \bar{\mathfrak{T}} \\ \delta_{\epsilon^n} & \text{otherwise,} \end{cases} \end{array}$$

and conversely $\tau^{-1}(t) = \delta_t$, $t \in \bar{\mathfrak{T}}$.

We finally obtain the following chain, with which we can define a reversible dynamics on texts, using section 3.D on page 165:



Non-labelled arrows of course represent projections on the last or first element of the string.

Proposition 3.5.2
The dynamics on texts defined by

$$\begin{aligned} &\mathcal{SM}(t,t') = \\ &\quad \sum_{\substack{\tilde{t},\tilde{t}' \in \bar{\mathfrak{T}}^+ \\ \breve{t},\breve{t}' \in \bar{\mathfrak{T}} \\ \breve{\mathscr{G}},\breve{\mathscr{G}}' \in \bar{\mathfrak{G}} \\ \mathscr{G} \in \mathfrak{G}}} \overline{\Pi}(t,\tilde{t}) \mathbb{1}(e(\tilde{t}) = \breve{t} = \breve{\mathscr{G}}) \mathcal{S}(\breve{\mathscr{G}},\mathscr{G}) \mathcal{P}(\mathscr{G},\breve{\mathscr{G}}') \tau(\breve{\mathscr{G}}',\breve{t}') \zeta(\breve{t}',\tilde{t}') \mathbb{1}(t' = b(\tilde{t}')) \\ &\qquad \times \left(\frac{\zeta(\breve{t},\tilde{t})}{\overline{\Pi}(t,\tilde{t})} \frac{\tau(\breve{\mathscr{G}},\breve{t})}{\mathbb{1}(\breve{t} = \breve{\mathscr{G}})} \frac{\mathcal{P}(\mathscr{G},\breve{\mathscr{G}})}{\mathcal{S}(\breve{\mathscr{G}},\mathscr{G})} \frac{\mathcal{S}(\breve{\mathscr{G}},\mathscr{G}')}{\mathcal{P}(\mathscr{G},\breve{\mathscr{G}}')} \frac{\mathbb{1}(\breve{t}' = \breve{\mathscr{G}}')}{\tau(\breve{\mathscr{G}}',\breve{t}')} \frac{\overline{\Pi}(t',\tilde{t}')}{\zeta(\breve{t}',\tilde{t}')} \wedge 1 \right) \end{aligned}$$

is reversible for $\mathbb{P}_S^{\otimes n}$.

PROOF. As always, the expression of $\mathcal{SM}(t,t')$ is symmetric in t,t', so we only have to prove that $\mathbb{P}_S^{\otimes n}(t) = \mathbb{P}_S^{\otimes n}(t')$.

Given any sequence $\tilde{t}, \tilde{t}' \in \bar{\mathfrak{T}}^+, \breve{t}, \breve{t}' \in \bar{\mathfrak{T}}, \breve{\mathscr{G}}, \breve{\mathscr{G}}' \in \bar{\mathfrak{G}}, \mathscr{G} \in \mathfrak{G}$ with a positive contribution to the sum, according to lemma 3.2.1 on page 125,

$$\begin{aligned}
\mathbb{P}_S^{\otimes n}(t) &= \prod_{k=1}^{n} \mathbb{P}_S\Big(B_{\varsigma\left(\breve{t}_k\right)}\Big) q_{B_{\varsigma(\breve{t}_k)}}(t_k) \\
&= \prod_{k=1}^{n} \left(\mathbb{P}_S\Big(B_{\varsigma\left(\breve{t}_k\right)}\Big) A_{\varsigma\left(\breve{t}_k\right)} \prod_{\lfloor_j e \in \mathscr{R}} \Big[A_e q_{B_j}\Big(B_e\Big)\Big]^{\chi(\breve{t}_k, j, e)} \right) \\
&= \prod_{k=1}^{n} \left(\sum_{\lfloor_0^k a \in \mathscr{G}} \mathbb{P}_S\Big(B_a\Big) A_a \prod_{\lfloor_j e \in \mathscr{R}} \Big[A_e q_{B_j}\Big(B_e\Big)\Big]^{\mathscr{G}(\lfloor_j e)} \right) \\
&= \prod_{k=1}^{n} \left(\mathbb{P}_S\Big(B_{\varsigma\left(\breve{t}'_k\right)}\Big) A_{\varsigma\left(\breve{t}'_k\right)} \prod_{\lfloor_j e \in \mathscr{R}} \Big[A_e q_{B_j}\Big(B_e\Big)\Big]^{\chi(\breve{t}'_k, j, e)} \right) \\
&= \prod_{k=1}^{n} \mathbb{P}_S\Big(B_{\varsigma\left(\breve{t}'_k\right)}\Big) q_{B_{\varsigma\left(\breve{t}'_k\right)}}(t'_k) \\
&= \mathbb{P}_S^{\otimes n}(t').
\end{aligned}$$

□

3.6 Estimating the language distribution

Let us remark that if B is a Markov substitute set such that $B \cap \operatorname{supp}(\mathbb{P}_S) \neq \varnothing$, then $B \subset \operatorname{supp}(\mathbb{P}_S)$ and $\mathbb{P}_{S|S\in B} = q_B$. As such, we can aim to identify a collection of Markov substitute sets that will cover the entire distribution, and from their relative frequencies, deduce the entire distribution.

If we put for any $s \in D^+$, $C(s) = \operatorname{supp}(\delta_s \sum_{j=0}^{\infty} k^j)$ the syntactic category of s, then $C(s)$ is a Markov substitute set and

$$\mathbb{P}(S) = \mathbb{E}\Big(q_{C(S)}\Big),$$

so that

$$\widehat{P} = \frac{1}{n} \sum_{i=1}^{n} q_{C(S_i)}$$

is an unbiased estimate of $\mathbb{P}_S$.

This formula can be used to estimate the probability of any sentence, provided we have access to the substitute measures $q_{C(s)}$.

Let us remark here that while

$$\operatorname{supp}(\delta_s k) = \bigcup_{e \in S^+, s \in B_e} B_e$$

is a context-free language (with multiple rewriting rules for the start symbols), since $E_s = \{e \in S^+, s \in B_e\}$ is finite, this is not the case anymore for $C(s)$, because nothing prevents the number of surface structures of strings in $C(s)$ to be infinite. Differently put, the syntactic category of a string s is not necessarily a context-free language, though it is still a countable union of context-free languages

$$C(s) = \bigcup_{e \in E(s)} B_e,$$

with

$$E(s) = \Big\{e \in S^+, \sum_{j=0}^{\infty} k^j(s, B_e) > 0\Big\}.$$

For any fixed, finite set E, $C(E) = \bigcup_{e \in E} B_e$ is still a context-free language, and as such its substitute measure $q_{C(E)}$ may be estimated using the tools presented in section 3.2.2 on page 127. This means that we can estimate $E(s)$ by

$$\widehat{E}(s) = \Big\{e \in S^+, \sum_{j=0}^{N} k^j(s, B_e) > 0\Big\},$$

which is finite and whose substitute measure may be estimated. The error

$$\mathbb{P}_S\left(\bigcup_{s \in \text{supp}(\mathbb{P}_s)} C(s) \setminus \bigcup_{i=1}^{n} C(\widehat{E}(S_i))\right)$$

may be estimated using the missing mass estimator (see [Goo53, MO03]).

Another view is to remark that, for any string s, and any expression $e \in S^+$, such that $s \in B_e$,

$$\mathbb{P}_S(s) = \mathbb{P}_S(B_e) q_{B_e}(s).$$

The substitute measure q_{B_e}, as we have already seen, is easy to estimate. To estimate $\mathbb{P}_S(B_e)$, we want to compute

$$\mathbb{1}(s \in B_e).$$

This function is a very simple parsing test, and is also very easy to compute. Thus we can estimate the probability of a given sentence s by drawing one or better yet, multiple parsing trees $t_1, \ldots, t_m$, $t_k \succcurlyeq s$, which gives several expressions $e_k = \varsigma(t_k)$, and set

$$\widehat{\mathbb{P}}(s) = \frac{1}{m} \sum_{k=1}^{m} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(S_i \in B_{e_k})\, q_{B_{e_k}}(s).$$

We thus obtain an estimator without bias of $\mathbb{P}(s)$, on the condition that we use a split sample scheme to learn the substitute measures on a different sample.

This approach can also be used as a grammaticality test. In this test we can choose not to use the substitute measures $q_{B_{e_k}}$, since we know they must have by construction a full support. We can indeed define the set of global structures of the language as

$$\{e \in S^+, \mathbb{P}(B_e) > 0\}.$$

We can then consider the set of maximal surface structures $\mathscr{E}_g$ for the inclusion relation $B_{e'} \subset B_e$. The support of $\mathbb{P}_S$ can then be characterized by the property

$$\mathbb{P}_S(s) > 0 \iff \exists e \in \mathscr{E}_g, s \in B_e.$$

For any string $s \in D^+$, the set $\mathscr{E}_g(s) = \mathscr{E}_g \cap \{e, s \in B_e\}$ is easy to compute from parsing, and $\mathscr{E}_g$ can then easily be estimated by

$$\widehat{\mathscr{E}_g} = \bigcup_i \mathscr{E}_g(s),$$

and the error

$$\mathbb{P}_S\left(\bigcup_{e\in\mathscr{E}_g} B_e \setminus \bigcup_{e\in\widehat{\mathscr{E}_g}} B_e\right)$$

may be once again estimated with the missing mass estimator.

3.A Support of the split and merge process and substitutions

We will study in this section the action of the split and merge process on the syntactic classes.

Let us consider a grammar $\mathscr{R}_0$, a corresponding reversible dynamics k_0, and its associated equivalence relation $\sim_0$. These can obviously be extended to texts. Take a compatible reference grammar $\mathscr{R}$, and let SMP be a split-and-merge process with reference $\mathscr{R}$, as defined in section 3.5.1 on page 151. We take a compatible grammar, and not simply the same grammar, for the first result.

Proposition 3.A.1
For any pair of texts $(t, t') \in \mathfrak{T}$,

$$SMP(t, t') > 0 \Longrightarrow t \sim_0 t'.$$

In particular, the split and merge process stays in the support of the language.

PROOF. We will focus in this proof on a single sentence, since the substitute process is independent on the sentences. Let us take a sentence s, and its result, s'. We may consider s' as a syntactic tree $\breve{s}'$, given by the merge operation sequence, and we can to the same with s, its structure $\breve{s}$ being given by the parse algorithm used to split. The only common point between $\breve{s}$ and $\breve{s}'$ is their surface structure.

We want to prove that substitutions can link any two $\mathscr{R}$-correct syntagms $\breve{s}$ and $\breve{s}'$, as soon as they share the same surface structure.

Since $\mathscr{R}$ is compatible, any two trees in the same $\breve{B}_i$ are equivalent. Since $\breve{s}$ and $\breve{s}'$ share the same global structure, any tree inside a pair of outer parentheses of $\breve{s}$ has a counterpart in $\breve{s}'$, and each may be substituted to its counterparts in $\breve{s}'$, which gives the result. □

The converse is obviously not true. The simple compatibility is not sufficient anymore, since the null grammar is compatible with any grammar, but has a trivial split and merge process. So let us now suppose that $\mathscr{R}$ defines the same equivalence relation. Under no further hypothesis, this is still not sufficient, since we need at least all elements of B_i to be in the text to hope to obtain them after a pass of the split and merge process.

However, if we define a slightly different split and merge process, we can get a converse result.

Definition 3.A.1
We will define a generalized split and merge process $GSMP$ as the result of a split process, followed by a merge from the obtained global expressions, using the elements of $\mathscr{R}$, without any restriction on the number of uses. Such a merge will be noted GMP.

The actual distribution of the merges may be taken as uniform on all possible merges, but since we are here only interested in supports, the actual distribution is not crucial.

Obviously, such a process is independent on each sentence of the text, and proposition 3.A.1 on the preceding page is still true with this new process.

Such a definition, however, is still not sufficient to have the reverse implication, as shown in the following example. Consider the reference grammar (compatible with itself)

$$\begin{array}{l} [_1\alpha \oplus [_1 ab \oplus \\ [_2\gamma \oplus [_2 bc. \end{array}$$

Obviously, $k^2(\alpha c, a\gamma) > 0$. However,

$$GSMP(\alpha c, \cdot) = \frac{1}{2}\delta(\alpha c) + \frac{1}{2}\delta(abc),$$

so that $GSMP(\alpha c, a\gamma) = 0$.

Nonetheless, we have the following, if k is the invariant dynamics defined from $\mathscr{R}$.

Proposition 3.A.2
For any pair of sentences (s, s'),

$$k(s, s') > 0 \Longrightarrow GSMP(s, s') > 0,$$

and as such,

$$\sum_k GSMP^k(s, s') > 0 \Longleftrightarrow s \sim s'.$$

This means that, more than staying in the support of the language, the generalized split and merge process will visit all sentences in each of the communication classes of k, provided it begins in it.

Proof. The proof is actually quite straightforward: if $k(s, s') > 0$, there are two $t, t' \in B_e$ for a certain expression e, and two chains $\tilde{t}$ and $\tilde{t}'$ from s to t and s' to t' respectively so that $\Pi(s, \tilde{t}) > 0$ and $\zeta(\tilde{t}', s') > 0$. Obviously then $\mathcal{S}(s, e) > 0$, and $GMP(e, s') > 0$, so $GSMP(s, s') > 0$. □

3.B Parsing using a ruleset

We will now describe how to build the different parses we used in chapter 3. In order to do so, we will use a well-known parsing algorithm for context-free grammar, the CYK algorithm.

The version of the CYK algorithm we will (briefly) outline here is that proposed in [LL09], which works on a grammar in the Binary Normal Form, that is, a grammar for which all right-hand sides of rules are of length at most 2.

Let us then take a set of context-free rules $(D,]_R, \mathscr{R})$, where the rule index set $R \subset \mathbb{N}$ is finite. Its size $|\mathscr{R}|$ is by definition the length of the concatenation of all the rules in $\mathscr{R}$. Its transformation to its binary normal form $\mathscr{R}'$ may be obtained in time $O(|\mathscr{R}|)$, and the resulting grammar is of size $O(|\mathscr{R}|)$, and the size of its non-terminal set is also $O(|\mathscr{R}|)$. The transformation is obtained by replacing any rule of length more than 2, $[_i \omega_1 \dots \omega_n$ by the rules

$$[_i \omega_1]_{i_1}; \dots; [_{i_{j-1}} \omega_j]_{i_j} \dots; [_{i_{n-1}} \omega_{n-1} \omega_n.$$

Let us suppose that we also built the unit graph

$$U = \Big\{ (]_a, e), e \in D \cup]_+, ([_a e \in \mathscr{R} \Big\},$$

which is built in time and space $O(|\mathscr{R}|)$, simply by adding the correct edges for each unit rule. This makes the computation of $U^*(]_a)$ (the set of reachable states from $]_a$ in the graph U) possible in time $O(|\mathscr{R}|)$.

Given a string $s_{1:n}$ to parse, the algorithm will actually compute the sets

$$T_v = \{x \in D \cup]_+, v \in B_x\}$$

for all substring v of s. These sets will be stored in a table T of size n^2. Each entry in this table is itself a boolean array indexed by pairs $(]_i, k)$, where $i \in R$ and $k \in [\![0, n]\!]$ (so the array is of size $|R| \times (n+1)$), and whose diagonal is filled with the terminal symbols $s_1, \ldots, s_n$.

The table is then filled so that $T(i,j)(]_a, 0) = 1$ when there is one symbol b such that $s_{i:j} \in B_a \cup B_b$, $[_a b \in \mathscr{R}$, and $T(i,j)(]_a, k) = 1$ when there are two symbols $b, c \in D \cup]_+$, such that $s_{i:k} \in B_b$, and $s_{k+1:j} \in B_c$, where $[_a bc \in \mathscr{R}$. In particular, $T(i,j)(]_a, k) = 1$ for some k means that $s_{i:j} \in B_a$.

The table is then filled so that $s_{i:j} \in B_a$, if, and only if, $s_{i:k} \in B_{e_1}$, and $s_{k+1:j} \in B_{e_2}$, where $[_a e_1 e_2 \in \mathscr{R}$ if, and only if, $T(i,j)(]_a, k) = 1$.

This is done by the following algorithm, where all cells in T are initially set to 0:

```
for i=1,...,n do
  T(i,i)+=s_i
  T(i,i)+=(U*(T(i,i)),0)
done
for j=2,...,n do
  for i=1,...,j-1 do
    for k=i,...,j-1 do
      for all rule [a XY do
        if T(i,k)(X,*)=1 and T(k+1,j)(Y,*)=1 then
          T(i,j)+=(a,k)
        done
      done
    done
    T(i,j)+=(U*(T(i,j)),0)
  done
done
```

with the notation `T(i,j)+=A` meaning that we change to 1 the value of each cell of $T(i,j)$ whose index is in A. Again, `U*(T(i,j))` is the set of symbols produced by the unitary rules of U whose left-hand side is equal to `T(i,j)` (seen as a list of symbols). On the other hand, `T(i,k)(X,*)=1` is the test that at least for one r, `T(i,k)(X,r)=1`.

This algorithm is in $O(n^4|\mathscr{R}|)$ time, and we could make it $O(n^3|\mathscr{R}|)$ (classical complexity) if we do not need to keep the index k. However, the k is necessary to be able to draw a certain parse from the table, which we'll need in section 3.B.2 on the next page. The space requirement is $O(n^3|\mathscr{R}|)$, $O(n^2|\mathscr{R}|)$ if we do not need the k.

With such a table, we can now present our parsing algorithms. As mentioned before, we are working on rulesets, and not complete context-free grammars. We are missing the start symbol. This means that we have two different approaches to the parsing. In one case, we seek a parsing kernel Π_e that stays inside the context-free language $\breve{B}_e$, which is the classic framework. In the other, we are not in a context-free language, but in the whole $\mathscr{T}$. We do not have, in this case, a start symbol (or rather, we have an infinity of them).

3.B.1 Parsing general syntagms

Parsing in $\mathscr{T}$ does not require the surface structure of the result to be of a particular form. This means that we can build the parse bottom-up.

Let us begin with the observation that any expression e such that $t \in B_e$ can be described by a succession of cells $(T(i_k, j_k, r_k) > 0)_{1 \leqslant k \leqslant K}$, so that $i_{k+1} = i_k + j_k$, $i_0 = 1$, $i_K + j_K = \ell(s)$. Indeed, in this case, $e = \rfloor_{i(r_1)} \dots \rfloor_{i(r_K)}$. The lengths j_k are additional informations, corresponding, for $s \preccurlyeq t \in \breve{B}_e$, to the lengths of the contents of each outer parentheses of t.

Definition 3.B.1
For any parse tree $t \in \mathscr{T}$, let us put

$$E(t) = \Big\{ t' = \alpha(x, ({}_i y)_i), (x, y) \in \mathscr{T}^3, t = \alpha(x, y), \varsigma(y) \in \mathscr{R} \Big\}.$$

The general parsing extension is defined as

$$\varpi(t, t') = \frac{\mathbb{1}(t' \in E(t))}{|E(t)|}.$$

Proposition 3.B.1
The parsing kernel Π defined from the general parsing extension kernel is such that

$$\operatorname{supp}(\Pi(t, \cdot)) = \Big\{ t' \in \mathscr{T}, \varphi(t') = \varphi(t), E(t') = \varnothing \Big\},$$

that is, Π charges all the maximal parses of t.

Proof. It is quite obvious that if $\Pi(t, t') > 0$, $t' \succcurlyeq t$ and $E(t') = \varnothing$.

Conversely, let us prove that all maximal parses are charged by Π. Let us take such a parse t'. Consider the sequence of trees $t = t_0 \succcurlyeq \dots \succcurlyeq t_n = t'$, where we add the parentheses of t' one pair at a time, following a left-to-right, bottom-up path. It is quite obvious that $\varpi(t, t') > 0$ and $\operatorname{supp}(\varpi(t', \cdot)) = \varnothing$, so $\Pi(t, t') > 0$.□

As before, we will be more interested in the general parsing kernel $\overline{\Pi}$ defined from ϖ, which keeps track of the whole sequence of trees built using ϖ.

The set $E(t)$ is quite easy to compute from the parse table of $\varphi(t)$, so the simulation and computation of $\overline{\Pi}(s,\cdot)$ is easy.

For the definition of reversible dynamics, we also need a converse kernel ζ that removes parentheses. Given a tree $T \in \mathscr{T}$, writing $s = \varphi(T)$, we will define the parsing extensions relative to T.

Definition 3.B.2
For any parse tree $t \in \mathscr{T}$, let us put

$$B_T(t) = \Big\{ t' = \alpha(x, ({}_i y)_i), (x,y) \in \mathscr{T}^3, t = \alpha(x,y), \varsigma(y) \in \mathscr{R}, T \succcurlyeq t' \Big\}.$$

The general parsing extension is defined as

$$\rho_T(t,t') = \frac{\mathbb{1}(t' \in B_T(t))}{|B_T(t)|}.$$

Once again, the set $B_T(t)$ is quite easy to compute from the parse table of $\varphi(t)$, so the simulation and computation of $\zeta(T,\cdot) = \rho_T^\infty(T,\cdot)$ is easy.

The Metropolis algorithm will consider, for a sequence of trees $\tilde{t}_i$, $0 \leqslant i \leqslant m$ the ratios

$$\frac{\varpi(\tilde{t}_i, \tilde{t}_{i+1})}{\rho_{\tilde{t}_m}(\tilde{t}_i, \tilde{t}_{i+1})} = \frac{|E(\tilde{t}_i)|}{|B_{\tilde{t}_m}(\tilde{t}_i)|} \mathbb{1}(\tilde{t}_{i+1} \in E(\tilde{t}_i)).$$

While these ratios are not necessarily close to 1, they do not depend on $\tilde{t}_{i+1}$ (apart the obvious requirement $\tilde{t}_{i+1} \in E(\tilde{t}_i)$). The actual values of these ratios depend on the ambiguity of the grammar, and a sentence for which only one maximal parse exists will have all the ratios equal to 1.

3.B.2 Parsing inside syntagms of certain type

Section 3.2.3 on page 129 required that we had a parsing Π_e inside of a particular language B_e. This is actually the main goal of standard parsing algorithms such as CYK, but we need to go a little further, for we need an easy way to draw such a parse, and to compute $\Pi_e(s,t)$ for any string s and tree t. This can easily be done by keeping in the table cell $T(i,j)$ pointers to the table cells $T(i,i+k)$ and $T(i+k,j)$ used to fill it, which can be done by adding k to the indices of the array in $T(i,j)$, as described before.

We have to consider, for any expression e, the set of incomplete parse trees from e, that is

$$T_e = \Big\{ t \in \breve{D}^*, \exists t' \in \breve{B}_e, t' \succcurlyeq t \Big\}.$$

Definition 3.B.3
For any incomplete parse tree $t \in T_e$, let us put

$$D(t) = \Big\{ t' = \alpha(x, ({}_k({}_i y)_i({}_j y')_j)_k), x \in \mathfrak{e} \times \breve{D}^*, y, y' \in D^+, \\ t = \alpha(x, ({}_k y y')_k), \exists ({}_i \breve{y})_i \in B_i, ({}_j \breve{y}')_j \in B_j, \\ [{}_k]_i]_j \in \mathscr{R}, \varphi(\breve{y}) = y, \varphi(\breve{y}') = y' \Big\}.$$

The parsing extension from e is defined as

$$\varpi(t, t') = \frac{\mathbb{1}(t' \in D(t))}{|D(t)|}.$$

It is quite obvious that if $t \in T_e$, $D_t \subset T_e$. The set $D(t)$ is, again, easy to compute from the table, on the condition that pointers are kept as described above.

The parsing kernel may then be defined as

$$\Pi_e(s, t) = \varpi^\infty(({}_0 s)_0, t) \\ = \exp\left(- \sum_{\substack{x \in (\breve{D}^*)^2, \\ y \in \mathscr{T}, k \in \mathbb{N}}} \mathbb{1}\Big(t = \alpha(x, ({}_k y)_k)\Big) \log\Big(\Big| D\Big(\alpha(x, ({}_k \varphi(y))_k)\Big)\Big|\Big)\right).$$

3.C Building crossing-over tree kernels

The construction of reversible dynamics defined in section 3.2.5 on page 134 requires a crossing-over tree kernel $\Big(\mathfrak{s}(t, t'), t, t' \in \widetilde{\mathfrak{T}}\Big)$. We will present here one way of building it.

In order to do so, it will be useful to index the parentheses of tree texts. We will then use a new parenthetized dictionnary $\dot{\breve{D}} = D \cup \{(_i^k,)_i^k, i \in \mathbb{N}, k \in \mathbb{N}\}$. The i will still be called labels, and the new k, indices. We build as before the set of trees $\dot{\mathscr{T}}$, with the additional requirement that matching parentheses have the same index, and that no two opening or closing parentheses have the same index. We of course define all the other sets as before, in particular $\dot{\widetilde{\mathfrak{T}}} = \dot{\mathscr{T}}^n$.

For any text $t \in \widetilde{\mathfrak{T}}$, let $t^{\mathbb{1}}$ be the same tree with the parentheses indexed in the order of the opening ones (left-to-right, depth first order). For any indexed tree text $\dot{t} \in \dot{\widetilde{\mathfrak{T}}}$, any permutation $\sigma \in \mathfrak{S}_m$, where m is the number of parentheses in $\dot{t}$, let us write as $\dot{t}^\sigma$ the text where the pair of parentheses in $\dot{t}$ with index k is instead indexed by $\sigma(k)$, that is, if $\dot{t}_i = \alpha(x, (^k y)^k)$, $\dot{t}_i^\sigma = \alpha(x, (^{\sigma(k)} y)^{\sigma(k)})$. It is quite obvious that $(t^\sigma)^{\sigma'} = t^{\sigma \circ \sigma'}$, and we will expand this operator on non-indexed tree texts with the notation $t^\sigma = (t^{\mathbb{1}})^\sigma$.

For any indexed tree text $\dot{t}$ with m pairs of parentheses, let $\phi(t)$ be the tree itself, without parenthesis indices, and $\sigma_{\dot{t}} \in \mathfrak{S}_m$ be the permutation such that $\dot{t} = \phi(\dot{t})^{\sigma_{\dot{t}}}$.

Let us begin by defining an indexing kernel

$$\mathfrak{I}(t, \dot{t}) = \frac{1}{m!}, \qquad t \in \bar{\mathfrak{T}}, \dot{t} \in \dot{\bar{\mathfrak{T}}}, \dot{t} = t^\sigma, \sigma \in \mathfrak{S}_m.$$

Consider, for any indexed tree text $\dot{t} \in \dot{\bar{\mathfrak{T}}}$, the sets

$$\begin{aligned}
A_K(\dot{t}) = \Big\{ & \dot{t}' \in \dot{\bar{\mathfrak{T}}}, \\
& \dot{t}_u = \alpha(x, ({}^k_j y)^k_j), \dot{t}_v = \alpha(x', ({}^{k'}_j y')^{k'}_j), \dot{t}'_u = \alpha(x, ({}^{k'}_j y')^{k'}_j), \dot{t}'_v = \alpha(x', ({}^k_j y)^k_j), \\
& u, v \in \mathbb{N}, j \in \mathbb{N} \setminus \{0\}, x, x' \in \left(\dot{\mathring{D}}^*\right)^2, y, y' \in \dot{\mathscr{T}}, \sigma_{\dot{t}}^{-1}(k') \geqslant \sigma_{\dot{t}}^{-1}(k) = K \Big\}, \\
B_K(\dot{t}) = \Big\{ & \dot{t}' \in \dot{\bar{\mathfrak{T}}}, \\
& \dot{t}_i = \alpha\Big(\alpha(x, ({}^k_j y)^k_j), ({}^{k'}_j y')^{k'}_j\Big), \dot{t}'_i = \alpha\Big(\alpha(x, ({}^{k'}_j y')^{k'}_j), ({}^k_j y)^k_j\Big), \\
& i \in \mathbb{N}, j \in \mathbb{N} \setminus \{0\}, x, x' \in \left(\dot{\mathring{D}}^*\right)^2, y, y' \in \dot{\mathscr{T}}, \sigma_{\dot{t}}^{-1}(k') \geqslant \sigma_{\dot{t}}^{-1}(k) = K \Big\}, \\
C_K(\dot{t}) = {} & A_K(\dot{t}) \cup B_K(\dot{t}).
\end{aligned}$$

We can now define the individual crossing-over kernels

$$\mathfrak{s}_K(\dot{t}, \dot{t}') = \frac{\mathbb{1}\left(\dot{t}' \in C_K(\dot{t})\right)}{|C_K(\dot{t})|}.$$

These kernels simply swap the contents of the Kth pair of parentheses (that has index $\sigma(K)$) with the contents of a pair of parentheses (with same label) further in the text (but not inside the parentheses). We could add one to the numerator if we wanted to allow no crossing-over to take place (and complete the kernel by giving a positive probability to $\dot{t} = \dot{t}'$).

We can now define the crossing-over kernel $\mathfrak{s} = \mathfrak{s}_1 \circ \ldots \circ \mathfrak{s}_m$. Simulation of $\mathfrak{s}$, as seen, is easy. Moreover, it is apparent from the definition that for any pair of indexed tree texts $\dot{t}, \dot{t}'$, $\mathfrak{s}(\dot{t}, \dot{t}') = \mathfrak{s}(\dot{t}^\sigma, \dot{t}'^\sigma)$.

Hence, to calculate $\mathfrak{s}(\dot{t}, \dot{t}')$, we can go back to the case where $\sigma_{\dot{t}} = \mathbb{1}$. In this case, if $\mathfrak{s}(\dot{t}, \dot{t}') > 0$, the pair of parentheses that were swapped with the kth pair in $\dot{t}$ was that of index $\sigma_{\dot{t}'}(k)$, and computing $\mathfrak{s}(\dot{t}, \dot{t}')$ is easy.

It is quite easy to check that $\mathfrak{s} \circ \mathfrak{I}$ verifies the eq. (3.2.4) on page 134.

The invariant dynamics is then

$$K\Big(s_{1:n}, s'_{1:n}\Big) = \sum_{t,t'\in\bar{\mathfrak{T}}^+, \dot{t},\dot{t}'\in\dot{\bar{\mathfrak{T}}}} \overline{\Pi}\Big(s_{1:n}, t\Big)\mathfrak{I}(e(t), \dot{t})\mathfrak{s}\Big(\dot{t}, \dot{t}'\Big)\mathbb{1}(\mathfrak{I}(e(t)', \dot{t}') > 0)\zeta\Big(t', s'_{1:n}\Big) \times \left(\frac{\zeta\Big(t, s_{1:n}\Big)}{\overline{\Pi}\Big(s_{1:n}, t\Big)} \frac{\mathfrak{I}(e(t)', \dot{t}')}{\mathfrak{I}(e(t), \dot{t})} \frac{\mathfrak{s}\Big(\dot{t}, \dot{t}'\Big)}{\mathfrak{s}\Big(\dot{t}', \dot{t}\Big)} \frac{\overline{\Pi}\Big(s'_{1:n}, t'\Big)}{\zeta\Big(t', s'_{1:n}\Big)} \wedge 1\right).$$

(remember that $e(t)$ is the end tree of the sequence of trees t)

Remark that $\mathfrak{s}(\dot{t}, \dot{t}') > 0$ implies that $\dot{t}$ and $\dot{t}'$ have the same number of parentheses, so that $\frac{\mathfrak{I}(e(t)', \dot{t}')}{\mathfrak{I}(e(t), \dot{t})} = 1$.

3.D Building general Metropolis reversible dynamics

The two reversible dynamics defined in section 2.2 on page 69 use a standard Metropolis algorithm, with a single intermediate step, while those in sections 3.2.4 and 3.2.5 on page 131 and on page 134 use multiple intermediate steps. We will here present a general method to define reversible dynamics, using as many intermediate steps as needed.

Let $A_1, \dots, A_n$ be a sequence of sets, and consider accordingly a sequence of Markov kernels $\pi_i \in \mathscr{M}_1(A_i)^{A_{i-1}}$, $\pi'_i \in \mathscr{M}_1(A_{i-1})^{A_i}$ and $\zeta \in \mathscr{M}_1(A_n)^{A_n}$.

$$A_0 \longrightarrow A_1 \longrightarrow \dots \longrightarrow A_n \longrightarrow A_n \longrightarrow \dots \longrightarrow A_1 \longrightarrow A_0$$

$$s_0 \xrightarrow[\pi_1]{} s_1 \xrightarrow[\pi_2]{} \dots \xrightarrow[\pi_n]{} s_n \xrightarrow[\zeta]{} s'_n \xrightarrow[\pi'_n]{} \dots \xrightarrow[\pi'_2]{} s'_1 \xrightarrow[\pi'_1]{} s'_0$$

For any $S = (s_0, s_1 \dots, s_n) \in \prod_{i=0}^n A_i$, let $\check{S} = (s_n, \dots, s_1, s_0)$.

Consider a distribution $\mathbb{P}$ on A_0, and a function $q : \prod_{i=0}^n A_i \times \prod_{i=n}^1 A_i \to \mathbb{R}$ such that, for any $S, S' \in \prod_{i=0}^n A_i$

$$\prod_{i=0}^n \pi_i(S_{i-1}, S_i)\zeta(S_n, S'_n)\pi'_i(S'_i, S'_{i-1}) > 0 \implies \mathbb{P}(S_0)q(S, \check{S}') = \mathbb{P}(S'_0)q(S', \check{S}). \tag{3.D.1}$$

Proposition 3.D.1
Under eq. (3.D.1) *on the preceding page, the Markov kernel* $(K(s, s')s, s' \in A_0)$

defined by

$$K(s,s') = \sum_{\substack{s_i,s_i'\in A_i \\ 0\leqslant i\leqslant n}} \mathbb{1}(s=s_0, s'=s_0')q(s_0,s_1,\dots s_n,s_n',\dots,s_1',s_0') \prod_{i=0}^{n}\pi_i(s_{i-1},s_i)\pi_i'(s_i',s_{i-1}')\left(\prod_{i=0}^{n}\frac{\pi_i'(s_i,s_{i-1})}{\pi_i(s_{i-1},s_i)}\cdot\frac{\pi_i(s_{i-1}',s_i')}{\pi_i'(s_i',s_{i-1}')}\wedge 1\right),$$

is reversible with respect to $\mathbb{P}$.

PROOF. Simply remark that

$$\mathbb{P}_S(s)K(s,s') = \sum_{\substack{s_i,s_i'\in A_i \\ 0\leqslant i\leqslant n}} \mathbb{P}(s_0)q(s_{0:n},s_{n:0}')\mathbb{1}(s=s_0,s'=s_0') \times\left(\prod_{i=0}^{n}\pi_i'(s_i,s_{i-1})\cdot\pi_i(s_{i-1}',s_i')\wedge\prod_{i=0}^{n}\pi_i(s_{i-1},s_i)\cdot\pi_i'(s_i',s_{i-1}')\right)$$

is obviously symmetric in s, s'. □

Chapter 4

Conclusion

We would like here to comment on a few possible uses of the objects presented in this work. We will focus on language analysis based on a corpus of independent texts $S^i_{1:n}$, followed by some considerations on the scope of our model, and on possible extensions and open questions.

4.1 Outline of possible uses of Markov substitute sets in natural language processing

The central notion of this work is that of Markov substitute sets, which corresponds to the notion of syntactic constituents in linguistics. Chapter 2 proposes some ways of testing whether a certain set is a Markov substitute set.

The first step of our language analysis from a statistical sample is to build the reference grammar, or a Markov ruleset, which is roughly equivalent to identifying a collection of Markov substitute sets.

We begin by initializing a collection of Markov substitute sets

$$\mathscr{B}_0 = (\{w\}, w \in S),$$

and their substitute measures $q_{\{w\}} = \delta_w$, which would correspond to the trivial reference grammar

$$\mathscr{R}_0 = \bigoplus_i 1 \otimes [_i w_i.$$

We then use the sample to make the reference grammar grow, in the way described in section 3.4 on page 144. This construction is recursive, using at each step a new sample (to avoid dealing with dependency issues). Each iteration works as follows:

– Test as many rules as possible that we could add to the reference grammar, using for example the simultaneous test proposed in section 2.7 on page 96, and add those that passed the test,

– Saturate the resulting reference grammar,

– Estimate the substitute measures for each category of rules, as proposed in section 3.2.2 on page 127.

Each step will use the parsing kernels corresponding to the current reference grammar, whose constructions are described in section 3.B on page 159.

When we arrive at a point where no more Markov substitute sets can be identified, or when we do not have anymore samples, or when a missing mass estimator tells us that the grammar we built so far produces surface structures covering almost all of the support of the language distribution, we stop.

The resulting reference grammar can then give us some information on the syntactic structure of the language, and help us define various reversible dynamics on it.

We saw three main types of reversible dynamics. The first one, the most powerful, but with an unknown off-model behaviour, is the substitute dynamics k, using knowledge of the substitute measures to replace instances of them in the initial sample. The second one is the crossing-over dynamics K, that avoids the necessity of knowing the substitute measures at the price of keeping the same number of each word as in the initial sample. When applied to a number of replicas of the statistical sample growing to infinity, the crossing-over dynamics computes the maximum likelihood estimator in a linear exponential family, and therefore has a provable off-model robust behavior. The last dynamics, the split and merge process, is quite similar to the crossing-over dynamics, and uses toric grammars to deconstruct a text in its smaller syntactic components and rebuild it.

Knowledge of a Markov reference grammar gives us parsing tools, that permit us to analyse any string, and, ultimately, to estimate its probability in the context-free language of sentences of the same type B_e. Estimating the frequencies of each type of sentences gives us finally a estimator of the whole language.

The support of the estimated language,

$$\widehat{\mathscr{L}} = \bigcup_{e \in \widehat{\mathscr{E}_g}} B_e,$$

is potentially larger than a context-free language, as a union of a possibly infinite number of such languages. However, with a finite sample, we will never be

able to obtain more than a finite number of surface structures. Nonetheless, the dynamics k defines a slightly bigger support for the estimator of the language,

$$\widehat{\mathscr{L}} = \bigcup_{i=1}^{n} C(S_i),$$

which may be larger than a context-free language, though still recursively enumerable. The actual position of $C(s)$ in the hierarchy of formal languages is still an open question.

The split and merge process was also used in chapter 1 to define a communication model, which defines another notion of language, as the invariant measure on each class of the communication chain.

If the reference grammar is fixed, we saw that each text was recurrent, and thus that the language could be obtained back, without error, by a Monte-Carlo simulation, from any text on its communicating class.

However, we could ask ourselves what happens when the reference grammar is learnt anew at each generation. We saw that if the reference grammar is indeed Markov for the language, the split and merge process stays in its support (and is even at least weakly reversible). The communicating classes could nonetheless be fragmented if the reference grammar is too poor.

In our study, we supposed that our tests for the Markov substitute property were never wrong (at least, had a false acceptance probability close to zero). This is obviously optimistic, and it would be interesting to study more precisely the distortions introduced to the language when the reference grammar is not Markov.

Some other uses of the model can also be considered. An interesting possibility is to define sub-models, focused on a certain type of texts, or topics, by restricting the Markov substitute sets learnt in a large corpus to the expressions appearing in a more specialized one.

Estimating the substitute measures on a smaller corpus could also be an interesting tool, for example to obtain a statistical description of the style of a given author.

4.2 Further considerations on the scope of the model

The structure of the model presented in this work appears as an extension of the conditional independence assumptions defining Markov random fields.

We saw that a $\mathscr{B}$-Markov process can be parametrized as an exponential family, with some sort of Markov field structure on a graph on D^+. Two vertices are neighbours if they differ only by two members of a Markov substitute set, the

ratio of the weights depending only on the non-common part of the two nodes:

$$\frac{\mathbb{P}_S(\alpha(x,y))}{\mathbb{P}_S(\alpha(x,y'))} = \frac{q(y)}{q(y')}.$$

In contrast to the description of the potential function of a Markov random field, depending on a decomposition on cliques, we proposed here a description in terms of minimal "free" edge types and anchor point on each connected component. The number of free edges reflects the number of free parameters in the choice of the substitute measures.

The edge structure considered here was defined using the Markov substitute property, however, we could also obtain linear exponential families using a different definition of the edges, that is of the probability ratios between pairs of states that are constrained to be identical.

We could in particular add restrictive conditions on the context. We could also constrain the probability ratio of states that are linked by a relation that is not a substitution, but that is based on a different type of transformation, for instance a movement of the form $\alpha(\alpha(x,\epsilon),y) \sim \alpha(\alpha(x,y),\epsilon)$.

We proved that the Markov substitute model is the thermodynamic limit of a crossing-over dynamics. This crossing-over dynamics can be interpreted as a communication process, which can be used to transmit a language distribution from speaker to speaker in an error free way, using a limited amount of memory, compared to the size of the support of the language.

This crossing-over dynamics is reversible with respect to the uniform measure on memorized sentences. The fact that the language can be maintained from one speaker to another by simple knowledge of the grammar, that is, the collection of Markov substitute sets (without their substitute measures), hints that language can be "stored" quite efficiently. This could give some insight for the "poverty of stimulus" problem put forward by linguists. In other words, you do not need to learn the parameters (the substitute measures), you need only to learn the model (the set of substitute sets), and then, from a relatively small amount of memorized sentences, you can generate a huge number of new sentences using random crossing-overs, while outputting sentences with a probability distribution close to the model distribution using the optimal model parameters (in terms of likelihood based on what you have heard from others).

To come back to the two steps of the estimation of the language:

- learning the model (i.e. a generating set of Markov substitute sets),
- learning the parameters (the substitute measures),

the second step is simply solved by the crossing-over dynamics.

Various scenarios can be imagined for learning the first step, ranging from theoretical answers based on simultaneous tests, as described in this thesis, or on penalized likelihood criteria available in a more generic way for collections of exponential families, to more supervised learning scenarios, where a teacher, supervising the output of a crossing-over dynamics, raises a red flag each time a wrong sentence is produced (allowing the pupil to cancel the Markov substitute pair involved in the crossing-over from its list of Markov substitute sets). However, children seem to already know the language when they first go to school. Studies in psycholinguistics, especially in children, hints that language learning is mostly based on positive information, and negative feedback is, in the first stages at least, virtually non-existent. Nonetheless, some sort of supervised school-like training could be a way to refine (and homogenize) the model selection step of language learning.

Bibliography

[Bak79] James K Baker. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132, 1979.

[Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.

[CG98] Zhiyi Chi and Stuart Geman. Estimation of probabilistic context-free grammars. *Computational linguistics*, 24(2):299–305, 1998.

[Chi99] Zhiyi Chi. Statistical properties of probabilistic context-free grammars. *Computational Linguistics*, 25(1):131–160, 1999.

[Cho56] Noam Chomsky. Three models for the description of language. *Information Theory, IRE Transactions on*, 2(3):113–124, 1956.

[Cho65] Noam Chomsky. *Aspects of the Theory of Syntax*. Number 11. MIT press, 1965.

[Cho95] Noam Chomsky. *The minimalist program*, volume 28. Cambridge Univ Press, 1995.

[Cla14] Alexander Clark. Learning trees from strings: A strong learning algorithm for some context-free grammars. *Journal of Machine Learning Research*, 14:3537–3559, 2014.

[CS12] Shay B. Cohen and Noah A. Smith. Empirical risk minimization for probabilistic grammars: Sample complexity and hardness of learning. *Computational Linguistics*, 38(3):479–526, 2012.

[DPDPG+94] Stephen Della Pietra, Vincent Della Pietra, John Gillett, John Lafferty, Harry Printz, and Luboš Ureš. *Inference and estimation of a long-range trigram model*. Springer, 1994.

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications: Volume One*. John Wiley & Sons, third edition, 1968.

[GK07] Thomas L. Griffiths and Michael L. Kalish. Language evolution by iterated learning with bayesian agents. *Cognitive Science*, 31(3):441–480, 2007.

[Goo53] Irving J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264, 1953.

[LL09] Martin Lange and Hans Leiß. To CNF or not to CNF? an efficient yet presentable version of the CYK algorithm. *Informatica Didactica*, 8:2008–2010, 2009.

[LY90] Karim Lari and Steve J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56, 1990.

[McA98] David A. McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234. ACM, 1998.

[McA99] David A. McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170. ACM, 1999.

[McA03] David A. McAllester. Pac-bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003.

[MO03] David McAllester and Luis Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *The Journal of Machine Learning Research*, 4:895–911, 2003.

[Nor98] James R. Norris. *Markov chains*. Cambridge university press, 1998.

[Roa01] Brian Roark. Probabilistic top-down parsing and language modeling. *Computational linguistics*, 27(2):249–276, 2001.

[Sta09] Edward P. Stabler. Mathematics of language learning. *Histoire, Épistémologie, Langage*, 31(1):127–145, 2009.

[Tom85] Masaru Tomita. *Efficient parsing for natural language: a fast algorithm for practical systems*. Kluwer Academic Publishers, 1985.

[Tom87] Masaru Tomita. An efficient augmented-context-free parsing algorithm. *Computational linguistics*, 13(1-2):31–46, 1987.

[TZZW12] Ming Tan, Wenli Zhou, Lei Zheng, and Shaojun Wang. A scalable distributed syntactic, semantic, and lexical language model. *Computational Linguistics*, 38(3):631–671, 2012.

[You67] Daniel H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and control*, 10(2):189–208, 1967.