Some contributions to computational Bayesian methods with application to phylolinguistics

Grégoire Clarté

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Université Paris Dauphine

Quelques contributions aux statistiques Bayesiennes, numériques et appliquées.

Soutenue par

Grégoire CLARTÉ

Le 6 octobre 2021

École doctorale n°543

SDOSE

Spécialité

Mathématiques

Composition du jury :

Mme. Antonietta MIRA Professeur, Università della Svizzera Italiana & Università dell'Insubria	*Rapporteuse*
M. Alexandre BOUCHARD-CÔTÉ Professeur Associé, University of British Columbia	*Rapporteur*
Mme. Judith Rousseau Professeur, University of Oxford	*Présidente*
M. Geoff NICHOLLS Professeur, University of Oxford	*Examinateur*
M. Pierre Jacob Professeur, ESSEC Business School	*Examinateur*
M. Christian ROBERT Professeur, Université Paris Dauphine & Warwick University	*Directeur de thèse*
M. Robin RYDER Maître de Conférences, Université Paris Dauphine	*Directeur de thèse*

Dauphine | PSL UNIVERSITÉ PARIS

Some contributions to computational Bayesian methods with application to phylolinguistics

Grégoire Clarté

October, 6th 2021

Remerciements

Je voudrais tout d'abord remercier mes directeurs de thèse Christian et Robin, qui m'ont soutenu pendant ces trois ans, et qui ont été deux grands modèles et sources d'inspiration pendant ma thèse. Je n'aurais bien évidemment rien fait de tout cela sans eux.

I would also like to thank the reviewers Antonietta and Alexandre for having accepted to review my PhD and for their useful comments ; and the members of the Jury, that have accepted despite the travel issues to participate : Judith, Geoff and Pierre.

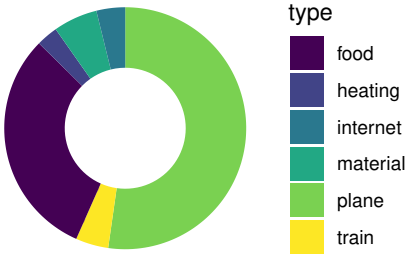
Je remercie aussi mes coauteurs, Julien, Carlo, Natasha, Jean, Antoine, Adrien et Luke, j'ai vraiment aimé travailler avec vous et j'espère bien continuer à le faire.

Ces trois ans à Dauphine ont été parmi les meilleurs, grâce à la merveilleuse ambiance de Dauphine. Je remercie tous les membres du labo pour leur accueil et leur sympathie, en particulier Vincent qui est l'exemple même de la diplomatie, Jean D. pour avoir participé à nos travaux sur l'environnement, César, Isabelle et Marie pour leur aide indéfectible, Alessandra, François, et tous les autres. Merci aux doctorants de m'avoir choisi comme représentant, par choisi je veux bien évidemment dire qu'aucun autre ne voulait le faire, j'ai été très heureux de manger de boire un café le midi avec vous. Merci à Jean C. pour avoir subi mes plaintes, je t'ai sans doute fait perdre un an de travail au total, à Louis car tu es un bon sujet de discussion, à Quentin même si tu n'étais pas souvent là, à Fabio qui chante si bien, à Ruihua pour les thés quand on ne voulait pas travailler, à Peter et Thinh pour les discussions, à Armand parce qu'on à fait quand même du bon boulot (même si on est surtout bons avec les bombes). Merci aussi à tous les autres précaires du CEREMADE, dont Donato et Giovanni avec qui j'ai bien sué, Kathi ma co-élue, Charly, Lorenzo et Théo les nouveaux qui doivent continuer la grande tradition consistant à perdre 2h tous les midis, à Jeanne et Tristan, vous qui tracez la voie. Merci à tous les autres que je n'ai pas cités.

Merci à mes amis non cités avant, Léa, Léa, Candice, Owen, Glen, Vlad, Aude, Mercedes, David, Lise, Anna, et tous ceux que j'aurai oublié, c'est toujours bien de pouvoir se plaindre à plusieurs.

Merci enfin à ma famille et en particulier à mes parents qui m'ont subi pendant tout le premier confinement.

Emission	$kgCO_2eq$ emitted
5000 cups of tea	0.003
3 kg of tea	12
17000 km by plane	5440
22800 km by train*	456
food	3200
heating†	300
books	10
paper	5
computer	600
internet use†	400



source: ADEME *base carbone*.
* this includes my daily use of the RER, for around 60%;
†overestimation, very few data is available.

FIGURE 0.1 – Carbon emissions associated with the realisation of the present work.

On the Carbon footprint of this work

Figure 0.1 summarises and represents the carbon emissions associated with the realisation of the present work. Noticeably, these emissions are higher than the sustainable values.

Contents

Part 1
Résumé en français

Ce travail résulte de la concaténation de plusieurs contributions aux statistiques bayésiennes. Dans la première, nous proposons un modèle d'évolution jointe pour le lexique et la phonologie des langues, afin de reconstruire une phylogénie des langues. Dans la deuxième nous introduisons ABC à la Gibbs (ABC-Gibbs) une méthode bayésienne approchée, pour modèles à vraisemblance intractable et en grande dimension. Nous prouvons sa convergence et son efficacité sur plusieurs exemples. La dernière s'intéresse à une autre méthode numérique que nous appelons Monte-Carlo Collectif, où des particules en interaction reproduisent le comportement d'une chaine de Markov non linéaire.

1 Familles de Langues

Au XVIIIème siècle, la linguistique comparée en est venue à observer que les certaines langues du monde partageaient de hauts degrés de similarité dans le vocabulaire, la grammaire et la phonologie. Ces similarités sont patentes lorsque l'on compare par exemple le vocabulaire courant des langues Romanes. En comparant les langues européennes et indo-iraniennes, les linguistes en ont inféré l'existence d'une grande famille, la famille Indo-Européenne, associée à un ancêtre commun hypothétique et reconstitué, le Proto-Indo-Européen. L'intérêt croissant durant le XIXème siècle pour les études Indo-Européennes n'était pas exempte d'arrières-pensées nationalistes. La polémique récente lancée par le livre de Jean-Paul Demoule a permis de montrer que l'existence d'un ancêtre commun à une famille de langues n'est pas nécessairement lié à l'existence d'un ancêtre génétique, ou culturel.

De nos jours, les linguistes ont décrit de nombreuses familles de langues : Indo-Européennes, Sino-Tibétaines, Finno-Ugriennes, *etc.* Pour quelques unes de ces familles, plusieurs sous-familles ont été distinguées, celles-ci sont quelque fois attestées par des sources écrites (par exemple la famille Sinitique, ou Romane) mais le plus souvent seules les langues contemporaines sont disponibles pour définir ces familles (par exemple la famille K'iche'). Pour définir ces familles, les spécialistes se basent le plus souvent sur des comparaisons qualitatives, sans description précise de modèle d'évolution, à la manière des familles linéennes, même si quelques avancées ont été faites dans ce domaine : modèle de diffusion par vagues [Ross et al., 1988], introduction de la charge fonctionnelle [Martinet, 1970]). Même de nos jours, alors que l'histoire des langues indo-européennes est particulièrement bien connue, de nombreuses familles de langues restent peu étudiées. Parmi ces

langues, les langues des signes sont parmi les plus délaissées.

L'objectif de la linguistique historique est de reconstruire l'histoire d'une famille de langues depuis son ancêtre commun jusqu'aux langues actuelles, en explicitant les relations entre chaque langue. Cette histoire est usuellement représentée sous la forme d'un arbre phylogénétique daté, dont la reconstruction est un problème complexe. Cette reconstruction nécessite en général de quantifier l'évolution des langues, et nécessite l'adjonction d'informations diverses — par exemple d'ordre archéologique — pour calibrer le modèle. Par exemple de grands évènements historiques, comme la domestication de certaines espèces peuvent se traduire dans le vocabulaire [Diamond and Bellwood, 2003].

Le premier essai d'étude quantitative pour dater un ancêtre commun à deux langues est connu sous le nom de *glottochronologie*, introduite par Swadesh [1952] ; cette méthode repose sur l'hypothèse que le vocabulaire évolue à une vitesse fixée de telle sorte que le nombre de mots en commun entre le vocabulaire (ici résumé aux mots de la liste Swadesh) mesure le temps pendant lequel les langues ont évolué depuis leur séparation. Cette méthode souffre de nombreux défauts, en particulier la vitesse d'évolution du modèle, qui n'est pas identifiable, a été fixée *a priori*, et reste fixe sur la totalité de l'arbre.

Plus récemment, des méthodes phylogénétiques, inspirées par la biologie, ont été appliquées aux données linguistiques.

2 Phylogénie des langues

La phylogénie est l'étude des relations évolutives entre des objets, caractérisés par des traits, descendant d'un même ancêtre commun. Cette approche a, dans un premier temps, été appliquée à l'étude de séquence d'ADN, permettant la construction d'arbres phylogénétique pour les espèces, et ainsi de se passer de la classification Linnéenne dépassée. Ces méthodes peuvent aussi être utilisées pour étudier des traits phénotypiques. De par la nature des données, de longues séquences d'ADN qui doivent d'abord être alignées et comparées, il est paru que l'usage de méthodes computationelles était particuièrement idoine ; d'autant plus que les progrès des méthodes statistiques ont permis de traiter des modèles de plus en plus complexes [Warnow, 2017, Yang, 1994]. Le parallèle entre linguistique et génétique est douteux, en particulier car les langues présentent dans leur évolution des phénomènes que les gènes ignorent: bilinguisme, emprunts, intervention humaine dans l'évolution. Cependant, la structure arborescente au cœur de l'analyse phylogénétique semble largement acceptée dans la communauté linguistique, en tout cas concernant les grandes échelles de temps.

L'usage de méthode numériques pour la linguistique est une addition récente. Dans le papier de Ringe et al. [2002], les auteurs calculent, pour un jeu de don-

nées constitué de divers traits (phonologiques, grammaticaux, syntactique) pour plusieurs langues indo-européennes, une phylogénie aussi proche que possible d'une phylogénie parfaite, c'est à dire sous laquelle les observations seraient parfaitement possible. Cependant, cette méthode n'est pas un traitement statistique des données au sens où aucun modèle statistique n'est proposé et où aucune mesure de l'incertitude de la reconstruction n'est possible. De plus, le jeu de données mélangeait des données de diverses sortes, souvent non comparables. Depuis, les linguistes se sont concentrés sur des jeux de données lexicaux.

Un jeu de donnée lexical est le plus souvent constitué de classe de cognats. Une classe de cognats est un ensemble de mots de même sens ayant un ancêtre commun. Dans Gray and Atkinson [2003], les auteurs proposent de reconstituer des arbres en se basant sur un modèle "restriction site", c'est à dire un modèle où les classes de cognats apparaissent et disparaissent avec des taux différents inconnus. Dans l'article fondateur [Nicholls and Gray, 2007], les auteurs, inspirés par Dollo [1887], ont modélisé l'évolution des langues au travers de l'évolution de leurs classes de cognats. L'évolution du lexique est donc modélisée par l'apparition et disparition de classes de cognats, suivant un processus ponctuel. Le modèle permet un calcul relativement efficace de la vraisemblance au travers d'une méthode récursive [Felsenstein, 1981], et les auteurs prouvent l'efficacité de la méthode par son application aux langues indo-européennes. Cette méthode, ses améliorations subséquentes, et les autres méthodes qu'elle a inspirées, ont été appliquées avec succès à de nombreux exemples: langues dravidiennes [Kolipakam et al., 2018], sino-tibétaines [Sagart et al., 2019], pama-nyungan [Bowern and Atkinson, 2012], ou austronésiennes [Gray et al., 2009]. Parmi les améliorations apportées, on peut citer Ryder and Nicholls [2009] dans lequel les auteurs ajoutent la possibilité de traiter des données manquantes, en intégrant sur les valeurs possibles des données manquantes. Cela a permis d'ajouter dans les jeux de données des langues mortes parcéllairement connues, par exemple le *Louvite* dans la famille indo-européenne, tout en gardant un coût computationel raisonnable. Une autre amélioration, [Ryder, 2010], a constitué en l'adjonction d'anisotropie dans le modèle, en ajoutant des *catastrophes*, c'est à dire des moments d'intense évolution. Plus récemment, le problème des emprunts a été traité par Kelly [2016], en ajoutant la possibilité pour un mot d'être latéralement emprunté. Cette dernière amélioration augmente cependant considérablement le temps de calcul, même si une implémentation efficace a permis d'appliquer la méthode aux langues polynésiennes orientales [Kelly and Nicholls, 2017]. D'autres modèles, toujours basés sur les classes de cognats ont permis d'inférer non seulement les relations historiques mais aussi les relations géographiques entre les langues. Par exemple, les récents résultats sur les langues austronésiennes [Bouckaert et al., 2018] ont permis de localiser la zone dans laquelle devait être parlé l'ancêtre commun de cette famille de langues.

Les classes de cognats restent donc au coeur des modèles actuels. Le principal problème de ces méthodes se trouve dans la constitution des jeux de données: les listes de cognats sont construites à la main, par des linguistes, au cours d'un processus long, complexe et sujet à caution et débat, le plus souvent ce processus repose sur des informations de différentes sortes, phonétiques ou grammaticales. Ces informations sont à leur tour elles-même basées sur des modèles linguistiques plus complexes. Il y a donc des risques que ces données soient biaisées. De plus, les linguistes doivent recueillir des formes phonétiques complètes pour ne garder qu'une information de cognacité. Des méthodes automatiques de constitution de classes de cognats ont été tentées [List et al., 2017]. Précédemment, Bouchard-Côté et al. [2013] avait supposé la phylogénie connue pour apprendre les classes de cognats. Parmi les modèles proposés, List et al. [2016] repose sur le calcul d'une distance de type Levenshtein sur les formes phonétiques écrites en alphabet phonétique international (API), la distance dépend alors de la proximité entre les phonèmes ; dans Rama [2018b], les auteurs proposent de se baser sur des modèles non paramétriques, de type processus du restaurant chinois, pour inférer des classes de cognats.

Tous ces modèles reposent sur la similarité entre les mots ; contrairement aux séquences d'ADN, la phonologie présente quelques particularités. La plus frappante d'entre elles est la régularité des transformations phonologiques dans de nombreuses reconstructions [Lavie, 2007], c'est à dire que la plupart des changements phonologiques s'appliquent uniformément dans le vocabulaire, *e.g.* du latin vulgaire au français, toutes les voyelles finales ont disparu [Joly, 1995]. Ces changements sont de plus très dépendents du contexte. Finalement, le nombre de phonèmes existant dans une langue est très contraint, obéissant à des règles encore peu connues [de Boer, 1997, Dunbar and Dupoux, 2016]. À notre connaissance, peu de travaux se sont intéressés à ce dernier aspect, si ce n'est un certain nombre de remarques dans Bouchard-Côté et al. [2013].

Parmi les modèles s'attachant à décrire la phonologie, Bouchard-Côté et al. [2013] propose de modéliser les évolutions phonologiques comme des transformations qui touchent indépendamment chaque mot, vu comme une chaîne de caractères. Sur une phylogénie connue, chaque mot est modifé par l'application de transduceur de chaîne, qui permet une prise en compte de l'influence du contexte. Les transformations les plus fréquentes — spiration, lénition, etc.— sont correctement reconstituées, et cette méthode permet également de reconstituer des classes de cognats en mesurant une similarité entre les mots. Les auteurs proposent qu'une méthode plus avancée basée sur la leur permettrait d'inférer jointement les évolutions phonologiques, lexicales, et la phylogénie.

Les méthodes automatiques d'inférence des classes de cognats ouvrent de nouvelles perspectives en particulier parce qu'elles réduisent considérablement le pré-

traitement à la main des données, d'autant que pour certaines familles de langues aucun travail de la sorte n'a encore été mené. Dans Rama [2018a], les auteurs constatent qu'en se basant sur de telles reconstructions automatiques de cognats, les résultats restent consistant avec les méthodes habituelles.

Ici, nous nous intéresseront aux langues des signes. Parlées partout sur Terre, les langues des signes restent particulièrement peu étudiées en comparaison de leurs contreparties orales. Alors que l'histoire de l'éducation et de la culture sourde sont connues, les processus évolutifs des langues et leurs liens historiques restent méconnus. Dans Power et al. [2020], les auteurs reconstruisent un *Neighbour net* en se basant sur une distance définie à l'occasion pour représenter les similarités entre les langues de signes. Le jeu de données utilisé se constituait des alphabets manuels, supposés être parfaitement exempts d'iconicité — même si celle-ci est claire dans les signes représentant les lettres *n, m, u, v, w* et *z*, qui imitent la forme de la lettre. Chaque signe est représenté par un certain nombre de traits décrivant la position, forme et mouvement de la main lors de la prononciation du signe. Ces traits sont à la base de l'étude de la phonologie des langues des signes.

3 Méthodes numériques pour l'inférence des phylogénies

L'inférence de phylogénie se fait généralement dans un contexte bayésien. La paradigme bayésien [Robert, 2007], introduit indépendemment par Pierre-Simon de Laplace et Thomas Bayes au XVIIIème siècle, permet une grande consistance dans l'inférence statistique. Dans ce paradigme, l'incertitude sur une quantité θ est représentée par une distribution de probabilités ; en conséquence, il est possible de mettre à jour les informations sur une quantité par l'usage de la formule de Bayes, transformant ainsi une loi *a priori* $\pi(\theta)$ en une distribution *a posteriori* $\pi(\theta \mid X)$, résumant l'information *a prior* et l'information apportée par les observations X, représentées par la vraisemblance $L(X \mid \theta)$:

$$\pi(\theta \mid X) \propto \pi(\theta) L(X \mid \theta).$$

En pratique, cela signifie qu'en calculant la loi a posteriori on mesure directement l'incertitude liée à l'inférence. Il est habituel de s'intéresser à des quantités de la forme $E_{\pi(\theta|X)}[f(\theta)]$. En de rares cas, il est possible de calculer exactement ces quantités, mais le plus généralement le coût computationnel de l'estimation de ces quantités est rédhibitoire. L'algorithme de Metropolis–Hastings [Robert and Casella, 1999] a constitué en ce domaine une révolution. D'abord introduit dans Metropolis et al. [1953], il repose sur l'idée de construire une chaîne de Markov dont la mesure stationnaire serait la loi recherchée. Pour compenser

ses défauts notables — mauvaises propriétés de mélange, choix des paramètres — les dernières années ont vu le développement de variantes de cette méthode. En dehors des versions adaptatives, Roberts and Rosenthal [2009], qui cherchent à apprendre les paramètres optimaux de l'algorithme en temps réel, d'autres méthodes ont été proposées. Les méthodes de tempérance de distribution cherchent à modifier la cible pour la rendre plus facile à explorer. En aplanissant une distribution, les barrières de potentielles deviennent franchissables pour un algorithme de Metropolis–Hastings. Cette méthode est à la base des algorithmes de *Parallel tempering* [Swendsen and Wang, 1986]. L'idée de base de la méthode est, partant d'une distribution simple et connue — par exemple le prior — de passer graduellement à la cible pour laquelle les propriétés de mélange sont mauvaises. On introduit alors autant d'étapes de tempering que nécessaire. Parmi les méthodes de tempérance on peut penser au tempering thermodynamique, qui est une des méthodes les plus simples, on introduit alors des distributions tempérées de la forme $\pi_t \propto \pi(\theta)L(X \mid \theta)^t,\ 0 \leq t \leq 1$, lorsque $t = 0$ nous retrouvons le prior, et pour $t = 1$ le posterior. Chaque distribution est associée à une chapine de Markov évoluant indépendamment des autres, sauf à quelques instants auxquels on propose à deux chaines voisines d'échanger leurs états. Cet échange est assez probablement accepté, de par la proximité des chaîne, et cela permet aux chaînes mélangeant inefficacement de bénéficier des meilleures propriétés des autres. Cet algorithme a l'avantage d'être efficacement parallélisable [Syed et al., 2019], en implémentant une chaîne par cœur il est possible d'utiliser des CPU de taille moyenne.

L'échantillonage d'importance est une autre branche majeure des méthodes de Monte Carlo, basé sur cette simple identité [Robert and Casella, 2013] :

$$\int f(x)\mathrm{d}x = \int f(x)h(x) \cdot \frac{\mathrm{d}x}{h(x)}.$$

Il est possible de réécrire cette identité pour des variables aléatoires. Lorsque l'on veut estimer une quantité de la forme $E_p[f(X)]$, on peut utiliser une variable aléatoire X de loi q différente, pour peu qu'on ajoute un terme correctif:

$$E_p[f(X)] = E_q[f(X)p(X)/q(X)],$$

le choix de q, la distribution d'importance est le principal problème. Pour résoudre cette difficulté, plusieurs méthodes ont été avancées. En particulier l'échantillonnage séquentiel d'importance, reposant sur la même idée de tempérance (*c.f. supra*), représenté par la méthode de Monte Carlo séquentiel (SMC) [Del Moral et al., 2006] a montré son efficacité sur des modèles où la suite de distributions tempérées est évidente, par exemple les modèles spatio-temporels avec des applications notables aux problèmes de filtrage. En plus des méthodes de type Monte-Carlo à particules(PMC), SMC peut être un outil efficace pour l'étude de distribution classiques ;

l'utilisateur doit alors choisir avec discernement ou inspiration la suite de distributions tempérées. Les méthodes SMC ont été utilisées par exemple pour l'inférence de phylogénies [Bouchard-Côté, 2014, Wang et al., 2015, 2019]. Dans ces articles, les auteurs proposent d'étendre le modèle à des forêts d'arbres et construisent par coalescence une suite de distribution tempérées sur les arbres. Plus précisément, dans Bouchard-Côté [2014] les auteurs s'intéressent à la validité de leur méthode, et dans Wang et al. [2019] ils étendent la loi de proposition pour augmenter l'efficacité de la méthode au prix d'une augmentation de la complexité numérique. Les méthodes particulaires, telles PMC et SMC reposent sur la grande quantité de particules, qui assure la convergence et la validité de la méthode [Del Moral, 2004]. À notre connaissance l'usage de GPU pour ces méthodes n'est pas répandu.

4 Phylogénies des langues des signes

Le modèle que l'on propose pour les langues des signes repose sur deux procédés d'évolution. Le premier représente l'évolution des mots, avec l'application de lois phonologiques régulières, et le second l'apparition de nouveaux mots, représentant un renouvellement du vocabulaire.

Plus précisément, chaque lexique est représenté comme une matrice U, chaque ligne représente un *sens* et chaque colonne un trait. Chaque sens est alors associé à un mot constitué d'un unique signe caractérisé par une certaine valeur sur chacun des traits. La matrice U est initialisée à la racine en tirant chaque ligne indépendamment depuis une distribution π_0 définie. U coïncide alors avec les données aux feuilles, c'est à dire aux langues observées. La matrice U est modifiée par deux processus stochastiques :

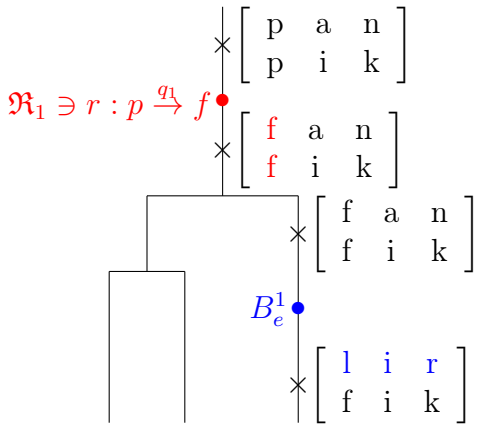
— Un processus de création, qui reproduit l'apparition de nouveaux mots, que ce soit par emprunt, création ou changement de sens ;

— un processus de mutation qui représente l'évolution lente des signes, par l'application de changements phonétiques.

Ces deux processus sont décrit comme des processus de Poisson indépendants sur l'arbre g. Nous représentons en figure 1.1 ces processus et leurs effets sur un petit vocabulaire, nous avons mis des valeurs phonologiques issues des langues orales pour une plus ample compréhension.

Le processus de création consiste en l'apparition, pour un sens donné, d'un nouveau mot. C'est à dire au renouvellement complet de tous les traits pour ce sens. Ce processus n'a pas d'équivalent dans la littérature, mais il est similaire au modèle Mk [Lewis, 2001]. À un processus de création, toutes les valeurs d'une ligne de U sont modifiées. Cela advient avec un taux μ, indépendamment pour chaque

FIGURE 1.1 – Stochastic evolution processes



sens. Le nouveau mot est tiré depuis π_0. Notons également que l'apparition d'un nouveau mot correspond à la création d'une nouvelle classe de cognats.

Le processus de mutation correspond à l'application séquentielle de lois d'évolutions tirées dans un ensemble $\mathfrak{R}_k$ définit par les modélisateurs, spécifique à chaque caractère. À un évènement de mutation, plusieurs éléments d'une colonne de U sont changés. C'est un nouvel ajout par rapport aux précédents modèles de phylolinguistique, car nous voulons simuler le processus d'évolution naturel de la représentation phonologique des mots, décrite par des lois d'évolutions. Pour chaque caractère k, un processus de Poisson avec taux λ_k donne la position des transformations, qui sont alors choisies indépendamment avec probabilité p_k depuis l'ensemble $\mathfrak{R}_k$. Lors d'une mutation, chaque mot est touché par cette transformation avec probabilité β_k indépendamment des autres mots. Cela permet de prendre en compte le fait que le contexte peut changer l'application de certaines lois.

Pour résumer, le modèle est décrit par les paramètres suivants :

— g la topologie de l'arbre phylogénétique, d'ensemble de branches E ;

— l'ensemble des longueurs de branches associées $(\ell_e)_{e\in E}$;

— μ le taux de renouvellement par sens ;

— λ_k le taux d'apparition des transformations pour chaque caractère ;

— p_k un vecteur de probabilités indiquant la fréquence de chaque transformation pour chaque caractère ;

— β_k la probabilité par sens qu'une transformation soit appliquée ;

— ν l'intensité du bruit.

Pour prendre en compte les incertitudes, nous réalisons l'inférence dans un cadre bayésien.

Le jeu de données sur lequel nous nous sommes basés a dû être réduit pour pouvoir être traité numériquement. Originellement, les signes sont représentés par 24 traits, nous avons, en accord avec les linguistes réduit ces traits à seulement 5 décrivant la forme de la main, sa position, son orientation, le mouvement associé et le nombre de mains nécessaire à sa réalisation.

Un autre problème de ce jeu de données repose sur le nombre de valeurs possibles pour les traits, un nombre trop élevé de modalités étant trop difficilement gérable par la mémoire de de l'ordinateur. Les linguistes ont donc réduit ces valeurs à un maximum de 9. Par exemple, la forme de la main ne prend plus en compte que le nombre de doigts tendus et l'état du pouce, ce qui réduit à 9 le nombres de valeurs possibles pour le trait *forme de la main*.

Nous avons été capables de calculer la vraisemblance de ce modèle, et de simuler un échantillon depuis le posterior associé en utilisant une méthode de type SMC. Les résultats montrent sur données simulées des résultats très satisfaisants, avec de bonnes reconstructions de la phylogénie et des variables latentes.

4.1 Résultats sur données réelles

Concernant les données réelles, il faut signaler deux problèmes qui sont particulièrement présents dans l'étude des langues de signes : l'iconicité et les emprunts entre les langues.

L'iconicité correspond à la perte d'indépendance entre le signifié et le signifiant, au sens de De Saussure [1989]. Alors que pour les langues orales, on peut considérer que la plupart des mots ont une forme qui n'a aucun rapport avec leur sens — par exemple, il n'y a aucun lien entre le mot *chat* et l'animal qu'il représente — à quelques exceptions près — chat se traduit en mandarin TIPAmā o, et en Lao TIPAmǣo. La présence de mots iconiques a deux conséquences, la première est que ces mots ne suivent pas l'évolution standard des autres mots. En effet, si le mandarin subissait la modification phonologique $m \rightarrow p/\#\_$, le lien entre signifié et signifiant serait perdu, et donc il est fort probable que les mots iconiques ne subissent que des modifications qui préserveront l'iconicité. La seconde conséquence est que plusieurs langues non liées par ailleurs — typiquement mandarin et lao — partagent des mots proches. Pour les langues des signes de très nombreux signes sont iconiques, car issus du mime ou reprenant des éléments particuliers de ce qu'ils décrivent. Par exemple, les signes désignant les lions font très souvent référence à leur crinière. À cela s'ajoute le problème que parfois c'est une partie du signe seulement qui est iconique, par exemple les signes signifiant *se rencontrer* impliquent presque toujours les deux mains et un mouvement convergent de celles ci, sans que la forme des mains soit imposée.

Les emprunts sont un autre problème particulièrement criant pour l'étude des langues des signes. Les communautés de signeurs sont souvent de faible taille, et les écoles pour sourds sont rarement le fait des communautés elles-mêmes, mais plutôt le résultat de l'arrivée dans la communauté de professeurs venus d'autres écoles, comme c'est le cas pour les États-Unis où la première école pour sourds a été fondée par un Français. Le résultat est que les langues des signes sont souvent issues d'un mélange de langues locales et de langues importées. De plus, les efforts d'uniformisation des langues de signes et l'intercompréhension possible entre plusieurs d'entre-elles encouragent les emprunts de mots. Le résultat est le même que pour l'iconicité, des relations apparentes non historiques.

Aucun de ces deux problèmes n'est pris en compte dans notre modèle. Nous présentons en Figure 1.2 deux arbres consensus résumant les distributions a posteriori sur les arbres lors de deux réalisations indépendantes de l'algorithme. Le jeu de données était constitué des langues asiatiques et européennes. Nous avons contraint les langues asiatiques à former un sous groupe distinct des langues européennes, cela correspondant à l'hypothèse habituelle formulée par les linguistes.

Les résultats sont stables et confirment un certain nombre d'hypothèses. La première est l'existence d'un sous groupe Est-européen, autour du cercle d'influence de l'URSS. Les langues les plus occidentales ont une position moins certaine, confirmant en cela leur lien avec les langues des signes germaniques. Les langues d'Europe occidentale sont moins clairement classées, on constate néanmoins la proximité entre la langue des signes américaine et française, confirmant en cela l'histoire. Du côté des langues asiatiques, la proximité entre la langue des signes de Taiwan et du Japon est faiblement supportée, malgré l'histoire commune de ces deux pays.

Une étude plus approfondie des résultats, en particulier des lois *a posteriori* des paramètres, montre des comportements très différents d'un caractère à l'autre, avec certains — comme le nombre de mains — évoluant particulièrement lentement, nous n'avons a posteriori presque aucun changement régulier associé à ce caractère, ce qui signifie que caractère, bien qu'évoluant, ne s'insère pas dans les hypothèses évolutives du modèle. En cela, nous confirmons l'hypothèse que ce caractère porte vraisemblablement une dimension icônique.

En ce qui concerne les âges internes, de nombreuses incertitudes demeurent, nous présentons en Figure 1.5 les résultats. Pour les langues des signes Japonaises et Taiwanaises, la date semble correspondre à l'annexion Japonaise de l'île. Concernant la LSF et la LSI, cet âge est contraint par la contrainte mise sur la langue des signes Américaines, mais la date semble correspondre à la date du *Risorgimento* à la fin du XIXème siècle.

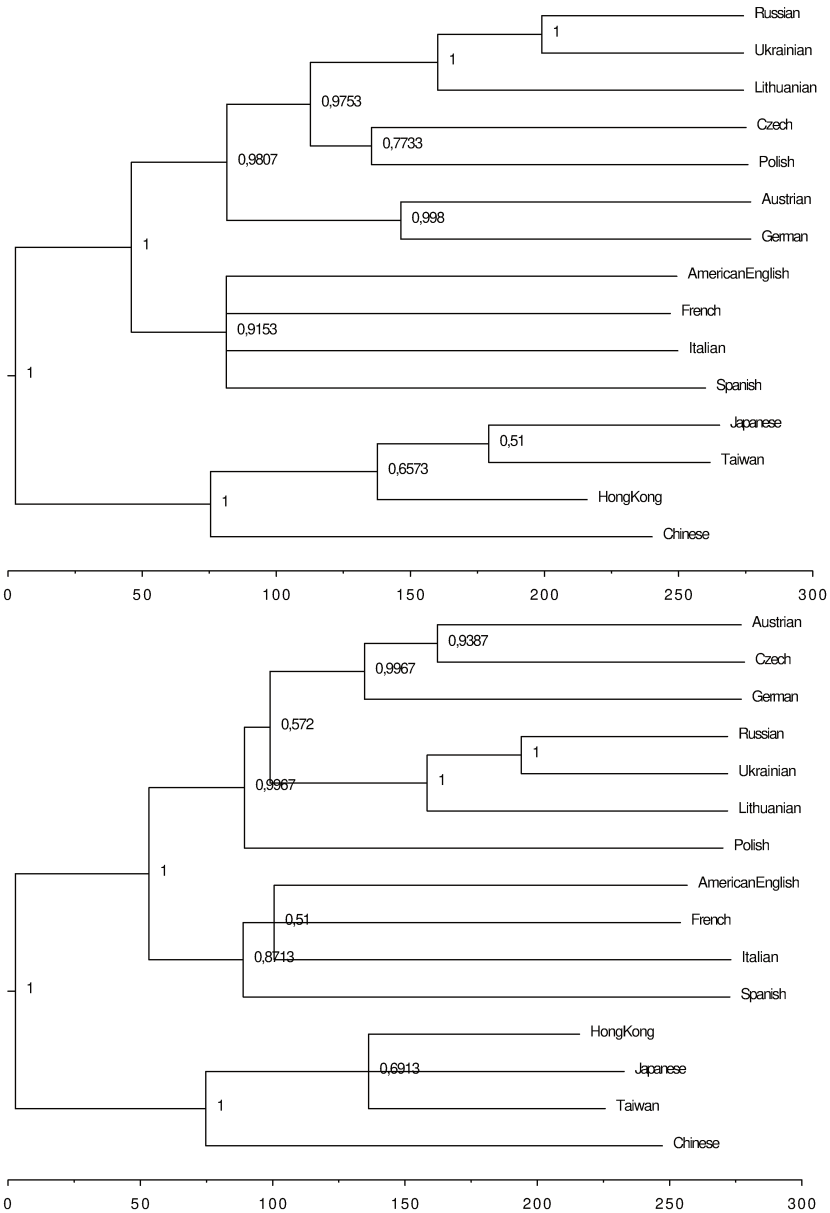


FIGURE 1.2 – Deux arbres consensus pour les langues asiatiques et européennes..

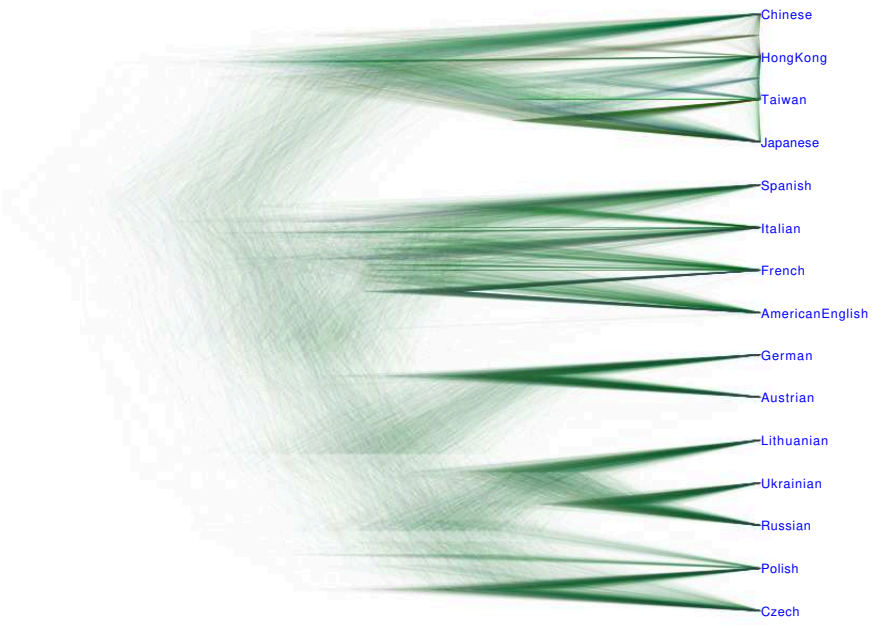


FIGURE 1.3 – Densitree de l'échantillon *a posteriori*.

5 Méthodes bayésiennes approchées

Malgré les capacités croissantes du matériel informatique, et les efforts des mathématiciens, de nombreux modèles développés pour des applications réelles, sont trop complexes pour être étudiés par des méthodes requérant le calcul de la vraisemblance. En particulier lorsque celle-ci repose sur de nombreuses variables latentes en grande dimension. De nombreux modèles proposés par les linguistes sont dans ce cas, en particulier en incluant des emprunts, et la variabilité au cours du temps des paramètres d'évolution.

D'un autre côté, les méthodes approchées, ne reposant pas sur le calcul de la vraisemblance, paraissent être une solution possible. Ces méthodes requièrent seulement de pouvoir simuler des pseudo-observations depuis le modèle génératif proposé, ce qui est indubitablement plus simple que de calculer une vraisemblance. Parmi ces méthodes, les méthodes ABC (approximate Bayesian computation), d'abord introduites en biologie et ensuite étudiées théoriquement et validées par exemple dans Tavaré et al. [1997], Beaumont et al. [2002], Toni et al. [2008], Csilléry et al. [2010], Moores et al. [2015] et Sisson et al. [2018], sont parmi les plus simples. L'idée des méthodes ABC peut être synthétisée en quelques mots : si un paramètre, tiré depuis son prior, génère des pseudo-observations proches des vraies observations, ce paramètre devrait être une réalisation du posterior. Il ne reste plus qu'à interpréter le terme "proche", c'est à dire à trouver une distance

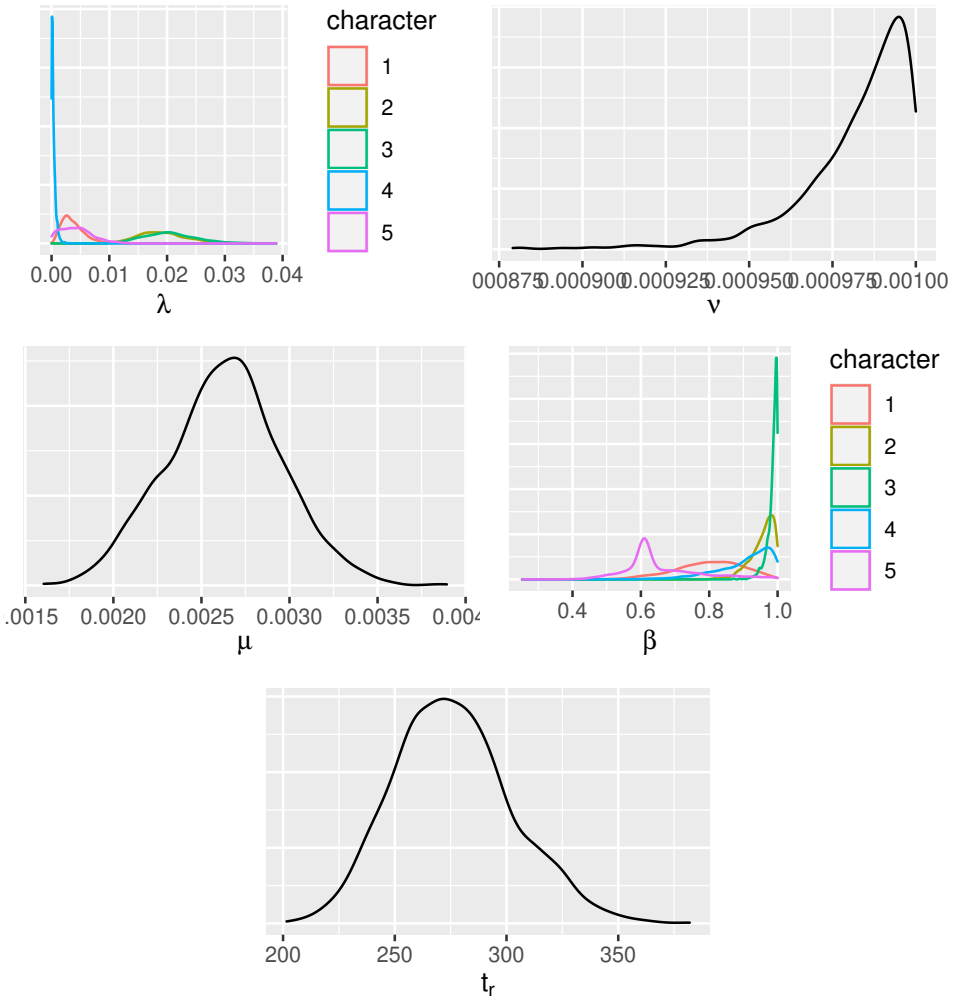


FIGURE 1.4 – Distributions *a posteriori* λ, μ, ν, β et l'âge de la racine.

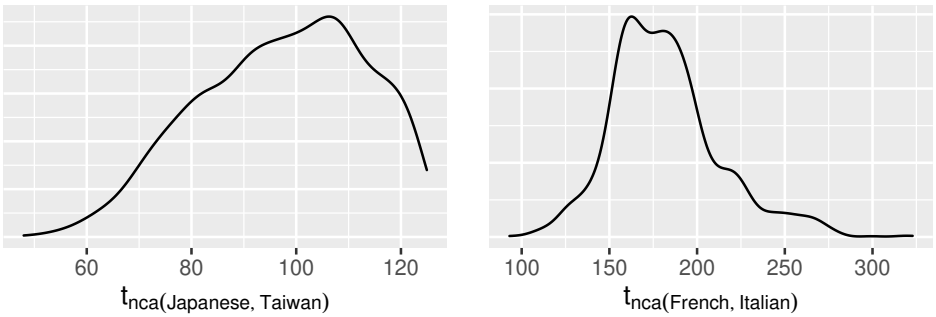


FIGURE 1.5 – Distributions *a posteriori* de l'âge du plus proche ancêtre commun de deux des langues a posteriori.

sur les observations.

Ce problème est patent depuis les premières heures des méthodes ABC. En général, la distance choisie est de la forme $d(x, x^\star) = d_s(s(x), s(x^\star))$, où s est une statistique résumée et d_s une distance, usuellement une simple distance euclidienne. Nous savons que pour peu que s soit une statistique suffisante, dans la limite des faibles distances l'inférence est exacte. Cependant, il est notable que l'existence de statistique résumée suffisante est rare. De plus, de par la malédiction de la dimension, atteindre de faibles distances peut se révéler impossible en pratique. Récemment, Fearnhead and Prangle [2012], Li and Fearnhead [2018] ont montré que le choix optimal pour s serait d'être de la même dimension que le paramètre. Choisir une telle statistique reste problématique, dans Raynal et al. [2019] les auteurs proposent de sélectionner parmi plusieurs statistiques proposées par l'utilisateur celles qui sont les plus utiles, malheureusement cette méthode est limitée aux cas monodimensionnels.

Un autre problème des méthodes ABC est leur inefficacité lorsque le paramètre vit dans un espace de trop grande dimension. Encore une fois, la malédiction de la dimension constitue un obstacle, et réduit drastiquement l'efficacité des méthodes, puisque tous les paramètres proposés ou presque seront dans des zones de faible posterior. La méthode ABC-MCMC a été conçue dans cet objectif, en introduisant des propositions locales, et donc plus informées que le prior. Cependant, en dimension supérieure à 30, les limites de cette méthode apparaissent. Une première expérience de méthode ABC à la Gibbs a été tentée dans Kousathanas et al. [2016], où s est choisie sous de très dures contraintes pour garder des garanties théoriques.

Input: nombre d'itérations N, point initial $\theta^{(0)} = (\theta_1^{(0)}, \dots, \theta_n^{(0)})$, seuils $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$, statistique $s_1, \dots, s_n$, distance sur les statistiques $d_1, \dots, d_n$.
Output: a sample $(\theta^{(1)}, \dots, \theta^{(N)})$.
for $i = 1, \dots, N$ **do**
  for $j = 1, \dots, n$ **do**
    $\theta_j^{(i)} \sim \pi_{\varepsilon_j}\{\cdot \mid s_j(x^\star, \theta_1^{(i)}, \dots, \theta_{j-1}^{i}, \theta_{j+1}^{(i-1)}, \dots, \theta_n^{(i-1)})\}$

Algorithm 1.1: ABC-Gibbs

Dans ce travail, nous proposons une méthode numérique inspirée de l'échantillonneur de Gibbs, lequel est notablement résistant à la malédiction de la dimension. Cette méthode, que l'on appelle ABC-Gibbs, repose sur la mise à jour séquentielle de chaque coordonnée du paramètre selon une méthode ABC classique, mais cette fois en faible dimension. L'algorithme est présenté en Algorithme 1.1. Dans cet algorithme, chacune des coordonnées de θ le vecteur des paramètres est mis à jour en tirant une nouvelle valeur selon une approximation de la loi condition-

nelle de cette coordonnée par rapport aux autres et aux observations : $\pi_{\varepsilon_j}\{\cdot \mid s_j(x^\star, \theta_1^{(i)}, \ldots, \theta_{j-1}^i, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)})\}$, si le choix de cette approximation est a priori libre, nous avons étudié le cas où cette approximation est issue d'un algorithme ABC standard, dont on rappelle la forme en Algorithme 1.2, qui a l'intérêt d'être particulièrement simple.

Input: Observations $x^\star$, nombre d'itérations N, seuil $\varepsilon > 0$, statistique résumée s.
Output: un échantillon $(\theta^{(1)}, \ldots, \theta^{(N)})$.
for $i = 1, \ldots, N$ **do**
  répéter
  $\theta^{(i)} \sim \pi(\cdot)$
  $x^{(i)} \sim f(\cdot \mid \theta^{(i)})$
  jusqu'à ce que $d\{s(x^{(i)}), s(x^\star)\} < \varepsilon$

Algorithm 1.2: Approximate Bayesian computation standard

Pour prouver la convergence de la méthode, on ne peut pas se contenter d'exhiber une loi stationnaire à la chaine de Markov ainsi définie, car il n'existe pas de forme analytique pour cet objet, à notre connaissance. Nous devons donc montrer la convergence de la chaine de Markov d'une manière non constructive. Pour cela, nous avons montré que l'opérateur itératif associé à une étape de l'algorithme était contractant au sens de la variation totale. Il possède donc un unique point fixe et la chaine de Markov converge en variation totale vers cette loi.

Pour montrer que cete loi limite présente un intérêt, il faut montrer d'autres propriétés sur celle-ci. Par des arguments de couplage, nous avons été capables de calculer la limite de la loi limite lorsque le temps attribué aux simulations des conditionnelles augmente. Sous cette limite et dans avec un certain choix de statistique résumée, il est possible de retrouver une loi connue.

En pratique, notre méthode montre une grande efficacité dans les dimensions élevées, en particulier sur des modèles hiérarchiques, ce à quoi l'on s'attendait de par le lien avec l'algorithme de Gibbs. Parmi les autres intérêts on notera la plus grande facilité dans le choix des statistiques résumées. Il reste néanmoins un certain nombre de questions sur la pratique de la méthode ainsi que sur son comportement exact. Même s'il ne s'est jamais produit de dégénérescence de la loi limite, cette possibilité est laissée ouverte par les résultats théoriques, et il serait intéressant d'étudier cette éventualité.

5.1 Convergence de la méthode

On synthétise ici la preuve de la convergence de l'algorithme que l'on propose. Nous avons montrer le résultat suivant:

Theorem 1. *Supposons que pour tout* $\ell \leq n$

$$\kappa_\ell = \sup_{\theta_{>\ell}, \tilde{\theta}_{>\ell}} \sup_{\theta_{<\ell}} \|\pi_{\varepsilon_\ell}\{\cdot \mid s_\ell(x^\star, \theta_{<\ell}, \theta_{>\ell})\} - \pi_{\varepsilon_\ell}\{\cdot \mid s_\ell(x^\star, \theta_{<\ell}, \tilde{\theta}_{>\ell})\}\|_{TV} < 1/2$$

avec $\theta_{>\ell} = (\theta_{\ell+1}, \theta_{\ell+2}, \ldots, \theta_n)$, *and* $\theta_{<\ell} = (\theta_1, \theta_2, \ldots, \theta_{\ell-1})$. *Alors, la chaîne de Markov définie par l'Algorithme 4.2 converge géométriquement en variation totale vers* ν_ε, *avec un taux* $1 - \prod_\ell 2\kappa_\ell$.

Pour démontrer ce résultat, on va en réalité montrer que l'opérateur $\mathcal{T}$ associé à une itération de l'algorithme est contractant dans l'espace des mesures pour la variation totale. Pour cela il suffit de montrer comment un couplage optimal de deux distributions μ et ν va être transformé en un couplage entre $\mathcal{T}\mu$ et $\mathcal{T}\nu$. Nous montrons ce résultat en utilisant les hypothèse sur chacune des conditionnelle approchée.

Notons que les hypothèses sont vérifiées dans l'hypothèse plus restrictive, mais néanmoins fréquente, que les paramètres sont tous à support compact et que toutes les densités ne s'annulent pas.

6 Méthodes de Monte-Carlo non linéaires et approximation particulaire

Dans le contexte des méthodes MCMC, les méthodes Markoviennes sont dites "linéaires". Dans ces méthodes itératives, la loi du point suivant ne dépend que de la position du point actuel, comme on le constate dans la forme de l'algorithme de Métropolis-Hastings. Dans la dernière décennie, de nouvelles méthodes ont essayé d'échapper à cette contrainte. Les méthodes dites *non linéaires* [Andrieu et al., 2011, 2007, Haario et al., 2001, Atchadé and Rosenthal, 2005] sont intéressantes pour leurs propriétés de convergence. Cependant, simuler ces dynamiques s'avère rapidement être complexe.

D'un autre côté, dans la communauté des EDP, il est connu que les processus markovien non linéaires apparaissent en tant que limite de systèmes de particules en interaction, lorsque le nombre de particules augmente [Sznitman, 1991, Méléard, 1996]. Le processus résultat, la loi du point suivant dépende la *la loi* du point actuel, et non seulement de sa position. Cette limite de champ moyen a d'abord été étudiée en physique statistique [Kac, 1956] et ensuite formalisée dans les travaux pionniers de McKean [1966, 1967], Kac [1956], Dobrushin [1979]. En biologie, ces modèles ont été développés pour décrire le comportement des nuées d'oiseaux ou des fourmis par exemple. La question est alors en général d'identifier la nature de la loi limite desdites dynamiques à partir du comportement microscopique des particules.

Input: Une population initiale de particules $(X_0^1, \dots, X_0^N) \in E^N$,
un temps maximum $T \in \mathbb{N}$, une loi de proposition Θ
une fonction d'acceptation h

Output: Un échantillon $\left(X_t^i\right)_{1 \leq i \leq N;\ 1 \leq t \leq T}$

for $t = 0$ **to** $T - 1$ **do**
  for $i = 1$ **to** N **do**
    (Proposition) Tirer $Y_t^i \sim \Theta_{\hat{\mu}_t^N}(\cdot\,|X_t^i)$ une proposition pour un nouvel état de la particule i;
    (Acceptation) Calculer $\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i) = \frac{\Theta_{\hat{\mu}_t^N}(X_t^i|Y_t^i)}{\Theta_{\hat{\mu}_t^N}(Y_t^i|X_t^i)} \cdot \frac{\pi(Y_t^i)}{\pi(X_t^i)}$;
    Tirer $U_t^i \sim \mathcal{U}([0,1])$;
    if $U_t^i \leq h\big(\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i)\big)$ **then**
      Définir $X_{t+1}^i = Y_t^i$; `// Acceptation, avec probabilité` $h(\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i))$.
    else
      Définit $X_{t+1}^i = X_t^i$; `// Rejet, probable si` $\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i) \simeq 0$.

Algorithm 1.3: Monte Carlo Collectif (CMC)

Dans ce travail, nous avons été capables de distinguer une classe d'algorithmes markoviens non linéaires, que l'on peut approcher en simulant l'évolution d'un système de (nombreuses) particules en interaction. Nous appelons ces méthodes Monte-Carlo Collectives (CMC), dont la forme est présentée en Algorithme 1.3. Il est alors possible d'interpréter ces méthodes de diverses manières. En plus de l'aspect non linéaire, on peut interpréter cette méthode comme un algorithme de Metropolis-Hastings où chaque particule se voit proposer une nouvelle position en fonction de l'entière population, et accepte cette position indépendamment des autres particules. Cette méthode est, sous cette deuxième acception, entre l'implémentation indépendante de nombreux Metropolis-Hastings, et un algorithme de Metropolis-Hastings où la cible est directement $\pi^{\otimes N}$, acceptant ou rejetant l'état de chaque particule ensemble.

Cette méthode reste markovienne : elle ne repose pas sur l'échantillonage d'importance, qui est à la base de nombreuses méthodes particulaires. Ces méthodes, à l'exemple de SMC [Del Moral et al., 2006], sont particulièrement adaptées à l'étude de cibles complexes. Cependant elles souffrent de la malédiction de la dimension, et le choix des paramètres est critique pour augmenter la qualité de l'approximation. La méthode que l'on propose souffre moins de ces difficultés.

Pour autant, CMC souffre de son coût numérique, qui n'est pas linéaire en

le nombre de particules, mais quadratique. Pour rendre malgré tout la méthode attractive, nous proposons une implémentation reposant sur les GPU. En particulier, la récente librairie KeOps [Feydy et al., 2020] a été développée justement dans l'objectif de simuler efficacement des systèmes de particules en interaction pour jusqu'à plusieurs milliards de particules, rendant par là même notre méthode compétitive.

6.1 Convergence de la méthode

La convergence de l'algorithme se démontre en deux temps. Tout d'abord, on montre que lorsque le nombre de particules augmente, la population se comporte comme une population de particules indépendantes de même loi. On appelle ce résultat limite de champ moyen, et on peut en déduire en corollaire la propagation du chaos, en suivant la dénomination de Sznitman [1991] :

Theorem 2 (Limite de champ moyen)**.** *Soit Θ une proposition qui satisfait les hypothèses 2, 3 et 4. Soit $(X_0^i)_{i\in\{1,\dots,N\}}$, N variables aléatoires indépendantes et identiquement distribuées de loi $\mu_0 \in \mathcal{P}(E)$ (*hypothèse de chaoticité*). Soit $t \in \mathbb{N}$ et soit $(X_t^i)_{i\in\{1,\dots,N\}}$, N particules construites à la t-ème itération de l'Algorithme 1.3. Alors la mesure empirique, $\hat{\mu}_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{X_t^i}$ satisfait :*

$$\hat{\mu}_t^N \underset{N\to+\infty}{\longrightarrow} \mu_t$$

où $\mu_t = \mathcal{T}^{(t)}(\mu_0)$ est le t-ème itéré de l'opérateur de transition (7) *partant de μ_0 et où la convergence est la convergence en loi (*i.e. *la convergence faible-$\star$ dans l'espace $\mathcal{P}(\mathcal{P}(E))$).*

La preuve de ce résultat est basée sur des arguments de couplage, inspirés par Sznitman [1991] et adaptés de Diez [2020]. La preuve complète peut être trouvée en Section 31.4. La propagation du chaos s'énonce alors ainsi :

Corollary 3 (Propagation of Chaos)**.** *Soit Θ une loi de proposition satisfaisant certaines hypothèses. Soit $t \in \mathbb{N}$, $\ell \in \mathbb{N}$ et soit $\mu_t^{\ell,N} \in \mathcal{P}(E^\ell)$ la loi jointe au temps t de n'importe quel sous groupe de ℓ particules construites par l'algorithme 5.1, initialement i.i.d de loi commune $\mu_0 \in \mathcal{P}(E)$. Pour chaque ℓ-uplet de fonctions continues bornées $\varphi_1, \dots \varphi_\ell$ sur E, on a :*

$$\int_{E^\ell} \varphi_1(x_1)\dots\varphi_\ell(x_\ell)\mu_t^{\ell,N}(\mathrm{d}x_1,\dots,\mathrm{d}x_\ell) \underset{N\to+\infty}{\longrightarrow} \prod_{k=1}^{\ell}\langle\varphi_k,\mu_t\rangle,$$

où $\mu_t = \mathcal{T}^{(t)}(\mu_0)$ est le t-ème itéré de l'opérateur de transition (7) *partant de la loi μ_0.*

Pour étudier la convergence lorsque $t \to +\infty$ du processus non-linéaire, nous commençons par une astuce artificielle pour simplifier les calculs. Nous transformons la dynamique en une dynamique à temps continu, en associant la chaîne de Markov à temps discret à un processus de Poisson de façon à ce que les temps de sauts de la chaîne continue coïncident avec ceux du processus de Poisson [Brémaud, 1991, Chapter 8, Definition 2.2]. Dans ce contexte continu, partant d'une distribution absolument continue $f_0 \in \mathcal{P}^{\mathrm{ac}}(E)$, la loi f_t au temps $t \in [0, +\infty)$ satisfait au sens faible l'équation intégro-différentielle suivante :

$$\partial_t f_t(x) = \mathcal{T}(f_t)(x) - f_t(x) = \int_E \pi(x)\Theta_{f_t}(y|x)h(\alpha_{f_t}(x,y))\left(\frac{f_t(y)}{\pi(y)} - \frac{f_t(x)}{\pi(x)}\right)\mathrm{d}y \quad (1)$$

Notons que f_t est aussi absolument continue par rapport à la mesure de Lebesgue. Le résultat principal est alors 4 qui affirme que la solution de l'équation intégro-différentielle (1) converge à vitesse exponentielle vers π lorsque $t \to +\infty$ avec un taux qui dépend de la condition initiale. ce qui est le cas sous nos hypothèses, lesquelle nous permettent de contrôler l'équilibre entre la loi de proposition et la loi cible π. Partant de la distribution f, la fonction c^- mesure combien loin de π est la loi de proposition de f . Le meilleur cas étant atteint pour $c^- \equiv 1$ comme dans ce cas on a $\Theta_f = \pi$.

Cette hypothèse est vérifiée par la plupart des noyaux que l'on propose.

Theorem 4 (Convergence du processus non-linéaire)**.** *Soit Θ une loi de proposition satisfaisant certaines hypothèses. Soit $f_0 \in \mathcal{P}_0^{\mathrm{ac}}(E)$ et soit f_t la solution au temps $t \in [0, +\infty)$ de l'équation intégro-différentielle* (16)*. Alors pour tout $t \geq 0$, on a:*

$$|f_t - \pi|_{\mathrm{TV}} \leq C_0 \mathrm{e}^{-\lambda t},$$

où $C_0 > 0$ dépend uniquement de f_0 et π et où

$$\lambda := c^- \left(\inf_x \frac{f_0(x)}{\pi(x)}\right) h(1) > 0.$$

La preuve de ce résultat est basée sur des méthodes d'entropie, où l'on montre qu'une quantité décroît strictement avec le temps. Notons que cette quantité peut être choisie avec une certaine liberté, par exemple on pourra s'intéresser à la divergence de Kullback-Leibler.

Part 2
Introduction

This work presents three contributions to applied and computational Bayesian statistics. In the first one, we infer from a model of joint lexical and phonological changes a reconstruction of language history. In the second one, we propose an Approximate Bayesian method with Gibbs-like steps (ABC-Gibbs), an approximate method for high dimensional models. We prove its convergence and show its efficiency on several examples. In the third one, we develop Collective Monte Carlo (CMC), a method relying on interacting particles to mimic a non-linear MCMC method. We prove its convergence and efficiency on several examples.

7 Language families

Since the 18th century, comparative linguistics came to observe that some languages in the world share high levels of similarities, in the vocabulary, the grammar or the phonology. These similarities appear clearly when comparing the basic vocabulary in some languages, see Table 2.1 for an example on Romance languages. By comparing European and Indo-Iranian languages, linguists conceived the idea of a unified family of languages (the Indo-European family) with a common ancestor (Proto-Indo-European). The great interest of 19th scientists for Indo-European studies was not exempt from nationalists *arrière-pensées*, in particular in German countries that found in Historical Linguistics a way to relate German people to antique civilisation.

Today, many other language families have been defined: Indo-European (IE), Semitic, Sino-Tibetan, Finno-Ugric, *etc.* For some of these families, several subfamilies have been distinguished, some which are attested through written sources (*e.g.* Sinitic or Romance languages) some through comparison of currently existing languages (*e.g.* K'iche' languages). These distinctions are based on pure comparison of languages, without quantitative studies, even though some models try to explain how change appears and spreads in languages (*e.g.* wave model [Ross et al., 1988], functional load [Martinet, 1970]). Even today, while the history of Indo-European languages is well known and mostly accepted, some language families have not been studied in depth. Among those, sign languages, whose evolutionary process, families and history remains *terra incognita*.

The goal of historical linguistics is to reconstruct the history from ancestral language to the current languages, by showing relationship between languages. This history is usually represented as a dated tree, whose construction is a complex matter. It often involves quantifying phonological changes, and usually needs to

English	Latin	Portuguese	Spanish	French	Italian
four	quattuor	quatro	cuatro	quatre	quattro
five	quinque	cinco	cinco	cinq	cinque
big	grandis	grande	grande	grand	grande
long	longus	longo	largo	long	lungo
wide	amplus, largus	largo	ancho	large	largo
thick	grossus	grosso	grueso	gros	grosso

TABLE 2.1 – Extract of Swadesh list for Latin and some Romance languages

introduce archaeological data to the study. For example when in a family all the languages use a similar word to designate a domesticated animal, we can date the apparition of the word with respect to the known history of domestication [Diamond and Bellwood, 2003]. To overcome these difficulties, some researchers have tried to systematise this work by introducing precise generative models.

The first *quantitative* work to date the common ancestor of two languages was known as *Glottochronology* introduced by Swadesh [1952]; this method relies on the assumption that vocabulary of languages evolve at a fixed known rate, so that by counting the number of words with common ancestry between the two vocabularies — originally Swadesh lists, which are constituted of a few hundred meanings that were considered the most stable, and expected to appear in every language — we can estimate the date of the common ancestor to these two languages, that is the time were the ancestral population splits into two population, whose language, initially the same, evolve independently into what will become the two languages studied. This method suffers many flaws, the most obvious one being the lack of information as for the evolution rate, the second being the fact that it is not reasonably efficient to build a phylogeny from the study of pairs of languages, especially because of the large variance of the estimators obtained from the method. Despite this, the use of similar methods, such as Neighbour Joining, remains of common use in historical linguistics.

More recently, phylogenetic models, originally used in biology, have been applied to linguistics.

8 Phylogenetic linguistics

Phylogenetics is the study of evolutionary relationships between objects, characterised by features, that all descend from a common ancestor. This approach at first used in genetics and biology is applied for example to study DNA evolution, allowing the inference of phylogenies for species. It can also be used with phenotype traits displayed by individuals. Due to the nature of the dataset, often

long DNA sequences that need to be aligned and then compared, it made sense to use computational methods, especially as improvements in statistics allowed for more complex models to be numerically studied [Warnow, 2017, Yang, 1994]. The parallel between genetics and linguistics is somehow debatable, as languages present many peculiarities, such as borrowing, bilingualism, and human intervention. However, the tree structure, at the heart of phylogenetics seems to be fairly accepted in the community for large time scales.

The use of computational methods for linguistics is a rather recent addition. In the early paper of Ringe et al. [2002], the authors compute from a dataset of characters (phonological, grammatical or syntactical) for several Indo-European languages a phylogeny that intends to be as close as possible to a perfect phylogeny, that is a phylogeny under which the observations are perfectly possible. This methods however is not a statistical treatment of the problem in the sense that it does not relies on statistical modelling and uncertainty cannot be assessed. This early model mixed several traits that are not comparable, and did not allow to lead statistical inference. The choice made since by computational linguists has been to focus on lexical datasets.

The lexical data is usually constituted of cognacy classes. A cognacy class is a set of words with same meaning and a common ancestor. In Gray and Atkinson [2003], the author propose to reconstruct trees from lexical datasets with a "rescriction site" model, that is a model where cognacy classes appear and disappear with different unknown rates. In the seminal paper [Nicholls and Gray, 2007] the authors, inspired by Dollo [1887], have modelled the evolution of languages through the evolution of cognacy classes. The evolution of the lexicon is then modelled by the apparition or disappearance of cognacy classes according to a point process. This model allows for relatively efficient computation of the likelihood, through pruning algorithm [Felsenstein, 1981], and the authors prove its efficiency on the well studied Indo-European languages. This method, and similar ones, have been improved and successfully applied to many examples: Dravidian languages [Kolipakam et al., 2018], Sino-Tibetan languages [Sagart et al., 2019], Pama-Nyungan [Bowern and Atkinson, 2012], or Austronesian languages [Gray et al., 2009]. Among the improvements brought, we can cite, firstly, Ryder and Nicholls [2009] in which the authors added the possibility to include missing data, by integrating on the possible value for this data. This allowed the authors to study dead languages, some of which are very little known, for example *Luwian* in the Indo-European family, while keeping the computational cost to a manageable level. A second improvement, [Ryder, 2010], has been the adjunction of some anisotropy to the model, by adding bursts of evolution along the tree. More recently, the matter of horizontal borrowing in the evolution has been studied in Kelly [2016], by adding the possibility that a word can be borrowed horizontally at

any time. This last improvement dramatically increases the computational time, although technical improvements have allowed to apply this model to Eastern-Polynesian languages [Kelly and Nicholls, 2017]. Other models, always based on cognacy datasets have been used to reconstruct not only phylogenies of languages but also geographical evolution of these languages. For example a recent work on Australian languages [Bouckaert et al., 2018] was focused on finding the location of the common ancestor to the language family studied.

In all these works, cognacy classes remain at the heart of the model, see for example the recent works on Dravidian languages [Kolipakam et al., 2018], Sino-Tibetan languages [Sagart et al., 2019], Pama-Nyungan [Bowern and Atkinson, 2012], or Austronesian languages [Gray et al., 2009]. The main caveat of this method lies in the collection of the dataset: cognacy lists are built by hand by linguists, in a time consuming and often debatable process, which is based upon other information, for example phonetic or grammatical. This information often comes itself from a more thorough linguistic model. Consequently, there are risks that the constitution of the dataset induces a bias in the inference, and increase the odds that the inferred phylogeny fits the phylogeny presupposed by the linguists who built the dataset. This method also implies that linguists have recorded full lexical form of the words and discard them to only keep their cognacy judgments. Automatic inference of cognacy classes is a rather new field of research [List et al., 2017]. Previously, Bouchard-Côté et al. [2013] supposed the phylogeny known, which is not feasible in general. Among the proposed models, List et al. [2016] relies on Levenshtein-like distance between the words written in phonological alphabet (IPA), the distance between phonemes is linked to the proximity of these phonemes with respect to some criteria; in Rama [2018b], the authors propose to add a Chinese restaurant process to find clusters of words that will be interpreted as cognacy classes, although the authors do not exactly propose an evolutionary model to explain the algorithmic solution adopted.

All these models rely on phonological similarity between the words; contrary to DNA sequences, phonology presents some particularities. The most striking particularity is the regularity of phonological transformations in most of the proposed models [Lavie, 2007], that is most of the transformations are absolutely regular, *e.g.* from Vulgar Latin to French, all the final vowels have disappeared [Joly, 1995], and most of the words in a language undergo this transformation at the same time; furthermore, these transformations are highly context dependent. Finally, and more importantly, the notion of phonological inventory is crucial to characterise which transformations are possible and how the different phonemes are organised together [de Boer, 1997, Dunbar and Dupoux, 2016]. To our knowledge there is for now no mathematical model interested in these questions, although some remarks on functional load in Bouchard-Côté et al. [2013] can be linked to

this question.

Among the works on the question of phonology, Bouchard-Côté et al. [2013] propose to model phonological changes with modifications that independently transform each word seen as a string of characters. On a known phylogeny, words evolve by undergoing on each branch a transformation written as a chain transducer, which allows for context dependency. The transition probabilities between phonemes can then be inferred. The most common phonological transformation — spirantization, lenition, *etc.* — are correctly reconstructed, and it is possible to reconstruct ancient forms from the ancestral languages. This method also allows the author to infer cognacy classes, to compute similarities between words using the previous model. The authors conclude that a more advanced version of their probabilistic model could be used to infer both phylogeny, cognacy classes and phonological transformations at the same time, but this problem was left as an open question.

The prospect of inferring cognacy classes automatically opens new possibilities, especially because they considerably reduce the human work needed to study language family, some of which are not thoroughly studied. In Rama [2018a], the authors compare phylogenies inferred from specialists built cognacy lists and automatically constructed lists, without notable change.

In this work, we will mainly focus on sign languages. Despite being present all over the globe, and spoken by millions of persons, sign languages remain understudied, especially compared to their spoken counterparts. While the European history of deaf educational institutions is well documented, the phylogenetic links between languages remain a complex issue, as well as the phonological, grammatical and lexical phenomena that drive sign languages evolution. In Power et al. [2020], the authors study, using a distance based Neighbour net, the proximity between some sign languages. The dataset used to this extend is constituted of the manual alphabets of each sign language, that can be seen as non iconic — however there is clear iconicity in the signs for *n, m, u, v, w* and *z*, which mimics with fingers the form of the letter. Each sign is characterised by some features describing the position and movement of the hands, these features constitute the basis of the current study of sign languages phonology.

9 Statistical and numerical methods for phylogenies

Phylogenetic inference is usually done in a Bayesian setting. The Bayesian paradigm [Robert, 2007], first introduced by Pierre-Simon de Laplace and Thomas Bayes in the 18th century, allows for an uncontested coherence in statistical inference. Un-

der this paradigm, a probability distribution on a quantity θ represent uncertainty on this quantity; as a corollary, we can update the uncertainty on a quantity by adding more observations according to Bayes rule, transforming a *a priori* distribution $\pi(\theta)$ into a posterior distribution that summarises both the prior knowledge and the knowledge added by the observations X, contained in the likelihood function $L(X \mid \theta)$:

$$\pi(\theta \mid X) \propto \pi(\theta) L(X \mid \theta).$$

In practice, this means that by computing the posterior distribution we can precisely measure uncertainty on a quantity. We generally wish to estimate integrals of the form $E_{\pi(\theta|X)}[f(\theta)]$. In some rare cases this can be done exactly, but in most cases the computational complexity of this operation prevented widespread use of these methods until the development of efficient numerical tools. One of the major breakthrough in this field is the Metropolis-Hastings method [Robert and Casella, 1999], first introduced in Metropolis et al. [1953], whose idea is to simulate a Markov chain whose stationary distribution is the target posterior distribution. We present in Algorithm 2.1 the general form of the method. To compensate its known flaws — bad mixing properties, choice of the parameters, the proposal q in particular — many improvements have been added in the recent years. Beyond the adaptive versions of Metropolis-Hastings Roberts and Rosenthal [2009], that intends to learn the best choice of parameters during the convergence of the chain, some other methods have been proposed. Tempering methods focus on changing the target so that it is easier to sample from. By flattening the distribution, potential barriers lowers and a Metropolis-Hastings algorithm can jump between the modes of the distribution. This is used for example in Parallel tempering [Geyer, 1991]. The basic idea of this method is to introduce a sequence of distributions from a simple and known distribution — the prior distribution — to the target for which the mixing properties of the Markov chain are bad. Between these two extremal chains we introduce as many chains as needed, so that the target distribution of each chain is close to the next and previous chain. For example, we can think of a likelihood tempered target, which is one of the most simple tempering, we introduce tempered distributions $\pi_t \propto \pi(\theta) L(X \mid \theta)^t,\ 0 \leq t \leq 1$, for $t = 0$ the tempered distribution is merely the prior and for $t = 1$ the posterior. In this case each distribution is associated with a Markov chain that runs independently from the others, except from some times at which we propose to exchange the states of two neighbouring chains. As the distribution are close, this has a large acceptance probability, and in the end if all the chains exchange sufficiently frequently, the top chain, that is the one with the right target distribution, will inherit the mixing properties of the lower chains. This algorithm has the major advantage of being simply parallelisable [Syed et al., 2019], by running each chain on a different core. This allows to leverage medium size CPU unit but does not clearly benefit from

modern GPU systems.

Importance sampling methods constitute an other major branch of Monte-Carlo methods, based on the following simple idea [Robert and Casella, 2013]

$$\int f(x)\mathrm{d}x = \int f(x)h(x) \cdot \frac{\mathrm{d}x}{h(x)}.$$

This identity can be rewritten in terms of random variables, when one wants to compute $E_p[f(X)]$, it is possible to use a random X from another distribution q, if a corrective term is added:

$$E_p[f(X)] = E_q[f(X)p(X)/q(X)],$$

the choice of q is the main problem of importance sampling. To simplify this choice, and in order to tackle more difficult problems, several methods have been developed. In particular, sequential importance sampling, relying on the same tempering idea as parallel tempering (*c.f. supra*), represented by Sequential Monte Carlo (SMC) methods [Del Moral et al., 2006] have proven their efficiency on state-space models, such as filtering problems, where the sequence of target is obvious and almost written in the model itself. We present in Algorithm 2.2 the general form of SMC algorithm. The forward kernel depends on the tempering scheme chosen; they are of notable interest in the context of state space models. Beside the less commonly used Particle Monte Carlo (PMC) methods, SMC can be beneficial for the study of classical distributions, where the tempering sequence is not clear; the user has then to wisely choose a tempering scheme. For example, SMC methods have been applied to phylogenetical inference [Bouchard-Côté, 2014, Wang et al., 2015, 2019]. In these articles, the authors propose to extend both the likelihood and the prior distribution to forest spaces, the sequence of tempered target comes then from the coalescence of the phylogeny. More particularly in Bouchard-Côté [2014] the authors focus on the validity of the method, and in Wang et al. [2019] the authors extend the proposal distribution to increase the mixing properties at the cost of a higher algorithmic complexity. Particle methods such as PMC or SMC relies on a high number of particles, usually several hundreds, as only the number of particles ensures the validity of the algorithm [Del Moral, 2004]. To our knowledge it has however not been used in concurrence with GPUs, as this method involve parallel complex operations (*e.g.* Metropolis-Hastings steps), and

not simple matrix multiplications.

Input: An initial point X_0,
a maximum time $T \in \mathbb{N}$, a proposal distribution q, a target π
for $t = 0$ **to** $T - 1$ **do**
  (Proposal) Draw $Y_t \sim q(\cdot \mid X_t)$ a proposal for the new state;
  (Acceptance) Compute $\alpha(X_t, Y_t) = \frac{q(X_t|Y_t)}{q(Y_t|X_t)} \cdot \frac{\pi(Y_t)}{\pi(X_t)}$;
  Draw $U_t \sim \mathcal{U}([0,1])$;
  if $U_t \leq \min\big(\alpha(X_t, Y_t), 1\big)$ **then**
    Set $X_{t+1} = Y_t$; `// Accept`
  else
    Set $X_{t+1} = X_t$; `// Reject`

Algorithm 2.1: Metropolis–Hastings

Input: A tempering scheme $q = \pi_0, \ldots, \pi_T = \pi$,
where π is the target and q a known (simple) distribution,
forward kernels $q^{t \mapsto t+1}$.
(Initialization): $X_1^1, \ldots, X_1^N \sim_{iid} q$,
associated weights $w_1^i \propto \pi_1(X_t^i)/q(X_t^i)$, for $i \in 1, \ldots, N$;
(Mutation) Update the particle position according to K_1 a kernel with stationary distribution π_1.
for $t = 2, \ldots, T$ **do**
  (Importance update) Sample $X_t^i \sim q^{t-1 \mapsto t}(\cdot \mid X_{t-1}^i)$ and compute the associated weight:
  $w_t^i \propto w_{t-1}^i \frac{\pi_t(X_t^i) q^{t \mapsto t-1}(X_t^i \to X_{t-1}^i)}{\pi_{t-1}(X_{t-1}^i) q^{t-1 \mapsto t}(X_{t-1}^i \to X_t^i)}$;
  (Resampling) if $\hat{ESS}(w_t^i) = (\sum w_i^{t\,2})^{-1} < N_{\min}$ **then**
    Sample $J_i \sim \mathcal{M}ultinomial(w_t^1, \ldots, w_t^N)$;
    Set $X_t^i = X_t^{J_i}$;
    Set $w_t^i = 1/N$
  (Mutation) Update the particle positions according to K_t a π_t stable kernel.

Algorithm 2.2: Sequential Monte Carlo

10 Likelihood free methods

Even with growing computational capacities, and with further work from mathematicians, many of the models developed in application fields, such as the one we present in this work, are too complex to be handled with methods that require the computation of likelihoods. The model we present in Part 3 relies on a high number of latent variables and a large memory storage to reduce computational time.

Handling such complex, correlated, almost singular densities, in high dimensional space, is a challenge that requires many numerical adjustments. More precisely, many of the evolutionary models linguists have propose are too complex to be studied in a reasonable amount of time — especially when including borrowings and variability of the evolution parameters.

On the other hand, likelihood free methods can be seen as possible solution. These methods only require to be able to simulate so called pseudo-observations from the generative model, removing in particular the need for most auxiliary variables. Among these methods, Approximate Bayesian Computation (ABC) first introduced in Biology and then thoroughly studied for example in Tavaré et al. [1997], Beaumont et al. [2002], Toni et al. [2008], Csilléry et al. [2010], Moores et al. [2015] and Sisson et al. [2018], is the simplest. We present in Algorithm 2.3 the general form of the method. The idea of ABC methods can be summarised in a few words: if a parameter, generated from the prior, can be used to produce pseudo-observations close to the true observations, this parameter should be a sample from the posterior. The main question at this point is to find a way to quantify the distance between observations and pseudo-observations.

This problem has been of main concern since the first days of ABC methods. Usually the distance is defined as $d(x, x^\star) = d_s(s(x), s(x^\star))$, where s is called a summary statistic and d_s is a distance, usually a simple Euclidean distance. Firstly, we know that if we compare pseudo-observations through a sufficient statistic, the inference is, in the limit of small distances, exact. However, there are in general no sufficient statistic available in the model studied, and no way to check a possible sufficiency as the likelihood is not available. Furthermore, because of the curse of dimensionality, reaching small distances in a high dimensional space is not feasible. More recently, Fearnhead and Prangle [2012], Li and Fearnhead [2018] have shown that the optimal size of the summary statistic is a statistic of same dimension as the parameter. Choosing such a statistic remains problematic: in Raynal et al. [2019] the authors present an automatic method to select from a set of statistic the best one for the inference of the parameter; however this method remains limited to a one dimension parameter.

Another major issue of ABC methods is their inefficiency when used to infer high dimensional parameters[1]. Again, the curse of dimensionality drastically reduces the efficiency of the algorithm as all the proposed parameters will provide uniformly bad pseudo-observations. ABC-MCMC methods have been proposed, along with SMC-ABC methods, to tackle a part of the issue by proposing parameters from an auxiliary, adapted, density rather than from the prior. Nevertheless, in our examples these methods remain unable to deal with around 30 dimensions. A first attempt to "Gibbs'ed" ABC can be found in Kousathanas et al. [2016], where

[1] this problem is not specific to ABC but the problem is particularly stringent in this case.

the statistics are chosen under strong constraints to preserve the interpretability of the limiting distribution.

In this work, we propose a numerical method inspired by the Gibbs sampler, that is far more resilient to high dimension, as its complexity is linear in the dimension. This method called Componentwise-ABC, or ABC-Gibbs, is inspired by the Gibbs sampler in which the parameters are updated one at a time to reduce the N-dimensional problem to N 1-dimensional problems. We show its efficiency in some high dimensional problems, study its convergence, and try to interpret the approximation of the target which is different in general from the traditional ABC approximation. We do not tackle the application of these methods to phylogenies.

Input: observed dataset $x^\star$, number of iterations N, threshold $\varepsilon > 0$, summary statistic s, a distance d.
Output: an i.i.d. sample $(\theta^{(1)}, \dots, \theta^{(N)})$ from $\pi_\varepsilon(\theta) \propto \int \pi(\theta) f(x \mid \theta) \mathbf{1}_{d(s(x),s(x^*)} \mathrm{d}x$.
for $i = 1, \dots, N$ **do**
  repeat
  $\theta^{(i)} \sim \pi(\cdot)$
  $x^{(i)} \sim f(\cdot \mid \theta^{(i)})$
  until $d\{s(x^{(i)}), s(x^\star)\} < \varepsilon$

Algorithm 2.3: Vanilla Approximate Bayesian computation

11 Non-linear methods through Particle implementation

In the MCMC framework, most of the Markovian methods can be said "linear". In iterative methods the law of the next point depends on the position of the current point, as seen in Algorithm 2.1, where X_{t+1} depends only on X_t. However, over the last decade, some methods have escaped from this framework. The so-called "non-linear" Markovian methods [Andrieu et al., 2011, 2007, Haario et al., 2001, Atchadé and Rosenthal, 2005] are appealing for their efficiency and convergence speed. However, simulating from these dynamics is challenging.

On the other hand, in the PDE community, it is well known that non-linear Markovian processes appear as the limit of system of interacting particles, when the number of particles increase [Sznitman, 1991, Méléard, 1996]. In the resulting process, the X_{t+1} depends on the *law* of X_t rather than its realisation. This so-called *mean field* limit, has been first studied in statistical physics [Kac, 1956] and then formalised in the pioneering work of McKean [1966, 1967], Kac [1956], Dobrushin [1979]. In Biology these models have been developed to study, for example,

the behaviour of groups of animals: birds flock, ants. In these situations, the question is usually to determine the limiting distribution of the non-linear Markovian process that intends to describe the microscopic behaviour of the particles.

In Part 5 we have been able to distinguish a general class of non-linear Markovian algorithm that can be approximated by simulating the evolution of an interacting particle system. We call these methods Collective Monte Carlo (CMC). This methods can be interpreted in several ways. Aside from the "non-linear" perspective, we can interpret this method as a *interacting* Metropolis-Hastings algorithm: at each iteration, we compute a distribution from the entire population of particles, which is then in turn used as a proposal distribution by each particle in an independent classical Metropolis-Hastings step. We can say that this algorithm is a middle way between a fully independent parallel implementation of MH and a Metropolis-Hastings algorithm where a population of particles targets $\pi^{\otimes N}$, accepting or rejecting a single proposal for the whole population of particles.

This method remains Markovian: it does not rely on importance sampling which is the basis of most of particle methods commonly used. These methods, following SMC [Del Moral et al., 2006], are particularly adapted to the study of complex densities. However it suffers from high dimensionality, as it can induce a degeneracy of importance weights, and the choice of the mutation kernel is critical in order to increase the variability in the particle population. The method we propose does not suffer drastically from these flaws.

The main flaw of CMC is its computational cost, which is not anymore linear in the number of particles as are most of the other algorithms. However in the recent days, the use of GPU, a technology increasingly cheaper, has allowed for many complex computations to be made in a very reasonable amount of time. In particular, the recent library KeOps [Feydy et al., 2020], has been developed to handle interacting systems of many particles. In the context of CMC, it can deal with thousands of iterations for billions of particles in a few minutes, removing the previous fears about the cost of the algorithm.

Part 3

A Model of Lexical and Phonological Changes, with an Application to the History of Sign Languages

We propose to model phonological and lexical evolution of languages jointly, starting from a raw, well aligned, lexical dataset in phonological form. The model developed here could be applied to any dataset of aligned observations, for example languages that are close enough, or that share a common strict syllabic structure (*e.g.* Tai languages, Arab varieties), or even phenotypical observations in biology or anthropology.

This alignment requirement is critical to the model, as the model do not deal with characters chains, but matrices. We insist on the difference between studying *aligned* characters chains, as for DNA [Needleman and Wunsch, 1970], where some characters have no correspondence with others, and what we call perfectly aligned data, in which each item has the same number of characters. It is however possible to add a value for an empty character, for example if we study monosyllabic words with structure $(C)V(C)$, the $\emptyset$ value is a possibility for initial and final consonant.

In linguistics, linguists usually represent language evolution through two processes: a regular transformation process, corresponding to the application of *sound laws*, and the creation of new words — this last process can also correspond to a change of meaning. In previous models, *e.g.* Dollo model [Nicholls and Gray, 2007], only the second process is accounted for. Here, we are also interested in the first one. We will describe an evolutionary dynamic on the sign languages, represented as vocabularies of signs associated with meanings. In our model, each sign will evolve as a word in spoken languages [Hock, 1991]; that is, according to *regular* transformations that applies uniformly among the words of the vocabulary. This comes directly from historical linguistics [Collinge, 1985]. For spoken languages, words tend to evolve according to *sound laws*, that allows very few exceptions. For example, proto-Celtic (PC) labiovelar /k$^{\text{w}}$/ has disappeared from contemporaneous Celtic languages. It has undergone two different evolutions: in Irish it systematically became /k/ while in Britton it became /p/. This explains most of the /k/ – /p/ pairs that exists between Irish and Britton: *ceathair* – *pedwar* ($<$ PC: *k$^{\text{w}}$etwares), *ceann* – *penn* ($<$ PC: *k$^{\text{w}}$ennom), *etc.* Contrary to genes, words are correlated, because a rule applies to each word without distinction. Finding these laws usually require to compare languages that underwent this law with other related languages that have not. To our knowledge there does not exist a

mathematical study of these *sound law*, and the frequency of these transformations is not documented. We must add that usually, sound laws depend on the context (*e.g.* Bartsch's law [Zink, 1986] in French states that, in Latin, an accentuated /a/ in an open syllable, after a palatal consonant becomes /i̯e/ in French: *chien* $<$ *canem*).

As with spoken languages, sign languages are diverse. Their history reflects both demographic, political and cultural movements of the deaf community. Currently there is usually one standardised sign language per country. Many questions remain unanswered about sign languages, as these languages have been — and are still in some countries — discredited of looked down on. In particular, there is no consensus on their evolutions pattern. Do they evolve like spoken languages? is there a notion of phonetic rule, that is transformations without exception? can we find families of sign languages, or even reconstruct a phylogenetic tree? In this part, we will try to show how statistical methods can be of interest, especially about the last question.

Sign languages have been formalised quite late in their history, in France during the 18th century, thanks to the work of Charles Michel de l'Épée. The growing interest for deaf education led to the formalisation, through the creation of schools, of a so called Old French Sign Language (OldLSF). The diffusion of the French institutions in the world along with natural evolution created several different languages that are all related to the OldLSF: American English, Italian, Spanish, German among others; there even exist some written sources on the OldLSF, which is an exception in the scope of sign languages. These elements led the linguists to distinguish a family of sign languages whose common ancestor is the OldLSF. Some other families have been proposed, such as British Sign Language (BSL) and New-Zealand Sign Language (NZSL) because of the historical relationships and migrations between the countries.

Interactions between sign languages, and between sign languages and spoken languages, in the lexicon, but also some parts of the grammar is another question. For example, some of the signs used for LSF fingerspelling are clearly inspired by the Latin alphabet. Measuring the intensity of the influence of spoken language on sign language is a difficult question to answer, and it might be yet another bias that our model will have to deal with.

12 Dataset and characteristics of Sign Languages

We based our work on a vast sign language vocabulary for dozens of sign languages around the word, with 100 meanings chosen according to Woodward [2000]. Each word of the vocabulary corresponds to a sign that has been summarised in 25 characters [Abner et al., 2020], with a high variability in the number of possible

English	Handshsape	twohands	handpart
cat	5	True	rip
father	H	False	finger_front
boring	E	False	finger_side
New Zealand			
cat	L	True	radial_side
father	1_nb	False	radial_side
boring	E	False	finger_front

FIGURE 3.1 – Example of data from sign languages

values per character: between 2 and 48. These characters can be understood as phonemes, in first approximation, of sign languages[2]. Video realisations of these signs are available at `https://www.spreadthesign.com/` for many sign languages. These traits characterise signs by describing the position of the hand, the form of the hand, the moves, *etc.* Proximity between signs then appears clearly, for example when comparing the signs meaning *boring* in British SL and in New Zealand SL, both implying the same form of the hand. Notice that the dataset assumes that there is one word per meaning and per language exactly, that is there are no synonyms.

We transform this dataset into a matricial dataset, that is we observe a 3-dimensional matrix $D(j, m, k)$, with $j \in \{1, \dots, J\}$, $m \in \{1, \dots, M\}$, and $k \in \{1, \dots, K\}$, where j is a language, m is a meaning, and k is a character. Figure 3.1 presents a subset of the data.

This model could also be used with other types of aligned datasets, for example for oral languages. We give an example in the Tai languages in Figure 3.2. In this case, the characters are the parts of the unique syllable that constitutes the words.

13 Model and Parameters

To simulate the change in the vocabulary, we describe a dynamic on U the matrix representing the state of a language, initialised at the root with independent samples from an initial distribution π_0 and coinciding with the data $D(j, m, k) = U(j, m, k)$ for j a leaf. This matrix is modified by two stochastic processes on the tree g.

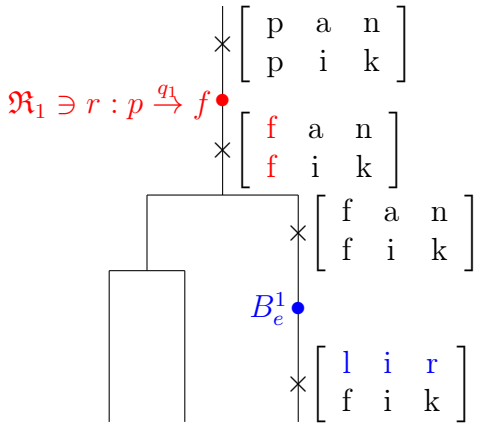
— A creation process, that accounts for apparition of new words, that is bor-

[2]This particular point is debatable, as some discussion with linguists seemed to indicate that characters are units of lower level, more similar in spoken languages to traits like *+height*/*-height* for vowels.

Siamese	Initial	Median	Coda	Tone
to obtain	d	aː	j	44
lunch	l	ɛː	ŋ	33
fog	m	ɔː	k	22
Tai-Dam				
to obtain	l	a	j	21
lunch	l	ɛ	ŋ	55
fog	m	ɔ	ʔ	45

FIGURE 3.2 – Example of data from Tai languages

FIGURE 3.3 – Stochastic evolution processes



rowings, meaning change, or creation.

— A mutation process, independent for each character, that accounts for the natural change in the phonology

These two processes are described by independent Poisson processes on the tree g. We represent in Figure 3.3 these processes and their effect on a small vocabulary.

The creation process consists in the apparition, for a given meaning, of a new word, that is an overall renewal of all traits for this meaning. This process has no true equivalent in the literature; however, it is similar to the Mk model [Lewis, 2001]. At a creation event, all the values of a row of the matrix U are modified. This happens with rate μ, independently for each meaning. A new word is created by drawing a new value for each character according to an initial distribution π_0.

The mutation process consists in the sequential application of evolution rules $r_i \in \mathfrak{R}_k$ specific to each character, where $\mathfrak{R}_k$ is a subset of the possible transformations between the values. At a mutation event, several elements of a column of

the matrix U are modified. This is a new adjunction to computational phylogenetical linguistics, as we are aiming at simulating the natural evolution of phonemes through time, as described by sound laws. For each character k, a Poisson process with rate λ_k, gives the position of the transformations, they are then chosen independently with parameters p_k from a set $\mathfrak{R}_k$ of possible transformations of the form $a_1 \to b_1, \ldots, a_N \to b_N$, with transformation $r_i : a_i \to b_i$ occurring with probability $p_k(i)$. When the vocabulary U goes through a transformation r_i, each occurrence of a_i is replaced by b_i with probability β_k independently across the meanings, and remains unchanged with probability $1 - \beta_k$. In the following, we will write r_i as well for the transition matrix associated with the transformation.

In the absence of a mutation process, the model can be compared with the model proposed in Pagel [1994], each meaning is independent of the other and we are only interested in the apparition of new cognates for a fixed number of meanings — although the dataset is clearly of different nature. In the absence of an apparition process, the model corresponds to a finite site model and each character is independent of the others.

The initial distribution is chosen as the product of independent uniform random variable on the possible values for each character. A more satisfactory choice would have been to take the limiting distribution of our dynamic associated with the other parameters. This would have been too computationally heavy.

In order to account for accidents and non modelled events, and also to simplify the computations, we introduce a noise: at the end of each edge, each character is changed uniformly to another with probability $(1 - \exp(-\nu\ell))$, with ν the noise coefficient and ℓ the length of the branch. The noise parameter is typically small.

To summarise, the whole process is described by:

— g the topology of the phylogenetic tree, with edges E;

— the associated set of edges length $(\ell_e)_{e \in E}$;

— μ the renewal rate of the meanings;

— λ_k the rate of the appearance of the transformations for each traits;

— p_k a vector of probabilities for each transformation of each character;

— β_k the probability that a transformation is applied;

— ν the intensity of the noise.

In order to account for uncertainty, we will perform the inference of the parameters in a Bayesian setting.

13.1 Prior distributions

13.1.1 On the evolution parameters

We take conjugate prior distributions on the evolution parameters. For λ, μ and ν we chose gamma priors fitted so that on each branch there are, in expectation, around five transformations or apparitions. There are two reasons for keeping the number of transformations low. The first one is the comparison with spoken languages, for which the number of transformations does not seem to be considerably higher. The second is that for a large number of transformations the state at the end of each branch is the equilibrium distribution of the process. We chose a beta $(1, 10)$ prior on β_k as the model bears little sense for small values of β. On the noise we choose a prior proportional to $1/\nu \mathbf{1}_{[10^{-6},10^{-2}]}(\nu)$, which is the prior used by Ryder [2010] and inspired by Jeffrey priors. It seems to be a more appropriate choice than exponential or Gamma priors as it will allow for bigger values of the noise, that mixes better.

13.1.2 On the tree

Following Nicholls and Gray [2007], we chose a prior on the trees of the form:

$$\pi(g) \propto \mathbf{1}_{t_r \in [a,b]} t_r^J,$$

where J is the number of leaves, t_r is the root age, and a and b are bounds on the root age. When the root age needs to be inferred, we further constrain this prior by adding constraints on clades and their ages:

$$\pi(g) \propto \mathbf{1}_{t_r \in [a,b]} t_r^J \prod_{c \in C} \mathbf{1}_{nca(c) \in [a_c, b_c]},$$

where C is the set of sets of leaves c for which bounds a_c and b_c have been defined on the age of $nca(c)$ their nearest common ancestor. Notice that learning the root age is only possible if enough internal constraints are provided. As appealing as learning the root age is, we advise against putting too much faith in the posterior; in general this age is little informed, resulting only in the propagation of a few constraints on the leaves. This was, anyway, not a primary concern of our study.

We also tried to add Gamma prior on the total length of the tree, that is

$$\pi(g) \propto \Gamma_{c,d}\left(length(g)\right) \mathbf{1}_{t_r \in [a,b]} t_r^J,$$

where $\Gamma_{c,d}$ is the density of a Gamma distribution with parameters c and d. The results were similar compared to the uniform prior, which indicates a small effect of the prior.

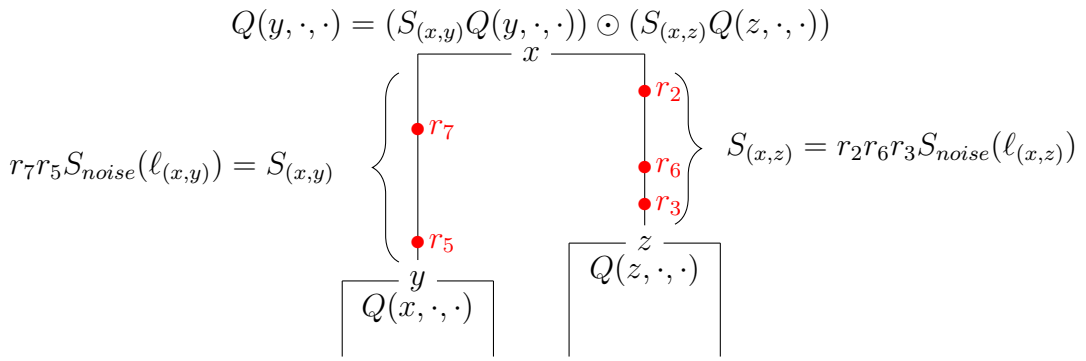


FIGURE 3.4 – Computation of the likelihood, without cognacy apparition, where $\odot$ designates a term-by-term product.

13.2 Computation of the likelihood

Computing the likelihood of the model is not straightforward. There is no closed form because of the intricacy of both evolution processes. We would have to integrate over all the possible transformations and apparitions on each branch, which is not possible. We thus add two latent variables for each character k and edge e in order to allow for computations:

— $R_e^k \in \mathfrak{R}_{\mathfrak{t}}^{\mathbf{N}}$ the list of the dated transformations for character k on edge e, with associated position on the branch;

— $B_e^m \in [0,1]^M$ the position of each apparition for each meaning m on edge e.

Given these auxiliary variables, the likelihood can be factorised on the traits, with $\theta = (g, \ell, \mu, \lambda, p, \beta, \nu, R, B)$ the set of all parameters and auxiliary variables, for $D^k = D(\cdot,\cdot,k)$:

$$\mathcal{L}(D \mid \theta) = \prod_k \mathcal{L}(D^k \mid \theta).$$

Then each term $\mathcal{L}(D^k \mid \theta)$ can be computed independently.

If there is no cognate apparition on the tree we can compute the likelihood with a pruning like method [Felsenstein, 1981]. The quantity on which we will prune is $\mathbb{P}(D(desc(j), m, k) \mid U(j,m,k) = u, \theta)$, where $desc(j)$ are the descendent leaves of j and $U(j,m,k)$ is the value for character k and meaning m at the node j anywhere on the tree. This value is never computed and is integrated over in the computation. For the sake of simplicity we drop the dependency in k and θ, and define $Q(j,m,u) = \mathbb{P}(D(desc(j), m, k) \mid U(j,m,k) = u, \theta)$. It verifies, for x a

node with successors y and z:

$$\begin{aligned} &Q(x,m,u) \\ &= \sum_{v,w} Q(y,m,v) P(U(m,y)=v \mid U(m,x)=u) \\ &\qquad\qquad Q(y,m,v) P(U(m,z)=w \mid U(m,x)=u). \end{aligned}$$

To compute the terms of the form $P(U(m,z)=w \mid U(m,x)=u)$, we have to account for both the transformations and the noise. We define S, the transition matrix of the value for a given character, by:

$$S_{(x,y)} = \left(\prod_{r \in R_{(x,y)}} r \right) \times S_{noise}(\ell_{(x,y)}),$$

where $S_{noise}(\ell_{(x,y)})$ is the matrix associated with the noise at the end of the edge (x,y) from x to y.

$$Q(x,m,u) = \sum_v \sum_w Q(y,m,v) Q(z,m,w) S_{(x,y)}(u,v) S_{(x,z)}(u,w)$$

with initialization $Q(x,m,u) = \mathbf{1}_{D(m,x)=u}$ if x is a leaf. Figure 3.4 summarises the relationship between the objects.

We have to store the value of Q at each node (that is, a matrix of size $n_k \times M$ with n_k number of possible values for character k and M number of cognates), for each character. This results in a high memory cost, however the gain in terms of computation time is quite important (compared to computing Q over all the tree each time it is needed), especially when only B or R are updated, as only the values of Q for the ancestral nodes of the modified node needs to be computed again.

If there is an apparition of a cognate on the edge, the formula changes, as the state of a particular cognate at this node has no influence on the state of the successor nodes: indeed, as there has been an apparition between x and y, the likelihood for this cognate is the same whatever the value at node x. In a way, we can say that the apparition *neutralises* terms in the likelihood, as it removes the information on all phenomenon lower on the tree.

Given an apparition time τ on the edge e and the time of the transformations $t(r), r \in R_e^k$, the law of the appeared cognate for character k at the end of the edge is:

$$\pi_{e,\tau} = \pi_0 \times \left(\prod_{r \in R_e^k,\ t(r)<\tau} r \right) \times S_{noise}(\ell_e),$$

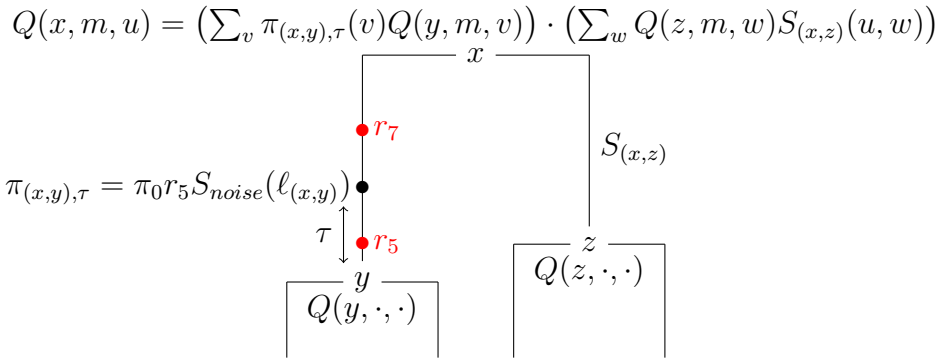


FIGURE 3.5 – Computation of the likelihood for meaning m, with an apparition on the edge (x, y).

given the apparition, the likelihood is the same for any value at node x, and $Q(x,m,u) = \sum_v \sum_w \left[\pi_{(x,y),\tau}(v)Q(y,m,v)\right] Q(z,m,w)S_{(x,z)}(u,w)$, that is the likelihood of the states for the meaning m at x for the subtree starting from y is now constant. We represent these computations in Figure 3.5

To summarise, the final likelihood for a character is:

$$\prod_{i=1}^{M} \sum_u \pi_0(u)Q(root,m,u).$$

13.3 Missing data

In our dataset, only 60% of the meanings have no missing data, and running the analysis only for these data points is a tremendous waste of the work from data collectors. In fact, there is an obvious and trivial way to deal with missing data, coming from the fact that in our model, each meaning is associated with a word and vice versa. A missing data appears clearly in the matrix D for a given character, a given meaning and given language.

The matrix Q at a leaf is initialised with the observations; it represents the probability, knowing the state at the leaf to observe the observations at this leaf, if we add the possibility for a data point to be missing, we have — as we know if the data is missing:

$$\begin{aligned} Q(j,m,k) &= P(D(j,m,k) \mid U(j,m,k), \text{missingness}) \\ &= \begin{cases} 1 & \text{if missing} \\ 0 & \text{if not missing and } D(j,m,k) \neq U(j,m,k) \\ 1 & \text{if not missing and } D(j,m,k) = U(j,m,k) \end{cases} \end{aligned}$$

Simply put, when the data is missing, any possible value would lead to that observation.

13.4 Inference of ancestral states

As stated in Section 13.2 the matrix Q contains the probability of the observations given the state of each character for each meaning. Given the latent variables R and L, it is possible to compute at any node x of the tree $\pi_x^k(m)$ the distribution of the values for meaning m, and character k at x, by descending the tree from the root, at which $\pi_r^k(m) = \pi_0$, to x. We can thus compute exactly, with n_k the number of possible values for character k:

$$\pi(U(x,m,k) \mid R, L, D) = \frac{Q(x,m,k) \odot \pi_x^k(m)}{\sum_{i=1}^{n_k} Q(x,m,k)(i)\pi_x^k(m)(i)},$$

the renormalization is computed easily as a sum over a dozen elements. Notice that in this case, when conditioning on R and L, the value for each meaning and each character is independent. If we are only interested in the distribution of the ancestral value for a given meaning and character, this is of little interest, but if we are interested in the *joint* distribution of several characters and meanings, care must be taken when renormalizing the conditional posteriors.

The posterior distribution is then:

$$\int_{R,L} \pi(U(x,m,k) \mid R, L, D)\pi(R, L \mid D)\mathrm{d}(R, L),$$

which can be approximated by Monte Carlo, as we already have a sample from $\pi(R, L \mid D)$.

The ancestral position can be chosen as any position of interest for the practitioner, designated as the nearest common ancestor of a set of languages — as we have to define this position on every trees of the posterior sample.

14 Numerical methods

As usually in numerical Bayesian statistics, we wish to obtain a sample from the posterior of trees g, parameters (λ, ν, μ, p, β), and latent variables (R and B). We encountered several difficulties linked with the nature of the posterior:

— The model has a high dimensionality, with numerous latent variables, and complex geometry;

— the latent variables have a definition that depends on the topology;

— the model includes both continuous and discrete random variables.

The first point is easily solved by the use of a Gibbs sampler. However using a Gibbs sampler constrains us on the update of the topology, because of a high correlation and the difficulty to write moves in the parameter space. Particle methods are then a possible solution to initialise several Metropolis Hastings in parallel in an area of high posterior probability, as they allow to use even bad mutation samplers as long as the number of particles increases. The expected gain is far superior to a mere increase in the running time of a single Markov chain. This is the solution we chose, focusing on SMC algorithm [Chopin and Papaspiliopoulos, 2020, Del Moral et al., 2006], presented in Algorithm 2.2. In this algorithm the particles X_i^j are updated though a bridge of distribution $q = \pi_0, \dots, \pi_{T+1} = \pi$ the posterior, with a forward kernel $q^{t \mapsto t+1}$. This sequence of distributions is called the *tempering scheme*. We call *mutation kernel* the kernel K_t.

All the code is available, along with the datasets used, at `https://github.com/GClarte/PhylogenyFromMatrices`. The file `GiveATry.R` contains all the information needed to launch the algorithm and instructions on the data formatting.

14.1 Choice of the tempering

The most simple choice of tempering [Chopin and Papaspiliopoulos, 2020] is to define $\pi_t(x) \propto \pi(x)^t$, as for $t = 0$ we have a known distribution and for $t = 1$ we have the target. However this is not possible in our case, as we cannot build a mutation kernel associated with this distribution. Nevertheless, we can find an almost equivalent tempering scheme. Indeed, the noise parameter can be understood as a temperature. For a high value of ν, on each branch the noise matrix S_{noise} is close to the matrix $(\pi_0, \dots, \pi_0)$, that is as if the model was a forest of single-leaf trees. Conversely, for ν small we retrieve the model we are looking for.

As the noise is a parameter that still needs to be learned, we propose the following target distributions. For $\nu_1 > \dots > \nu_T$ a sequence we define:

$$\pi_t(g, \mu, \lambda, p, \beta, q, \nu \mid D) = \pi(g, \mu, \lambda, p, \beta, q \mid \nu = \nu_t, D),$$

with a final target

$$\pi_{T+1}(g, \mu, \lambda, p, \beta, q, \nu \mid D) = \pi(g, \mu, \lambda, p, \beta, q, \nu \mid D).$$

That is we fix the noise parameter and we reduce it until it reaches a value that will lie in a part of the space with non negligible posterior probability. In chains $1, \dots T$, ν is fixed; it varies only in chain $T + 1$. Indeed, we start with high values of noise that have a null prior density, as we only need the final value of the noise

to have non-zero density. The forward kernels $q_{t\mapsto t+1}$ are just identity in this case, which greatly simplifies the algorithm.

In the numerical experiments, we chose the following sequence of noise tempering $\{10^{-1}, 5\cdot10^{-2}, 3\cdot10^{-2}, 2\cdot10^{-2}, 10^{-2}, 9\cdot10^{-3}, 5\cdot10^{-3}, 1\cdot10^{-3}, 5\cdot10^{-4}\}$, with some slight variations in some experiments.

This does not ensure a proper mixing of the SMC algorithm, as the number of tempering steps needed can be quite high — and a high number of steps would require a higher number of particles, which becomes quite heavy in terms of memory costs — leading to a genealogy degeneracy. This is a severe limitation to the use of our current version of the method, as this prevents us from estimating reliably the marginal likelihood of the model.

14.2 Mutation kernels

For the mutation steps, we run $4\cdot10^5$ iterations[3] of a Metropolis-Hastings within Gibbs sampler. That is we update one by one each coordinate of the parameter and latent variables conditionally on the value of the data and the other coordinates. More precisely we adopt the following update schemes; the probability to choose each of these update is chosen by the user so that the most important and most difficult updates are the most frequent. We update the latent variables R and B and the topology g more frequently than the other parameters, as they seem to be the most difficult to explore.

14.2.1 Update of R

We sample the proposed value of the transformations from the prior. Our experiments showed that this choice leads to better mixing than adding or removing one by one the transformations. This is due to the high correlation of the transformations, especially if all the pairwise transformations do not exist. The prior on the transformations is induced by ℓ, p and λ.

14.2.2 Update of B

We choose an edge and we change the whole vector of apparitions for each cognate, we propose a new value of B from the prior and accept it independently. This provides better results than the addition and removal of a single apparition in B. Although this problem can be seen as a random dimensionality, it corresponds merely to the result of a point process, which explains the simple treatment of the proposal. In particular, we know the distribution of the positions under the prior.

[3]For simulated datasets we reduced this number to $2\cdot10^5$, we also tried some real datasets examples with $8\cdot10^5$ although the results are not presented, as they were equivalent.

14.2.3 Update of the parameters λ, μ, p, β and ν

Some parameters (λ, μ, p and β) have been given a conjugate prior, which allows for an exact sample from the posterior conditional. For ν we rely on a Metropolis-Hastings step with normal proposal with standard deviation 10^{-4}.

14.2.4 Update of ℓ

Concerning the update of the length of the edges, we use a simple MH step, where we propose to move a node in the possible range of time constrained by the parent and children nodes: if x has w as parent and y, z as children, we propose a new age $t'_x \sim \mathcal{U}([\max(t_y, t_z), t_w])$. We propose also to rescale the tree, or part of the tree. For this last one, we sample a new age $t'_x \sim \mathcal{U}([0, t_w])$, and we rescale the ages of the subtree g_x starting from x by a factor t'_x/t_x.

14.2.5 Update of the topology

This last move is the most important and consequently the most difficult to choose. In the previous works [Gray and Atkinson, 2003, Ryder, 2010] a new tree is built by an SPR (sample-prune-regraft) move, in practice a subtree is always regraft at the nearest possible position — its cousin, meaning that the (sub)-tree $((a, b), c)$ becomes $(a, (b, c))$.

The question that remains is how to transport the latent variables of the changed parts of the tree. In order to preserve the intuition behind the latent variables, it seems reasonable to either keep all the ages of the nodes constant, or to change the ages while only moving to the nearest possible position — it is not possible to keep both as it would be incompatible most of the time. We describe in algorithm 3.1 the main method we used, a graphical representation can be found

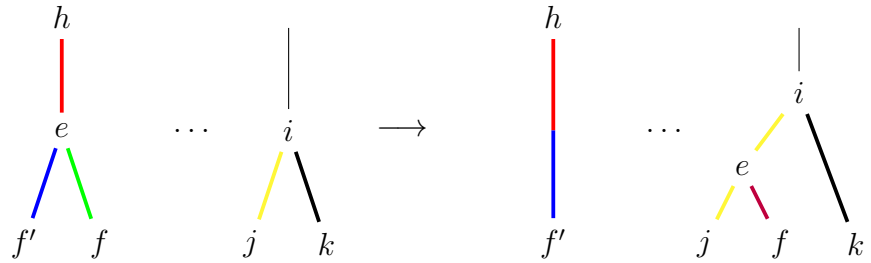


FIGURE 3.6 – Move of the subtree starting at f so that f has j as brother. The colors represent the edge-attached latent variables.

in Figure 3.6, the case of change with the nearest neighbour corresponds to $i = h$.

Let g be a tree, X the edges dependent latent variables.
Choose $f \in g$ a non root node, let e be its parent, h its grandparent and f' its brother.
Choose j a node, whose parent is denoted i, such that $haut(e) \in [haut(j), haut(i)]$ where $haut$ is the height of node.
Set j as the new brother of f, by splitting the edge $\langle i, j\rangle$ into $\langle i, e\rangle$ and $\langle e, f\rangle$, such that $haut(e)$ remains constant. Merge the edges $\langle h, e\rangle$ and $\langle e, f'\rangle$.
Merge the latent variables on $\langle h, e\rangle$ and $\langle e, f'\rangle$ into latent variables on $\langle h, f'\rangle$.
Split the latent variables on $\langle i, j\rangle$ into latent variables on $\langle i, e\rangle$ and $\langle e, j\rangle$.
Sample from the prior latent variables on $\langle e, f\rangle$.

Algorithm 3.1: Sample Prune and Regraft algorithm with latent variables. See Figure 3.6.

14.3 Numerical cost

These methods are surprisingly cheap in term of computation time. Our implementation of the MCMC algorithm only takes a few seconds for 1000 steps. Most of the cost is due to the computation of the likelihood, that requires the pruning of the tree. We implemented several improvements to reduce the number of computations, but some of these improvements have proven to dramatically increase the memory cost of the method — in a few words, we store more and more of the partial computations — at a point that the 268GB of the cluster we used were not enough. A more clever implementation or a better handling of memory, maybe allowed by the use of some lower level programming language, might be the key to overcome this problem.

On the real dataset, our implementation in `R` takes around 24 hours for 1500

particles, 4000 MH steps between tempering steps and 40000 final MH steps, on a cluster constituted of 30 CPU Intel Xeon CPU E5-2630 v4 2.20GHz.

15 Validation

Validation of statistical and numerical methods can be challenging: as long as we have no access to the true posterior we cannot know if the sample resulting from our algorithm is a good approximation. Furthermore, there is no reason why the posterior should be concentrated around the true value of the parameters, if the model is too complex to ensure the traditional assumptions for the concentration of the posterior [Ghosal et al., 2000]. Not to mention that, as the model we study is an approximation of the true phenomenon, the notion itself of *true parameter value* has little to no sense.

However, we can run some proofs of concept to show that the algorithm returns non absurd results. That is what we strive to do in this part. We can also check the behaviour of the model when the dataset comes from a slightly different model; all models are wrong but we can hope that ours provides seemingly meaningful result even if the true model were more complex.

15.1 On synthetic data

First, we can run numerical methods on a synthetic dataset, to verify that the returned sample is likely to be close to the parameters used to generate these synthetic observations. It can also allow to run some sanity checks on identifiability, numerical methods, and understand the main points of uncertainty in the posterior under this model.

For this experiment, we chose 4 characters with the following specifications:

Number of values	Possible transitions	λ	π_0	p	β
5	$1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5 \leftrightarrow 1 \leftrightarrow 4$	10^-2	$\mathcal{U}\{1, \dots, 5\}$	uniform	1
5	$1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5 \leftrightarrow 1 \leftrightarrow 4$	10^-2	$\mathcal{U}\{1, \dots, 5\}$	uniform	1
2	$1 \leftrightarrow 2$	10^-3	$\mathcal{U}\{1, 2\}$	uniform	1
2	$1 \leftrightarrow 2$	10^-3	$\mathcal{U}\{1, 2\}$	uniform	1

The true tree is represented in top of Figure 3.7. We constrain leaves $\{9, 11, 10\}$ and $\{8, 12, 13, 15\}$ to form clades and we added the following constraints on some internal node ages: the nearest common ancestor of 10 and 9 in $[10, 200]$, of 1 and 2 in $[10, 200]$, of 8 and 15 in $[20, 30]$. Different constrains on the internal nodes led to very similar results.

We generated 100 data points with these characters on a 15-leaved tree presented on the top in Figure 3.7. We ran the previously described SMC sampler. We used $2 \cdot 10^3$ particles, for the mutation step we ran 10^3 iterations of our Metropolis within Gibbs algorithm. After the last tempering, we ran a last mutation step of 10^4 iterations for each particle. In total, on a cluster constituted of 30 CPU Intel Xeon CPU E5-2630 v4 2.20GHz, it took roughly 5 hours.

The consensus tree is represented on the bottom in Figure 3.7. The overall topology seems quite well reconstructed, with variations on the small branches, which seems normal, as the small branches rarely host even a single transformation. Noticeably each group of leaves reconstructed by the consensus tree corresponds to a true clade. Node ages do not seem accurately reconstructed, in accordance with the root being too ancient.

As for the parameters, we provide in Figure 3.8 the posterior distributions for the main evolution parameters. They seem quite efficiently reconstructed — particularly the transformation rate β which is close to 1, the true value — with an underestimation overall the edge length parameters, which is linked to the overestimation in edge length — the number of events per branch is well reconstructed in the lower parts of the tree. This induces a poor performance in the reconstruction of cognacy when studying distantly related leaves, as the number of apparitions on the long edges starting from the root is underestimated. Maybe the number of character should be increased so that the creation of new words becomes "cheaper" compared to the evolution. In Figure 3.9 we represent the posterior cognacy probability. We have around 20% of false positive. However, in general, meanings with low cognacy probability do corresponds to words that are non cognates in truth. The root age is overestimated in accordance to what has been said before.

Concerning the reconstructed ancestral state, the result highly depends on the age of the common ancestor. Inferring the state of the common ancestor to 9 and 10 is fairly simple. More interestingly, we present in Figure 3.10 the posterior probability for the values of characters 1 and 4 at the common ancestor between 2 and 6, this node is correctly reconstructed only with 87%. If the majority of the values are correctly reconstructed, some are completely off. Noticeably, the first character, that evolves faster, is more uncertain than the fourth. We advise to use this reconstruction method with caution, as the efficiency of the method can greatly vary between meanings and characters.

The major risk of SMC is the degeneracy of the genealogies, which increases the variance of the resulting estimators. We can represent in Figure 3.11 the genealogies on a particular run to check that the common ancestor to the particles is not too "recent". The degeneracy is incontestable, which means that our algorithm does not mix enough. We would need more particles, or we could have smoothed the decreasing in ν but that may not have solved every problem, as this would still

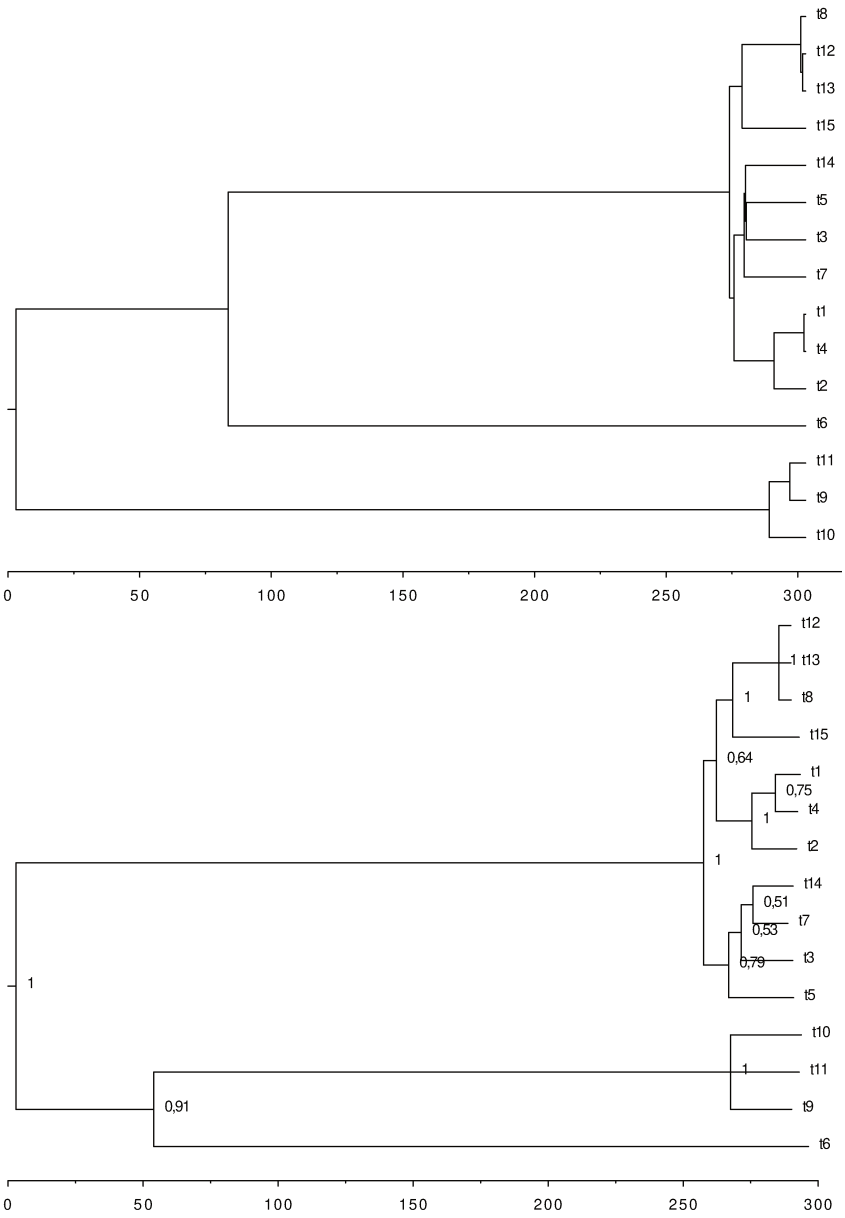


FIGURE 3.7 – True tree and consensus tree of the final sample, simulated dataset. Internal nodes of the true tree are tagged with their posterior probability.

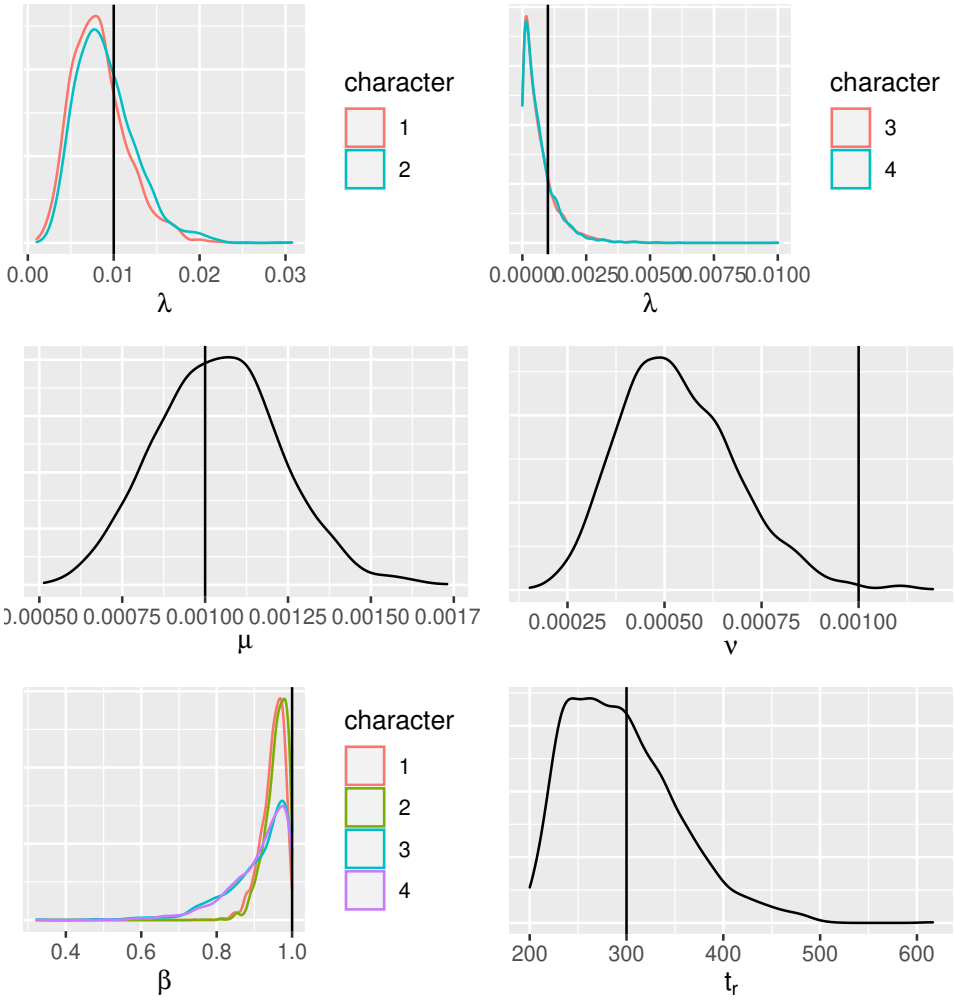


FIGURE 3.8 – Posterior estimations of λ, μ, ν, β ant the root age. The true values are indicated by the vertical lines.

	$p < 1/4$	$1/4 < p < 3/4$	$p > 3/4$
True	1	16	25
False	32	16	6

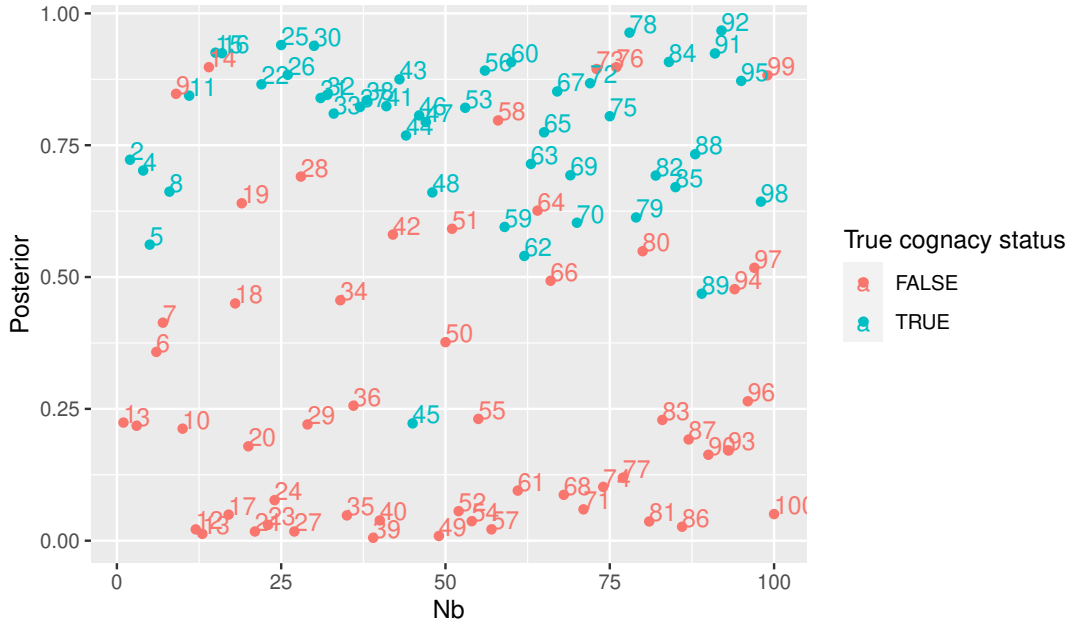


FIGURE 3.9 – Results on cognacy judgment based on the posterior cognacy probability (tabular) and posterior cognacy probability for each meaning (Nb) between $t4$ and $t6$. The color indicates the true cognacy status (graph).

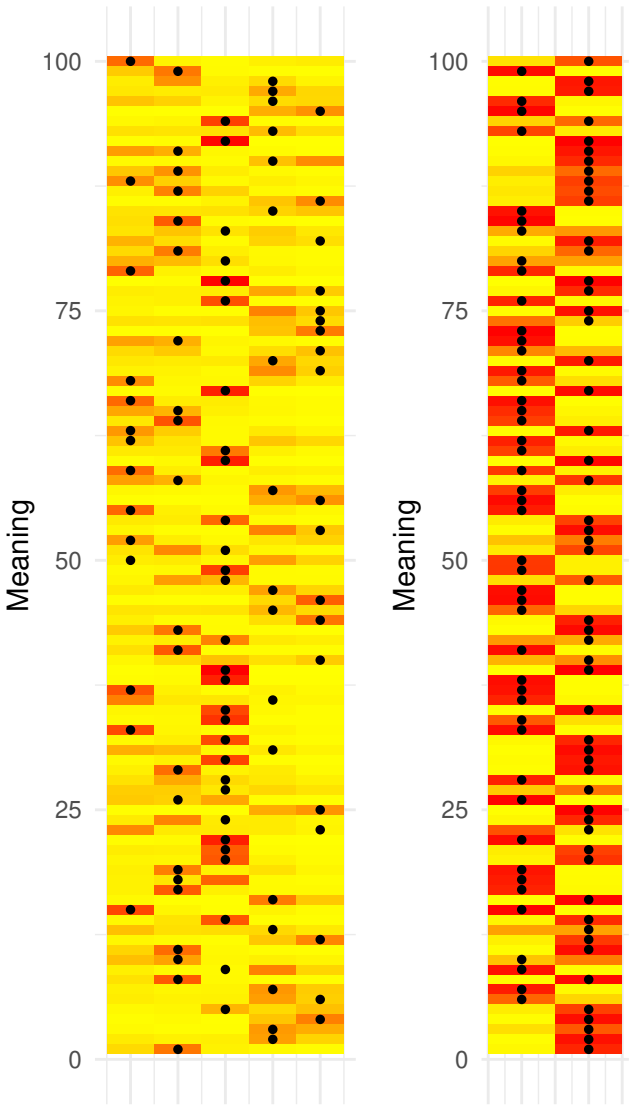


FIGURE 3.10 – Posterior distribution of internal ancestral values for the first and fourth character at the nearest common ancestor of 6 and 2. The color represents the posterior probability for each value, the true value corresponds to the black dot.

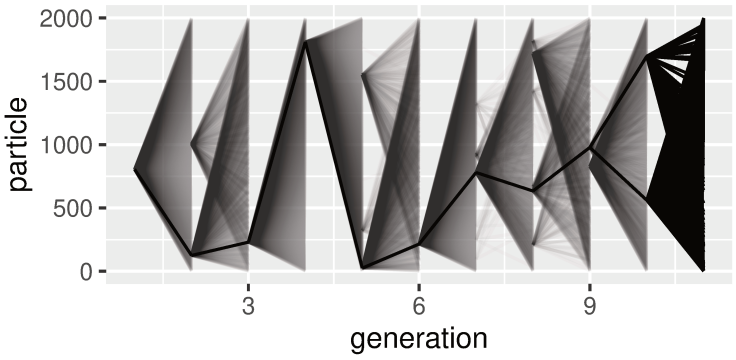


FIGURE 3.11 – Genealogy of the final particles

require to increase the number of particles. However, we can notice that even with a same ancestor for each particle, we have in the final sample around one thousand of different topologies, meaning that this method is efficient for initializing a Metropolis-Hastings algorithm.

15.2 Stability and resilience of the methods

All models are wrong, but we can check the resilience of our methods if the data is not simulated from the model on which we base the inference, but from a slightly different model. These slightly different models are usually too complex to allow a full analysis, justifying the interest of these experiments.

15.2.1 Gamma prior on the length of the tree

As stated before, we tried a more classical prior on the tree, with the same specifications as in Section 13.1.2. Leading to the results presented in Figure 3.12. The reconstructed trees are quite similar to those obtained with our prior. This indicates a small influence of the prior, and justifies our choice as it is of little importance.

15.2.2 Correlated traits

There are no true consensus on the definition of the traits in sign languages. Their definition is arbitrary and it is possible that true underlying character structure — if it exists — would be completely different. In spoken languages the analogous of traits are the characteristics of phonemes (*e.g.* vocalisation, nasalisation, *etc.*), and we know that those characteristics, depending on the language, can be discriminant or not, meaning that the underlying structure is not only complex but variable

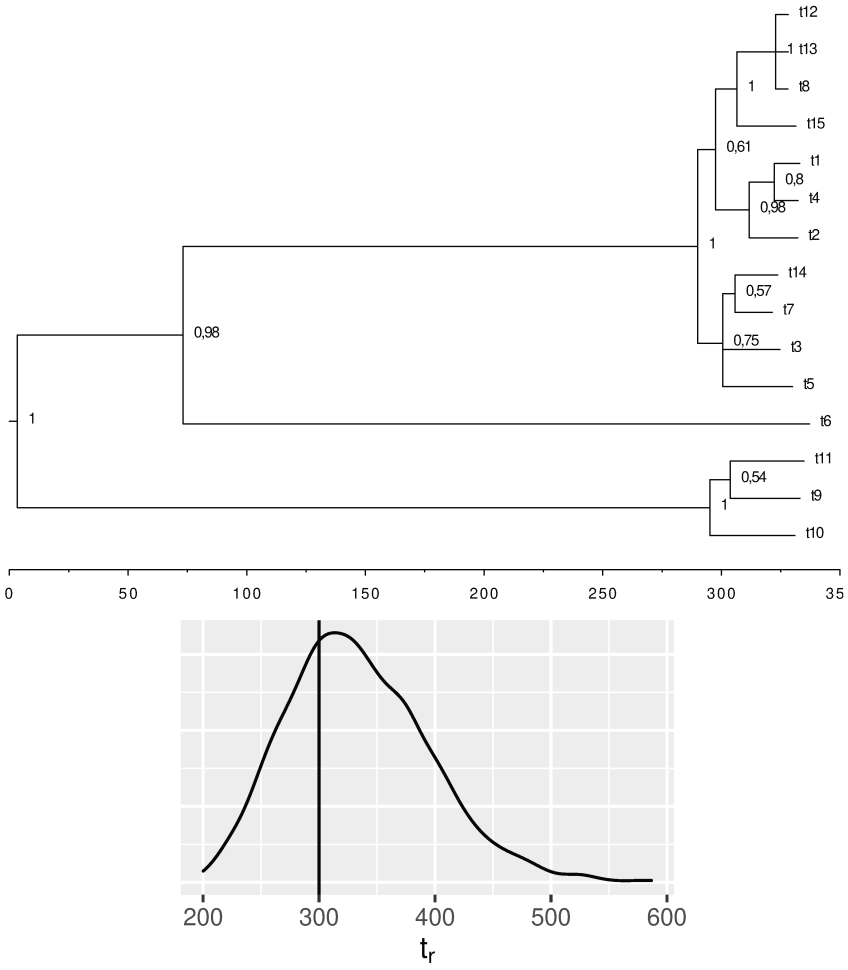


FIGURE 3.12 – Consensus tree and root age posterior distribution for the Gamma prior.

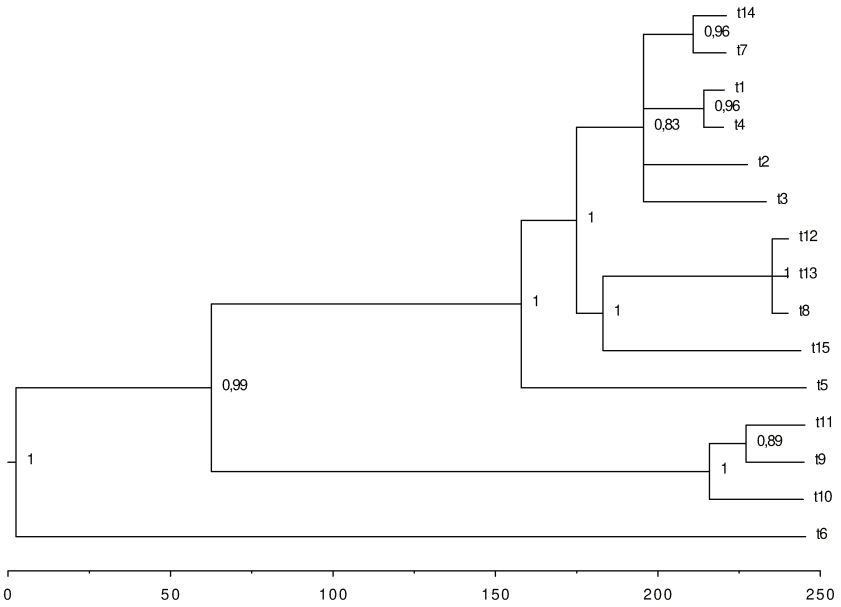


FIGURE 3.13 – Consensus tree for the correlated dataset test.

between the languages. Evidence of such a phenomenon have not yet been found in sign languages.

We ran the experiment on the same tree, with the same characters and values — although with a newly generated dataset, with the particularity that the values for the second character do not come from the above described dynamic. At the end of each edge, the value for the second character is the same as the first one with some noise:

$$D(m,j,k_1) = D(m,j,k_2)\mathbf{1}_{U_{j,m}<1-exp(-\nu*\ell_j)} + \varepsilon_{m,j}\mathbf{1}_{U_{j,m}>1-exp(-\nu*\ell_j)},$$

where k_2 is the character correlated with k_1, which follows the standard dynamic, ℓ_j is the length of the edge going to j, $\varepsilon_{m,j}$ are i.i.d. from π_0 and $U_{j,m}$ are $\mathcal{U}(0,1)$ i.i.d.

The resulting consensus trees are presented in Figure 3.13. Overall the inference does not seem very much affected, the tree posterior is strikingly concentrated around the consensus topology. Strangely, the parameters associated with the second character seem a little bit off, while we would expect both characters to be badly reconstructed.

15.2.3 Selection of transformations

If the set of possible transformations in the implementation is larger than the true one, we can assume that the inference will lose in efficiency. In the previous

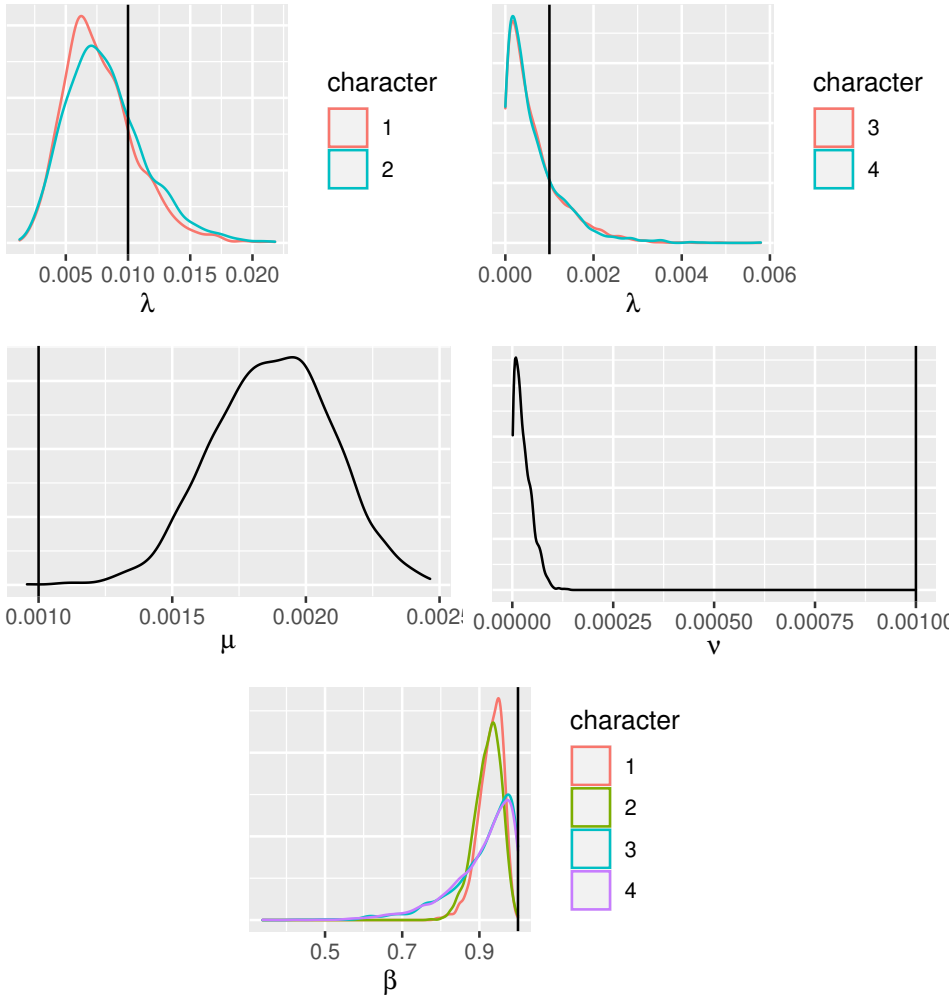


FIGURE 3.14 – Posterior estimations of λ, μ, ν, β and the root age. The true values are indicated by the vertical lines.

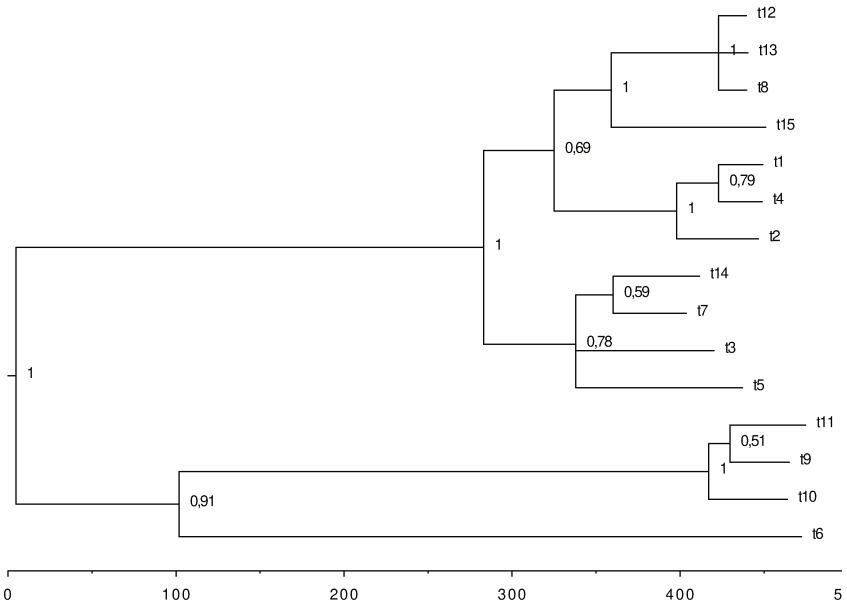


FIGURE 3.15 – True and consensus tree for the unknown transformations sets

model, it is clear that the likelihood of the proposed model is drastically changed if the possible transformations are changed. The identifiability of the model is even unclear, as there are several parameters p that can lead to a same likelihood.

We ran our numerical methods on such a model; the parameters of the experiment are the same as the synthetic exact experiment. The only change is that we did not reduce the number of possible transitions for the two first characters (we have 20 possible transitions instead of the 12 truly present).

In these experiments, the topology is quite accurately reconstructed, while the inference of the other parameter is of lesser quality, with parameter overall underestimated, see Figure 3.16. The inference of the probability of the transformations is not quite satisfying, as we can see in Figure 3.17, there is no significant difference in the reconstituted probability, this comes certainly from the small number of transformations in total on the tree — around a dozen.

However, this is clearly not efficient if the size of $\mathfrak{R}_k$ is too high. For example, if we allow all the pairwise transformations in a character with 30 value we get more than a thousand pairs; the convergence of the numerical methods are then vastly slower and we cannot reach a satisfying equilibrium, and the data for this character can only be explained with noise or apparitions, which is not optimal. We recommend to reduce to less than 30 the size of $\mathfrak{R}_k$.

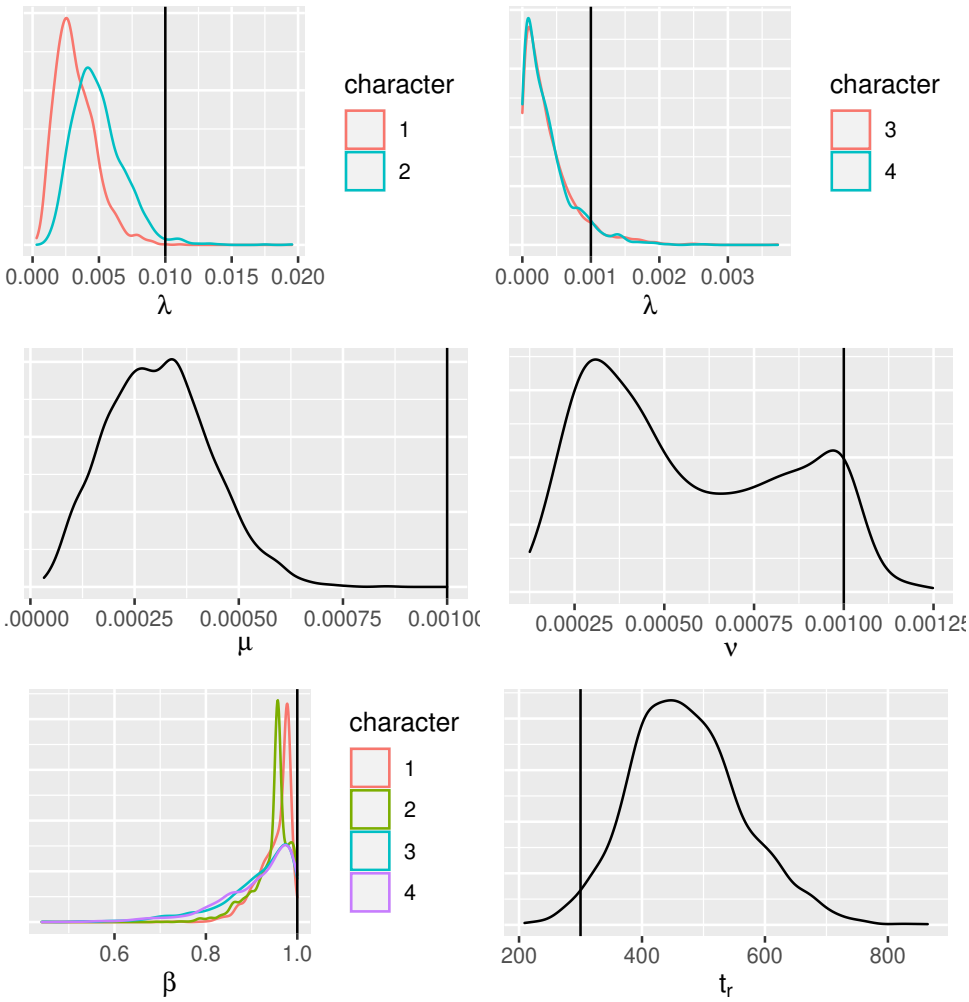


FIGURE 3.16 – Posterior estimations of λ,, ν, β and the root age for the experiments with unknown $\mathfrak{R}_k$. The true values are indicated by the vertical lines.

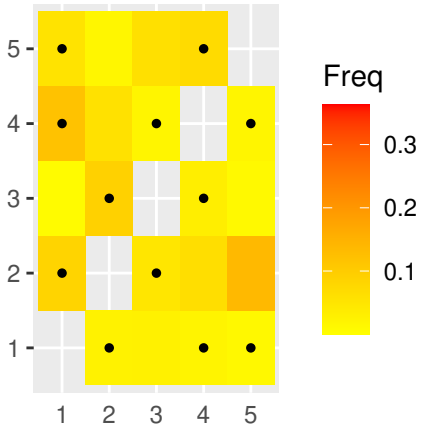


FIGURE 3.17 – Posterior of the transformation repartition p for the first characters the real transformations are indicated by dots. The color represents the posterior intensity.

15.2.4 Iconicity

Some of the signs in the vocabulary are suspected to be highly iconic, that is not to follow the arbitrariness of linguistic sign in the sense of Saussure [De Saussure, 1989]. For example, animals tends to be represented by particular features or by their way of moving; a characteristic sign might be the one for *snake* which represents either the fangs of the snake or its way of crawling on the ground. This has two consequences, the first one is that those signs *should* not undergo too many phonological transformations, as they would lose the link between signified and significant, and the second is that if they are replaced it would be by a particular sign also iconic.

We propose two ways of representing iconicity. For a part of the meanings, 30%, called *iconic*, we propose the following production rules:

1. we generate on the tree values for these meanings as in the assumed model, then we take randomly 4 of these values and each language — except the one having produced these values — borrows equiprobably one of these 4 words. This represents the fact that iconic meaning tend to be represented as one or two archetypal sign.

2. we choose randomly for each of these meanings a chacacter among the 4 that will not evolve along the tree. Only some characters will be different between languages; the other being fixed at the value they had at the root. This tends to represent the fact that some part of signs only are iconic (for example, the two handedness of signs).

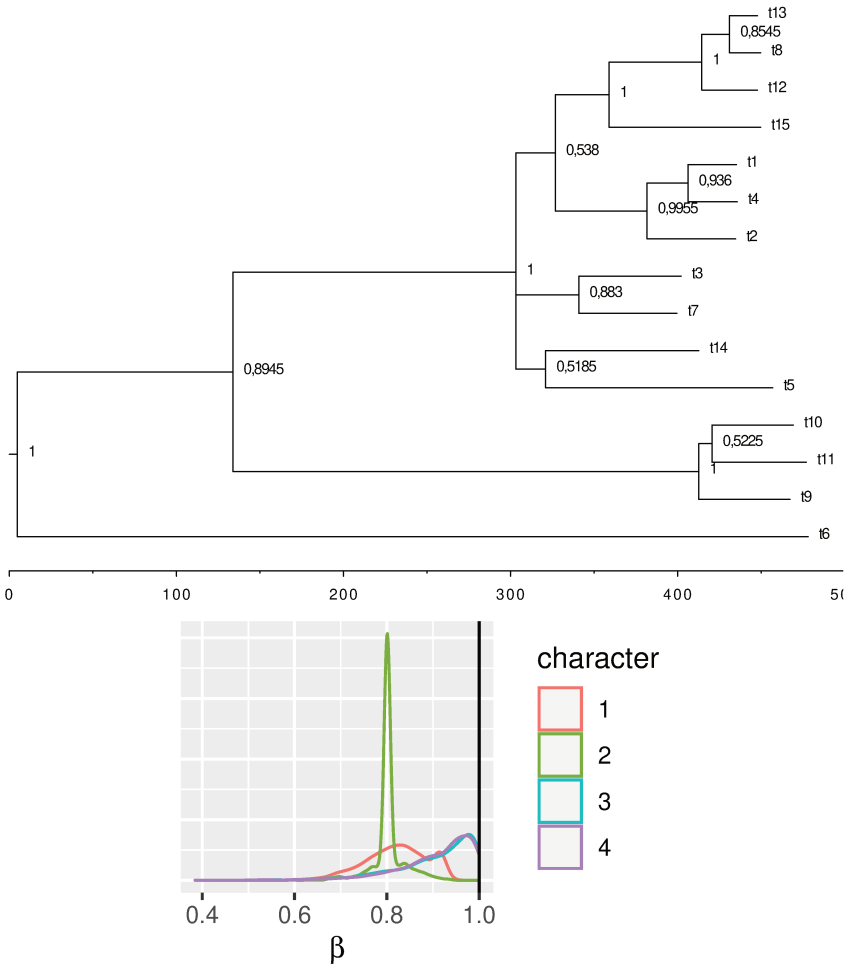


FIGURE 3.18 – Consensus tree and posterior on β for the first iconicity example.

We chose the same parameters as in the previous sections.

Figure 3.18 and 3.19 presents the results for a dataset coming from respectively each of the inconicity representation described above. Although the topology of the consensus tree is satisfying, the root age and the parameters are less accurate. Noticeably, in the case of the first iconicity the parameter β presents significatively low values compared to the others. This might indicate that such a behaviour on a real dataset might be explained by iconicity.

15.2.5 Data simulated from a forest

In this last example, which is of particular interest for sign language application, we concatenate the dataset generated independently from a 11-leaved tree and a

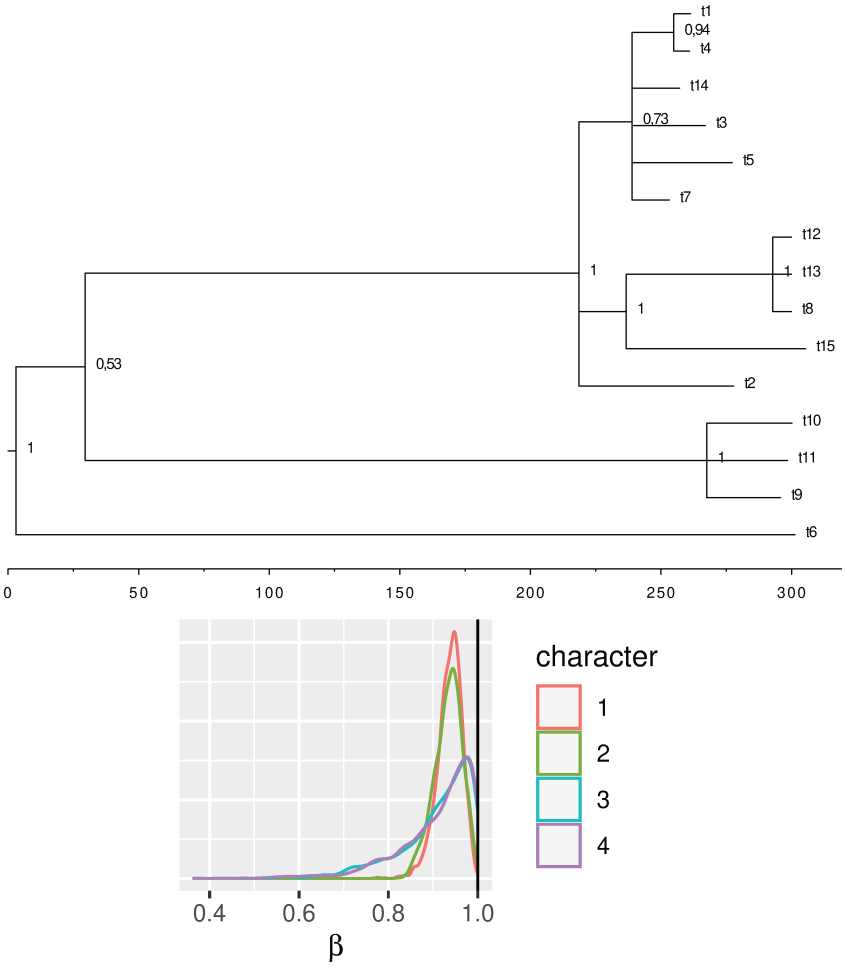


FIGURE 3.19 – Consensus tree and posterior on β for the second iconicity example.

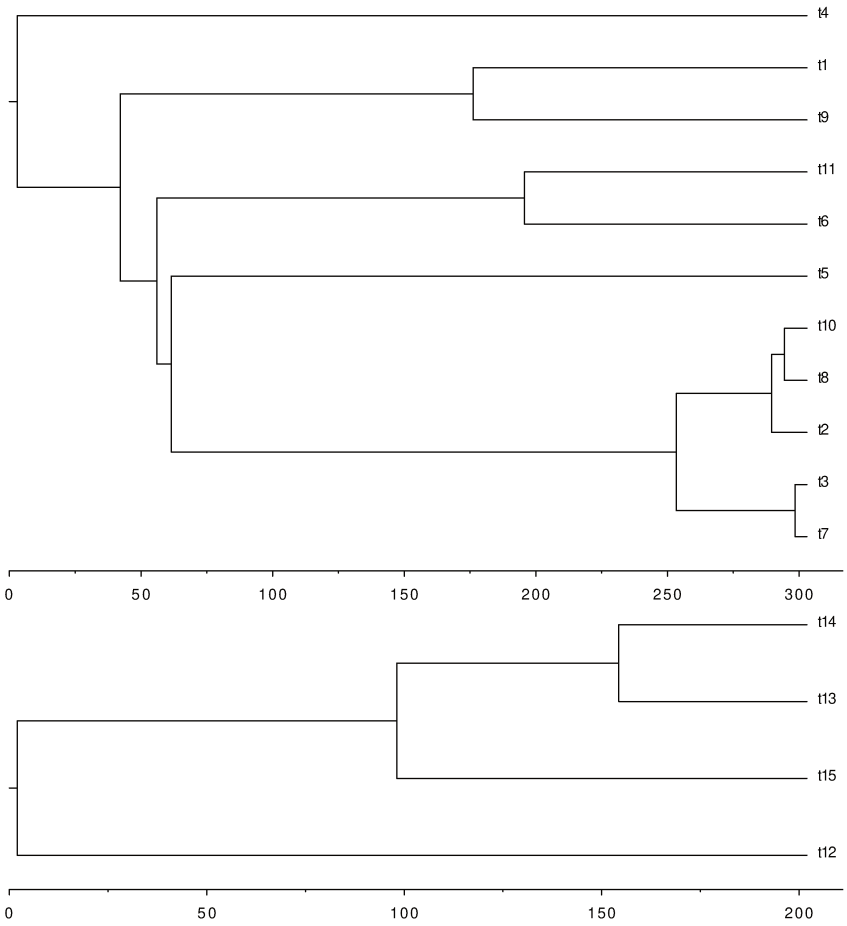


FIGURE 3.20 – Two trees used for the forest example

4-leaved tree represented in Figure 3.20. That is we try to infer a single tree where there should be two. This is a quite important case as we are never sure that some languages are indeed related; in our case the proposed families of sign languages are almost all debated.

The results are presented in Figure 3.21. Interestingly enough, the two groups of languages are reconstructed as subtrees with rather long edges. Clearly, if the evolution parameter were higher this would not be the case, especially for a small number of characters and a small number of possible values. The evolution parameter however are quite off, as they must be quite important to explain this unique tree. The root is not so old, which is an effect of the prior.

This result is clearly the best we could hope for, as it shows that the topology is at least accurately reconstructed, while such a misspecification is prone to leading to spurious inferences.

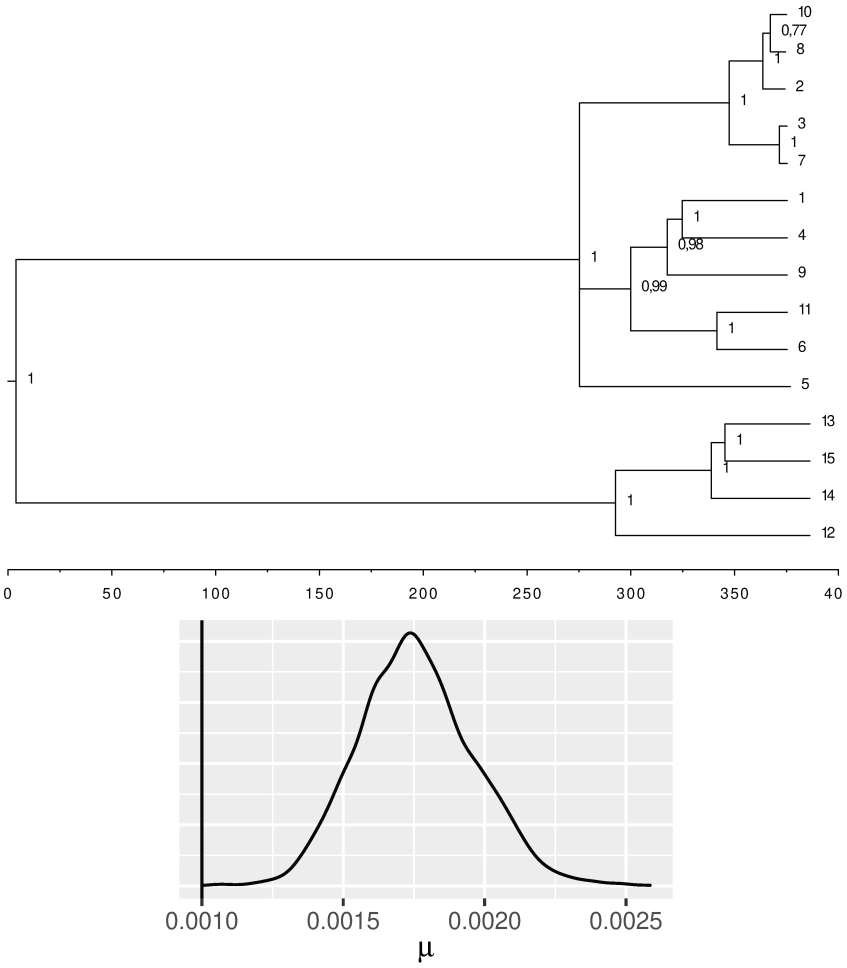


FIGURE 3.21 – Results of the inference for the topology and the apparition parameter μ on the forest synthetic dataset.

16 Application to sign languages phylogenies

16.1 Dataset

As stated before, we apply the previous method to the previously described sign languages dataset. As the dataset is too large for being handled directly, we had to reduce it, in accordance with the specialists' recommendation.

16.1.1 Choice of the languages

Little is known about families of sign languages. Our model assumes that the languages are related so it could be problematic if the selected languages are clearly unrelated. In practice, as shown in Section 15, this should not be an issue. Far more problematic is the existence of *pidgin* sign languages, that are related to several other sign languages. For this reason, we had reasonable evidence to remove from our study Portuguese, Swedish, Icelandic, Turkish and Brazilian. We also include the Asian sign languages — Hongkong, Chinese, Japanese and Taiwan — which we constrain, following the experts' knowledge, to form an outgroup as they should not be related to the others.

Our implementation has been able to handle around 15 languages without problems, higher number of language can lead to memory overload.

A digression on dialects In phylogenetics, we claim that we study species or languages. However, contrary to some philosophers, we do not pretend to study universals, but to study individual realisations of the universals that in our case are languages. This is an idea to answer a famous ancient problem[4]. The language, or the species in itself cannot be summarised in a dataset, as it only exists as a theoretical construction; any sample taken from the real world will come from an individual, subject to variations. Thus, it is a misuse of language to say that our dataset is constituted of "languages". Our dataset is constituted of individual realisations of the languages studied. Our tree should represent the phylogeny of these variants. However, as most of the possible variants for a single language are close, we can say that running the same analysis with any of the variant would lead to the same result.

[4] *For the moment, I shall naturally decline to say, concerning genera and species, whether they subsist, whether they are bare, pure isolated conceptions, whether, if subsistent, they are corporeal or incorporeal, or whether they are separated from or in sensible objects, and other related matters. This sort of problem is of the very deepest, and requires more extensive investigation.* [Porphyrios, c. 268].

16.1.2 Choice of the characters

The choice of characters is a matter of primary concern. It is in fact the most important choice to be done, as it can drastically reduce the efficiency of numerical methods or even lead to biased results, as we have seen in examples on simulated datasets, in Section 15.

First, the question of iconicity is even more stringent in sign languages [Lepic et al., 2016], meaning that the evolution of the sign is correlated with the meaning in a way that is not accounted for in the model. In particular, there is a high certainty that two-handed signs tends to represent some categories of meanings, in particular those involving two agents (such as *to sign*, or *to meet*). In other words, the evolution for different meaning are not the same.

On a numerical side, characters with binary values may be problematic as they tend to evolve with a very slow rate, leading to a higher likelihood of an apparition compared to a transformation for this particular character. This can also lead to an overconcentrated posterior distribution, as a difference on a binary character will be explained by a renewal of the word, rather than a transformation.

On the other side of the spectrum, characters with too many values tend to lead to inefficient numerical methods, as this increase the overall dimension of the problem.

Too few characters will reduce the difference between the two dynamics we describe, while too many characters also increase the computational cost. For now, our implementation can handle around 5 characters.

The chosen characters should be decorrelated and contain as much information as possible. Classically, signs are described through 3 or 4 features: handshape, movement, part of the body at which the sign occurs, and sometimes orientation of the hand. However there are 48 possible handshapes for signs. We thus prefer to use, following the advices of the expert linguists, a reduced version of this character, with only 9 modalities. The same is true for the body part. These characters tends to be mostly decorrelated, as there is little incompatibility between the values of each characters; a more precise study of these correlations remains to be done. We also added the binary character accounting for the two-handedness of the signs, this last character was the most debatable.

In the end, we chose 5 characters that covers most of the classical description of the signs in most of the languages:

1. Handpart: describes the orientation of the hand;

2. Handshape: describes the form of the hand during the sign, accounting in 9 values of the fingers state;

3. Point of articulation: designates the area of the body around which revolves the sign;

4. Two handedness: is a boolean encoding the fact that both hands are used for signing this word;

5. Movement encodes the direction with respect to the signer of the movement — if there is one.

16.1.3 Choice of the meanings

We have not changed the selected meanings, 100 being already a small number for our work, the meanings are presented in Section 18. We can see that some of the meanings will be prone to iconicity, in particular those related to animals or action, while other will most probably avoid this issue, the best example being interrogatives. We tried to extract sub-datasets with words of known iconicity without much results as there were too few data points to have a clear posterior signal.

16.1.4 Summary of the dataset

This reduced dataset can be summarised by the pairwise distance between the languages. We chose as distance the number of different values in the previous matrix, and the resulting distances are presented in Figure 3.22.

16.2 Results

The parameters of the experiments are those described above. We launched the algorithm twice and once with longer mutation steps with similar results. All the dates are given before 2000.

We present in Figure 3.23 the resulting consensus trees from two independent realisations of the same procedure, to check the stability of the method. The most noticeable feature is the stability of some subgroups, such as Russian, Lithuanian and Ukrainian sign languages, or Austrian and German sign languages. Some of the other languages have unclear positions, for example French Sign Language or Czech Sign Language. This was to be expected given the dataset. Figure 3.24 represents a densitree of the posterior sample, showing the uncertainty on the Asian clade.

As for the parameters, Figure 3.25 provides the posterior estimations. The characters exhibits different behavior, in particular the first one, with fewer transformations associated with smaller probability β. This character being the one transformed for the purposes of the analysis it may be of interest.

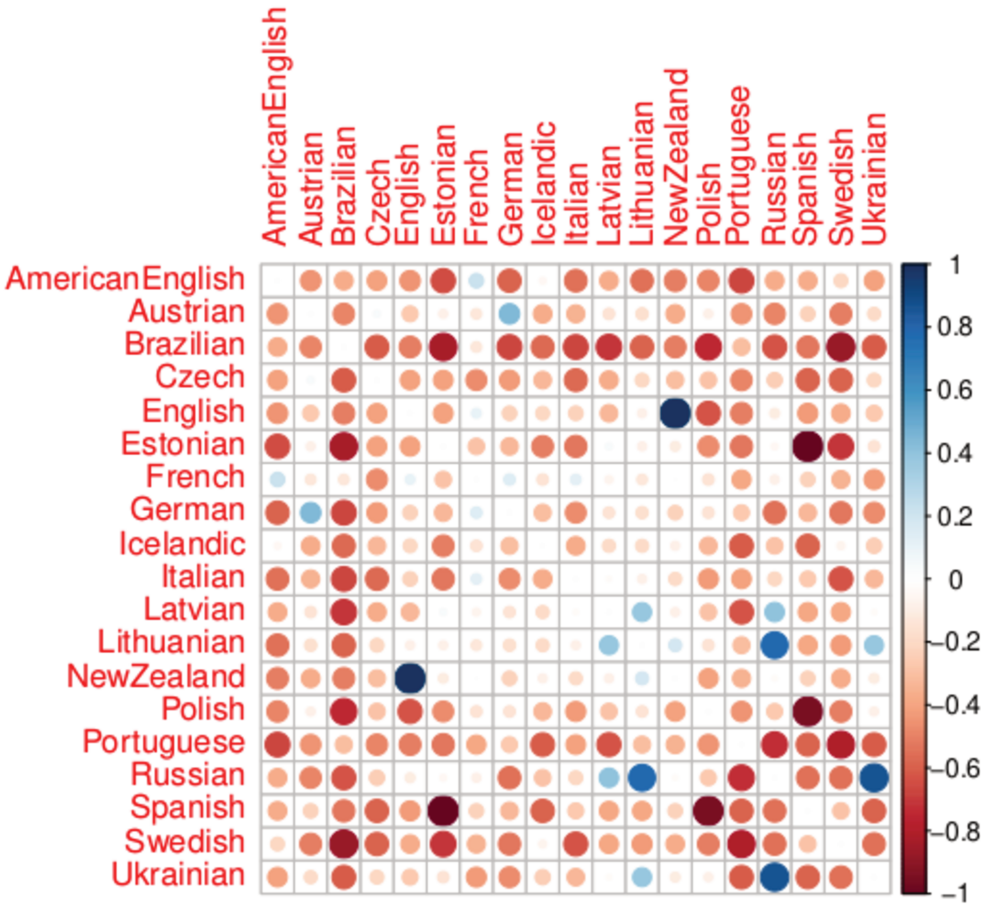


FIGURE 3.22 – Normalised pairwise distance between some of the languages.

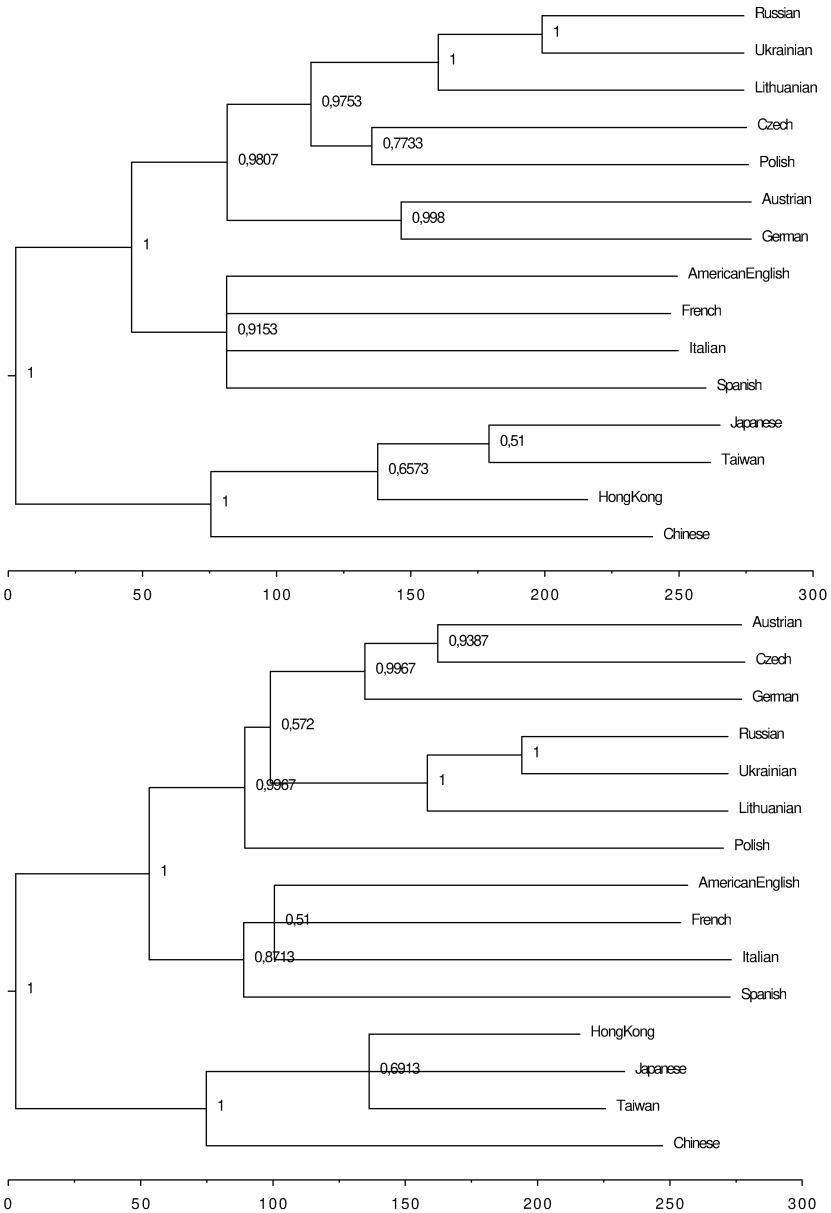


FIGURE 3.23 – Two consensus trees for OldLSF-Asian dataset.

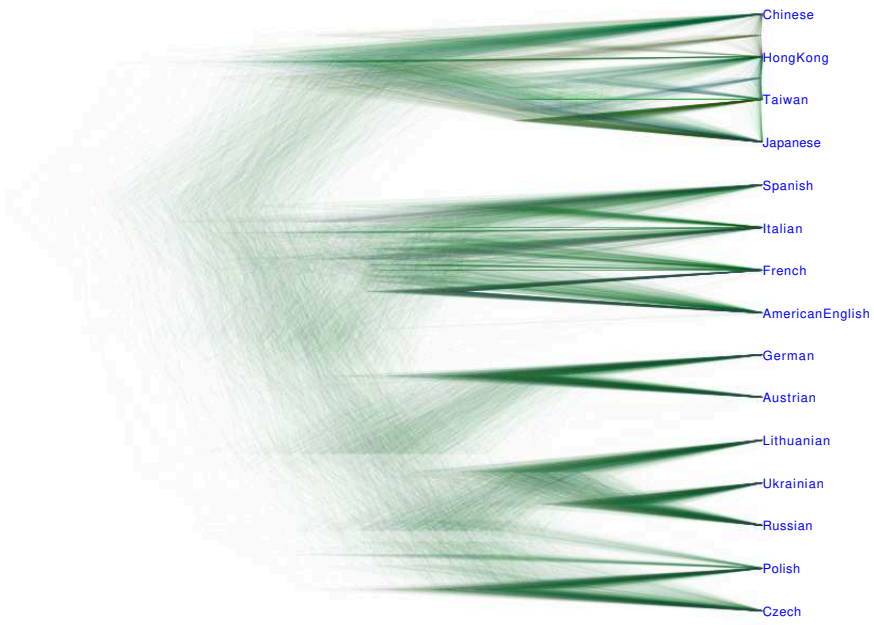


FIGURE 3.24 – Densitree for the OldLSF-Asian dataset.

These results, that have appeared stable through the runs, confirm several hypothesis of the linguists. First the existence of an *Eastern-European* clade of languages, especially for the Russian, Ukrainian and Lithuanian SL. The western group is less clearly defined, most probably because of the interactions between those languages. On the Asian side, proximity of Taiwanese and Japanese SL is debatable, but it seems that the Hongkong SL is an intermediary between China and the other countries. The relatively short edge between the two subgroups of languages do not incitate to think Asian and European SL are clearly separated, even though contacts were scarce, converging evolution might explain this behaviour — especially in the context of high iconicity.

The fourth character, Two Handedness, presents a particular behaviour. There is no transformation for this character, leaving the changes to renewal and noise. This means that the character, although carrying phylogenetic signal, does not fit into our phonological transformation model. The fifth character has an associated β significatively lower than the other characters, this also indicates that the character does not fit clearly into the *systematic* phonological rule model by which we were inspired for our work. These two results are of particular interest for linguists, as it informs the comparison between spoken languages and sign languages, showing that the notion of sound laws might not be valid for some of the characters of signs.

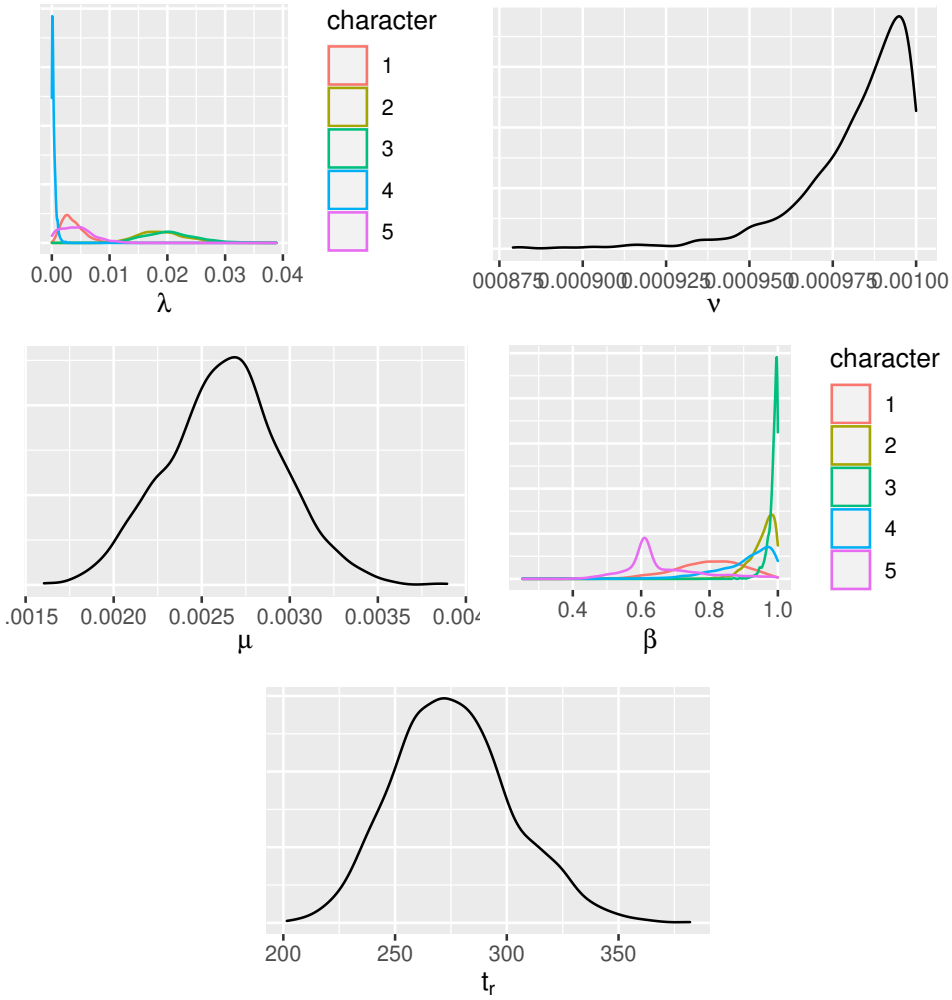


FIGURE 3.25 – Posterior estimations of λ, μ, ν, β and the root age for the OldLSF-Asian dataset.

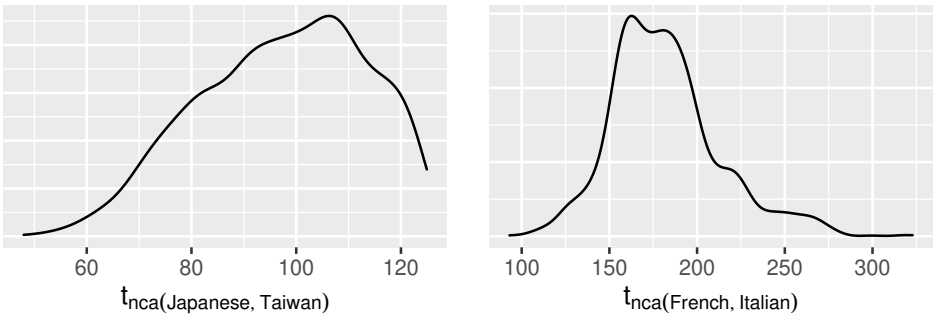


FIGURE 3.26 – Posterior estimations for internal nodes, designated as nearest common ancestors of pairs of languages.

On the side of internal node ages, there are still lots of uncertainty. We present some of the results in Figure 3.26. For the Japanese and Taiwanese SL, the date seems to correspond to the Japanese invasion of Taiwan in 1895. For the French and Italian, this age is highly constrained by the French and American constrain we added, but it corresponds to the Italian unification at the end of the 19th century.

17 Conclusion

This model is one of the first that can handle "phonological" datasets, although with much care in the selection of the dataset. This can also mean that other applications might be though about, such as cultural products, for example ceramics: in Karasik and Smilansky [2011] the author compares curvatures and thickness of different ceramics according to a discretised scale. This model allows to trace discontinuity in evolutions starting from the surface forms, while these discontinuities were assessed by hand in the previous works. We will be working on applying this method to spoken languages in the near future, with a particular care on interaction between characters and their selection. Nevertheless, the model is debatable. It totally ignores interactions between the characters — as in some conditional sound laws — and it strongly relies on the perfect alignment of the data, which makes difficult the study spoken languages like French or Irish that underwent extensive elisions.

Although the model behaves well on simulated datasets, with quite accurate topologies, the evolution parameters are not trustworthy in general under misspecification. The results on sign languages are debatable, and might not inform the phylogeny of those languages, in particular given the exceptional complexity of their evolution.

The numerical cost is the major drawback of our method, albeit SMC was the only efficient method we found, we cannot compute the marginal likelihood of our model, which would have been a major achievement.

On this model, several straightforward improvement can be made. The first of which is the adjunction of some inhomogeneity between edges, for example by adding "catastrophies", the noise could be differentiated between characters, which would in fact allow for "variable selection" or at least moderation of the importance of each variable. To be applied to most spoken languages, this model would require to be extended to non-aligned datasets. Learning forests instead of trees would also be a great achievement, probably not that far away, this is not restricted to our model, as Stochastic Dollo could also be a starting point for this. Future models will require more advanced method, either for the exploration of the posterior when the likelihood is available, or more probably the use of likelihood free methods to avoid falling in the lure for latent variables.

18 Notations

— $\{1, \dots, M\}$ meanings studied

— k one of the traits

— $\{1, \ldots, n_k\}$ possible state for character k

— $D(m, j, k)$ value of character k for the meaning m at leaf j

— D set of all the data

— $\mathfrak{R}_k$ possible transformations for character k (that is a function kernel from $\{1, \ldots, n_k\}$ into itself)

— r a transformation

— $M_{noise}(\ell)$ transition matrix associated with the noise with length ℓ

— λ_k Poissonian rate of apparition of a transformation for character k

— g a topology for the tree

— E set of edges in g

— e an edge in E

— $U(m, j, k)$ value of meaning m, character k at (internal) node j

— R_e the set of transformations for a given character for edge e

— ℓ a set of length of the edges of g

— ν rate of the noise

— p_k probability of each transformation in $\mathfrak{R}_p$, to colour the Poisson process

— μ rate of renewal of the meanings

— B_e^m apparitions for meaning m on edge e

— $\pi_{e,\tau}$ distribution of the newly appeared cognate for a certain character at a certain point (e, τ) of the tree

— π_0 initial distribution for a given character

List of meanings

The 100 meanings constituting the list we study is presented in Figure 3.1, with the associated Semantic Groups.

Meaning	Swadesh 1955 Semantic Group	Meaning	Swadesh 1955 Semantic Group
bird	Animals	sister	Kinship
animal	Animals	right	Location
dog	Animals	with	Locatives
fish	Animals	kill	Miscellaneous
louse	Animals	not	Miscellaneous
snake	Animals	other	Miscellaneous
worm	Animals	name	Miscellaneous
fat	Body Parts and Substances	cat	Animals
feather	Body Parts and Substances	full	Descriptives
meat	Body Parts and Substances	heavy	Descriptives
tail	Body Parts and Substances	moon	Natural Objects and Phenomena
egg	Body Parts and Substances	pig	Animals
blood	Body Parts and Substances	fire	Natural Objects and Phenomena
vomit	Body Sensations and Activities	river	Natural Objects and Phenomena
live	Body Sensations and Activities	dust	Natural Objects and Phenomena
die	Body Sensations and Activities	earth	Natural Objects and Phenomena
white	Colors	mountain	Natural Objects and Phenomena
yellow	Colors	stone	Natural Objects and Phenomena
red	Colors	snow	Natural Objects and Phenomena
black	Colors	sea	Natural Objects and Phenomena
green	Colors	rain	Natural Objects and Phenomena
because	Correlatives	water	Natural Objects and Phenomena
if	Correlatives	ice	Natural Objects and Phenomena
dance	Cultural Objects and Activities	salt	Natural Objects and Phenomena
work	Cultural Objects and Activities	star	Natural Objects and Phenomena
play	Cultural Objects and Activities	sun	Natural Objects and Phenomena
count	Cultural Objects and Activities	wind	Natural Objects and Phenomena
hunt	Cultural Objects and Activities	sing	Oral Activities
rope	Cultural Objects and Activities	laugh	Oral Activities
dirty	Descriptives	man	Persons
dull	Descriptives	child	Persons
wet	Descriptives	woman	Persons
bad	Descriptives	person	Persons
smooth	Descriptives	wood	Plants & Plant Parts
sharp	Descriptives	flower	Plants & Plant Parts
warm	Descriptives	grass	Plants & Plant Parts
new	Descriptives	tree	Plants & Plant Parts
good	Descriptives	leaf	Plants & Plant Parts
dry	Descriptives	lie	Position and Movement
old	Descriptives	stand	Position and Movement
how	Interrogatives	sit	Position and Movement
what	Interrogatives	all	Quantitatives
when	Interrogatives	short	Size
where	Interrogatives	narrow	Size
who	Interrogatives	thin	Size
wife	Kinship	long	Size
husband	Kinship	wide	Size
father	Kinship	night	Time Periods
mother	Kinship	day	Time Periods
brother	Kinship	year	Time Periods

TABLE 3.1 – List of meanings studied, and semantic group associated

Part 4
Likelihood free numerical methods

This part is a slightly modified version of a work I did with Christian P. Robert, Robin J. Ryder and Julien Stoehr, published as Clarté et al. [2019b].

19 Introduction

Approximation Bayesian computation (ABC) is a computational method which stemmed from population genetics to deal with intractable likelihoods, that is models whose likelihood cannot be (easily) computed but which can be simulated from [Tavaré et al., 1997, Beaumont et al., 2002]. Since then, it has been applied to numerous other fields: see for example Toni et al. [2008], Csilléry et al. [2010], Moores et al. [2015], Sisson et al. [2018]. The principle of the method is to simulate pairs of parameters and pseudo-data from the prior predictive, keeping only the parameters that bring the pseudo-data close enough (within a pseudo-distance ε) to the observed data. Proximity is often defined in terms of a projection of the data, called a summary statistic. In general, practitioners of ABC aim to use informative summary statistics and select ε to be as small as possible, since this leads to a higher-quality approximation. From the start, this method has suffered from the curse of dimensionality in that the dimension of the parameter to be inferred imposes a lower bound on the dimension of the corresponding summary statistic to be used (results by Fearnhead and Prangle [2012] and Li and Fearnhead [2018] imply that the dimension of the summary statistic should be identical to the dimension of the parameter). This constraint impacts the range of the distance between observed and simulated summaries, with the distance choice having a growing impact as the dimension increases. Reducing the dimension of the summary is thus impossible without reducing the dimension of the parameter, which sounds an impossible goal unless one infers about one parameter at a time, suggesting a Gibbs sampling strategy where a different and much reduced dimension summary statistic is used for each component of the parameter. The purpose of this paper is to explore and validate this strategy, producing sufficient conditions for the convergence of the resulting algorithms.

Additionally, the Gibbs perspective allows us to account for the current values of the other components of the parameter and therefore to shy away from simulating from the prior which is an inefficient proposal. This feature connects this proposal with earlier solutions in the literature such as the Metropolis version of Marjoram et al. [2003] and the various sequential Monte Carlo schemes [Toni et al., 2008, Beaumont et al., 2009]. There have been earlier ABC versions with Gibbs

features, including Wilkinson et al. [2011], where a two-stage ABC-within-Gibbs algorithm is proposed towards bypassing the intractibility of one of the conditional distributions used in their Gibbs sampler. Since the other conditional distribution is simulated exactly, there is no convergence issue with this version. Note also that the summary statistics used in that paper are not chosen for dimension reduction purposes. Kousathanas et al. [2016] also run a Gibbs-like ABC algorithm that assumes the availability of conditionally sufficient statistics to preserve the coherence of the algorithm. Rodrigues et al. [2019] propose another Gibbs-like ABC algorithm in which the conditional distributions are approximated by regression models.

A Gibbs version of the ABC method offers a range of potential improvements compared with earlier versions, induced in most cases by the dimension reduction thus achieved. First, in hierarchical models, conditioning decreases the number of dependent components, and some of the conditionals may be available in closed form, which makes the approach only semi-approximate. Second, since the conditional targets live in spaces of low dimension, they can more easily be parametrised by low dimension functions of the conditioning terms. This justifies using a restricted range of collection of statistics, which may in addition depend on other parameters. Third, reducing the dimension of the summary statistic improves the approximation since a smaller tolerance can then be handled at a manageable computing cost.

This heuristic leads us to propose in Section 20 a generic algorithm called ABC-Gibbs. To show the theoretical validity of this idea, we successively show that, under some conditions:

i) for all $\varepsilon > 0$, our ABC-Gibbs converges to a certain limiting distribution ν_ε in total variation distance,

ii) when $\varepsilon \to 0$, $\|\nu_\varepsilon - \nu_0\|_{TV} \to 0$, with ν_0 a distribution,

iii) ν_0 is the limiting distribution of Vanilla ABC with tolerance ε set to 0.

The first result corresponds to Theorem 5 in the general case; Theorem 8 states this result for hierarchical models under looser assumptions. The second result is a consequence of Theorem 7. The last result follows from the results of Section 25.

In all this part, we define Θ_j as the domain of θ_j. For the proofs that pertain to model (2) we define $\mathcal{A}$ as the domain of α and $\mathcal{B}$ as the domain of μ. For a space E, $\mathcal{P}(E)$ is the space of the probability distributions over E.

20 Approximate Bayesian Gibbs sampling

20.1 Vanilla approximate Bayesian computation

Approximate Bayesian computation methods, summarised in Algorithm 2.3, provide a technique to sample posterior distributions when the corresponding likelihood $f(x|\theta)$ is intractable, that is the numerical value $f(x|\theta)$ cannot be computed in a reasonable amount of time, but the model is generative, that is it allows for the generation of synthetic data given a value of the parameter. Given a prior distribution on the parameter θ, it builds upon samples from the associated prior predictive $(\theta^{(i)}, x^{(i)})_{i=1,\ldots,N}$ by selecting pairs such that the pseudo-data $x^{(i)}$ stand in a neighbourhood of the observed data $x^\star$.

Since both the simulated and observed dataset may belong to a space of a high dimension, the neighbourhood is usually defined with respect to a summary statistic $s(\cdot)$ of a lesser dimension and an associated distance d (see Marin et al., 2012 for a review). Fearnhead and Prangle [2012] show that the optimal statistic is of the same dimension as the parameter θ; in practice, the choice of s remains a crucial issue.

The output of Algorithm 2.3 is a sample distributed from an approximation of the posterior [Tavaré et al., 1997, Sisson et al., 2018]. Its density is written, with a notation coherent with the next sections:

$$\pi_\varepsilon\{\theta \mid s(x^\star)\} \propto \int \pi(\theta) f(x \mid \theta) \mathbf{1}_{d\{s(x), s(x^\star)\} < \varepsilon} \, \mathrm{d}x.$$

This approximation depends on the choice of both the summary statistic s and the tolerance level ε. Frazier et al. [2018] show its consistency, namely that when the number of observations tends to ∞ and the tolerance tends to 0 at a proper relative rate, the approximate posterior concentrates at the true value of the parameter, albeit as a posterior distribution associated with the statistic s, rather than the true posterior, when s is not sufficient. The shape of the asymptotic distribution is further discussed in Li and Fearnhead [2018] and Frazier et al. [2018].

More to the point, given a fixed number of observations, the approximate posterior also converges to the posterior $\pi\{\theta \mid s(x^\star)\}$, rather than to the standard posterior $\pi(\theta \mid x^\star)$, when the tolerance level goes to 0. In practice, however, the tolerance level cannot be equal to zero and is customarily chosen as a simulated distance quantile [Sisson et al., 2018]. In practice, a large sample of pseudo-observations is generated from the prior predictive and the corresponding distances to the observations are computed. We use the term reference table for this collection of parameters and distances. The tolerance is then derived as a small quantile of these distances.

20.2 Gibbs sampler

The Gibbs sampler, first introduced by Geman and Geman [1984] and generalised by Gelfand and Smith [1990], is an essential element in Markov chain Monte Carlo methods [Robert and Casella, 2004, Gelman et al., 2013]. As described in Algorithm 4.1, for a parameter $\theta = (\theta_1, \ldots, \theta_n)$, it produces a Markov chain associated with a given target joint distribution, denoted π, by alternatively sampling from each of its conditionals.

Input: number of iterations N, starting point $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_n^{(0)})$.
Output: a sample $(\theta^{(1)}, \ldots, \theta^{(N)})$.
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, n$ **do**
    $\theta_j^{(i)} \sim \pi(\cdot \mid \theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)})$

Algorithm 4.1: Gibbs sampler

Gibbs sampling is well suited to high-dimensional situations where the conditional distributions are easy to sample. In particular, as illustrated by the long-lasting success of the BUGS software [Lunn et al., 2010], hierarchical Bayes models often allow for simplified conditional distributions thanks to partial independence properties. Considering for instance the common hierarchical model [Lindley and Smith, 1972, Carlin and Louis, 1996] defined by

$$x_j \mid \mu_j \sim \pi(x_j \mid \mu_j)\,, \qquad \mu_j \mid \alpha \overset{\text{i.i.d.}}{\sim} \pi(\mu_j \mid \alpha)\,, \qquad \alpha \sim \pi(\alpha)\,. \tag{2}$$

The joint posterior of $\mu = (\mu_1, \ldots, \mu_n)$ conditional on α then factorises as

$$\pi(\mu \mid x_1, \ldots, x_n, \alpha) \propto \prod_{j=1}^{n} \pi(\mu_j \mid \alpha)\pi(x_j \mid \mu_j).$$

This implies that the full conditional posterior of a given μ_j only depends on α and x_j, independently of the other (μ_ℓ, x_ℓ)'s.

20.3 Component-wise ABC

When handling a model such as (2) with both a high-dimensional parameter and an intractable likelihood, the Gibbs sampler cannot be implemented as the conditionals are unavailable, while the vanilla ABC sampler is highly inefficient. This curse of dimensionality attached to the ABC algorithm is well documented [Li and Fearnhead, 2018].

Bringing both approaches together may subdue this loss efficiency, by sequentially sampling from the ABC version of the conditionals, whose density $\pi_{\varepsilon_j}(\cdot \mid s_j(x^\star, \theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}))$ is proportional (see (20.1)) to

$$\int \pi(\theta_j \mid \theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}) \\ f(x \mid \theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}, \theta_j, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}) \mathbf{1}_{d\{s_j(x), s_j(x^\star)\} < \varepsilon_j} \, \mathrm{d}x \,.$$

Each step in Algorithm 4.1 is then replaced by a call to Algorithm 2.3, conditional on the other components of the parameter. We obtain a generic componentwise approximate Bayesian computational method, summarised as Algorithm 4.2.

This algorithm can be analysed as a variation of Algorithm 2.3 in which the synthetic data $x^{(i)}$ are simulated from the conditional posterior predictive, rather than from the prior predictive. This may result in simulating both parameters and pseudo-data component-wise from spaces of smaller dimension. This also allows the use of statistics of lower dimension, as exemplified in Section 24.

Each stage j of the algorithm now requires its own tolerance level ε_j and statistic s_j. This statistic can be a function of the observations, but also of the other parameters $(\theta_i)_{i \neq j}$ which are conditioned upon at stage j. Typically, θ_j is of dimension 1 and so s_j should also be of dimension 1, per the results of Fearnhead and Prangle [2012]. Finding a good unidimensional statistic for each θ_j in ABC-Gibbs may prove easier than finding a good high-dimension statistic for Vanilla ABC.

Input: number of iterations N, starting point $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_n^{(0)})$, thresholds $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$, statistics $s_1, \ldots, s_n$, distances on the statistics $d_1, \ldots, d_n$.
Output: a sample $(\theta^{(1)}, \ldots, \theta^{(N)})$.
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, n$ **do**
    $\theta_j^{(i)} \sim \pi_{\varepsilon_j}\{\cdot \mid s_j(x^\star, \theta_1^{(i)}, \ldots, \theta_{j-1}^{i}, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)})\}$

Algorithm 4.2: ABC-Gibbs

If $\varepsilon_j = 0$ and if s_j is a conditionally sufficient statistic, the corresponding jth step in Algorithm 4.2 is an exact simulation from the corresponding conditional. In particular, if some of the conditional distributions can be perfectly simulated, this cancels the need for an approximate step in the algorithm. In practice, to simulate from the approximate conditional, and similarly to Algorithm 2.3, we take ε_j as an empirical distance quantile. In other words, for the jth component of the parameter, conditional on the other components, we simulate a small reference

table from its conditional prior and output the parameter associated with the smallest distance.

At first, the purpose of this algorithm may sound unclear as the limiting distribution and its existence are unknown. As shown in Theorem 5, convergence can indeed be achieved, based on a simple condition. For simplicity's sake, we initially only consider the case when $n = 2$ in Algorithm 4.2.

Theorem 5. *Assume that there exists* $0 < \kappa < 1/2$ *such that*

$$\sup_{\theta_1, \tilde{\theta}_1} \|\pi_{\varepsilon_2}[\cdot \mid s_2(x^\star, \theta_1]) - \pi_{\varepsilon_2}\{\cdot \mid s_2(x^\star, \tilde{\theta}_1)\}\|_{TV} = \kappa.$$

The Markov chain produced by Algorithm 4.2 then converges geometrically in total variation distance to a stationary distribution ν_ε*, with geometric rate* $1 - 2\kappa$*.*

Proof. In this proof, we drop the conditionings on $x^\star$, s_1, and s_2, as they have no use in the computations and create a notational burden.

Let Θ_j be the domain of θ_j. For a space E, $\mathcal{P}(E)$ is the space of the probability distributions over E.

We only need to prove that the Markov chain $(\theta_1^{(i)})_{1 \leq i \leq N}$ has a stationary distribution. We show that $Q : \mathcal{P}(\Theta_1) \to \mathcal{P}(\Theta_1)$, the mapping associated with the transition kernel, is a contraction; that is, we prove that there exists $L > 1$ such that for all ν and $\tilde{\nu}$ in $\mathcal{P}(E)$

$$\|Q\nu - Q\tilde{\nu}\|_{TV} \leq L\|\nu - \tilde{\nu}\|_{TV}.$$

To build a coupling between $Q\nu$ and $Q\tilde{\nu}$ we construct a coupling kernel $\tilde{Q} : \mathcal{P}(\Theta_1 \times \Theta_1) \to \mathcal{P}(\Theta_1 \times \Theta_1)$, which takes a coupling ξ_0 as argument, such that $\int \tilde{Q}\xi_0(x, y)\mathrm{d}x = Q\nu(y)$ and $\int \tilde{Q}\xi_0(x, y)\mathrm{d}y = Q\tilde{\nu}(x)$. This coupling kernel is explicitly defined by the following procedure, which takes as input $(\theta_1, \tilde{\theta}_1) \sim \xi_0$ a coupling of ν and $\tilde{\nu}$, and returns $(\theta_1', \tilde{\theta}_1') \sim \tilde{Q}\xi_0$:

Input: $(\theta_1, \tilde{\theta}_1) \sim \xi_0$, $\xi_1(\cdot \mid \theta_1, \tilde{\theta}_1)$ an optimal coupling between $\pi_{\varepsilon_2}(\cdot \mid \theta_1)$ and $\pi_{\varepsilon_2}(\cdot \mid \tilde{\theta}_1)$, $\xi_2(\cdot \mid \theta_2, \tilde{\theta}_2)$ an optimal coupling between $\pi_{\varepsilon_1}(\cdot \mid \theta_2)$ and $\pi_{\varepsilon_1}(\cdot \mid \tilde{\theta}_2)$.
Output: $(\theta_1', \tilde{\theta}_1') \sim \tilde{Q}\xi_0$.
$(\theta_2, \tilde{\theta}_2) \sim \xi_1(\cdot \mid \theta_1, \tilde{\theta}_1)$;
$(\theta_1', \tilde{\theta}_1') \sim \xi_2(\cdot \mid \theta_2, \tilde{\theta}_2)$.

Algorithm 4.3: Coupling procedure for Theorem 5

This procedure verifies that if $\theta_1 = \tilde{\theta}_1$ then $\theta_1' = \tilde{\theta}_1'$, since for any distribution ν_0, the optimal coupling between ν_0 and itself is $(x, y) \mapsto \nu_0(x)\delta_{x=y}$.

In the proofs we need to choose ξ_0 as the optimal coupling between ν and $\tilde{\nu}$. In the following, $\tilde{\gamma} = \tilde{Q}\xi_0$, so that

$$\begin{aligned}
\|Q\nu - Q\tilde{\nu}\|_{TV} &= \frac{1}{2} \inf_{\gamma \in \Gamma(Q\nu, Q\tilde{\nu})} \mathrm{pr}\{\theta_1' \neq \tilde{\theta}_1' \mid (\theta_1, \tilde{\theta}_1) \sim \gamma\} \\
&\leq \frac{1}{2}\mathrm{pr}\{\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 = \tilde{\theta}_1, (\theta_1', \tilde{\theta}_1') \sim \tilde{\gamma}\}\mathrm{pr}_{\xi_0}(\theta_1 = \tilde{\theta}_1) \\
&\qquad + \frac{1}{2}\mathrm{pr}\{\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1, (\theta_1', \tilde{\theta}_1') \sim \tilde{\gamma}\}\mathrm{pr}_{\xi_0}(\theta_1 \neq \tilde{\theta}_1) \\
&\leq \frac{1}{2}\mathrm{pr}\{\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1, (\theta_1', \tilde{\theta}_1') \sim \tilde{\gamma}\}\mathrm{pr}_{\xi_0}(\theta_1 \neq \tilde{\theta}_1) \\
&\leq \|\nu - \tilde{\nu}\|_{TV}\mathrm{pr}\{\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1, (\theta_1' \neq \tilde{\theta}_1') \sim \tilde{\gamma}\}.
\end{aligned}$$

It is now sufficient to bound $\mathrm{pr}\{\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1, (\theta_1', \tilde{\theta}_1') \sim \tilde{\gamma}\} = 1 - \mathrm{pr}\{\theta_1' = \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1, (\theta_1', \tilde{\theta}_1') \sim \tilde{\gamma}\}$, that is to find a lower bound on the probability that two different values θ_1 and $\tilde{\theta}_1$ transition to the same value.

If $\theta_2 = \tilde{\theta}_2$ then necessarily, $\theta_1' = \tilde{\theta}_1'$, in other words, if the coupling is successful at the first step of the procedure it is sufficient. This means that a lower bound on the coupling probability is the coupling probability at the first step of the procedure. Now,

$$\begin{aligned}
\mathrm{pr}\{\theta_1' = \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1, (\theta_1', \tilde{\theta}_1') \sim \tilde{\gamma}\} &\geq 1 - 2\|\pi_\varepsilon(\cdot \mid \theta_1) - \pi_\varepsilon(\cdot \mid \tilde{\theta}_1)\|_{TV} \\
&\geq 1 - 2\kappa > 0.
\end{aligned}$$

This proves that the map $Q : \nu \mapsto Q\nu$ is a contraction. The space of all measures on $\mathcal{A}$ is complete when endowed with the total variation distance. Furthermore, the subspace of all probability distributions on Θ_1 is stable by Q. Hence, by the Banach fixed-point theorem, it enjoys a fixed point and in particular the sequence $(Q^n\pi)$, with π an arbitrary prior distribution, converges to this fixed point with rate $1 - 2\kappa$. □

The above assumption is satisfied in particular when the parameter space is compact. Possible relaxations are not covered in this paper. This theorem suffers from its generality, as the most practical situation in which the conditions are satisfied is obtained if all the parameters live in a compact space. However we can refine the previous result for many graphical models; such refinements are explored in the next sections.

We can extend the convergence result of Theorem 5 to the general case $n > 2$:

Theorem 6. *Assume that for all* $\ell \leq n$

$$\kappa_\ell = \sup_{\theta_{>\ell}, \tilde{\theta}_{>\ell}} \sup_{\theta_{<\ell}} \|\pi_{\varepsilon_\ell}\{\cdot \mid s_\ell(x^\star, \theta_{<\ell}, \theta_{>\ell})\} - \pi_{\varepsilon_\ell}\{\cdot \mid s_\ell(x^\star, \theta_{<\ell}, \tilde{\theta}_{>\ell})\}\|_{TV} < 1/2$$

with $\theta_{>\ell} = (\theta_{\ell+1}, \theta_{\ell+2}, \ldots, \theta_n)$, *and* $\theta_{<\ell} = (\theta_1, \theta_2, \ldots, \theta_{\ell-1})$. *Then, the Markov chain produced by Algorithm 4.2 converges geometrically in total variation distance to a stationary distribution* ν_ε, *with geometric rate* $1 - \prod_\ell 2\kappa_\ell$.

The proof of this theorem is a straightforward adaptation of the previous proof, with the same coupling procedure. The condition comes from the fact that in this procedure we sequentially try to couple each θ_ℓ using the $\theta_{<\ell}$, already coupled; as a consequence the condition for $\ell = n$ is always satisfied. In the case $n = 2$, we recover Theorem 5.

The limiting distribution ν_ε is not necessarily a standard posterior. We can however provide an evaluation of the distance between ν_ε and the limiting distribution ν_0 of Algorithm 4.2 with $\varepsilon_1 = \varepsilon_2 = 0$. In a compact parameter space, ν_0 always exists, but it may differ from the joint distribution associated with a vanilla ABC sampler, because the conditionals may be based on different summary statistics s_1 and s_2.

Theorem 7. *Assume that*

$$
\begin{aligned}
L_0 &= \sup_{\varepsilon_2} \sup_{\theta_1, \tilde{\theta}_1} \|\pi_{\varepsilon_2}\{\cdot \mid s_2(x^\star, \theta_1)\} - \pi_0\{\cdot \mid s_2(x^\star, \tilde{\theta}_1)\}\|_{TV} < 1/2\,, \\
L_1(\varepsilon_1) &= \sup_{\theta_2} \|\pi_{\varepsilon_1}\{\cdot \mid s_1(x^\star, \theta_2)\} - \pi_0\{\cdot \mid s_1(x^\star, \theta_2)\}\|_{TV} \xrightarrow[\varepsilon_1 \to 0]{} 0\,, \\
L_2(\varepsilon_2) &= \sup_{\theta_1} \|\pi_{\varepsilon_2}\{\cdot \mid s_2(x^\star, \theta_1)\} - \pi_0\{\cdot \mid s_2(x^\star, \theta_1)\}\|_{TV} \xrightarrow[\varepsilon_2 \to 0]{} 0\,.
\end{aligned}
$$

Then

$$
\|\nu_\varepsilon - \nu_0\|_{TV} \leq \frac{L_1(\varepsilon_1) + L_2(\varepsilon_2)}{1 - 2L_0} \xrightarrow[\varepsilon \to 0]{} 0.
$$

Proof. The assumptions on L_2 and L_0 imply with the triangular inequality that the assumptions of Theorem 5 are verified, and thus that μ_ε exists.

In this proof, we need a coupling between two chains with different transition kernels. Let ν_ε be the target distribution of the approximate Gibbs sampler and ν_0 be the target distribution of the exact Gibbs sampler. Let $(\theta_1, \tilde{\theta}_1)$ be a realisation of an optimal coupling ξ_0 between ν_ε and ν_0. As before we propose a coupling procedure:

As the distributions ν_ε and ν_0 are stationary for the evolution process, we have

$$
\begin{aligned}
\mathrm{pr}(\theta_1' \neq \tilde{\theta}_1') =& \mathrm{pr}(\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1)\mathrm{pr}(\theta_1 \neq \tilde{\theta}_1) + \mathrm{pr}(\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 = \tilde{\theta}_1)\mathrm{pr}(\theta_1' = \tilde{\theta}_1') \\
\leq& \frac{\mathrm{pr}(\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 = \tilde{\theta}_1)}{\mathrm{pr}(\theta_1' = \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1)}.
\end{aligned}
$$

Input: $(\theta_1, \tilde{\theta}_1) \sim \xi_0$, $\xi_3(\cdot \mid \theta_1, \tilde{\theta}_1)$ an optimal coupling between $\pi_\varepsilon(\cdot \mid \theta_1)$ and $\pi(\cdot \mid \tilde{\theta}_1)$, $\xi_4(\cdot \mid \theta_2, \tilde{\theta}_2)$ an optimal coupling between $\pi_\eta(\cdot \mid \theta_2)$ and $\pi(\cdot \mid \tilde{\theta}_2)$.
Output: $(\theta_1', \tilde{\theta}_1') \sim \tilde{Q}\xi_0$.
$(\theta_2, \tilde{\theta}_2) \sim \xi_3(\cdot \mid \theta_1, \tilde{\theta}_1)$;
$(\theta_1', \tilde{\theta}_1') \sim \xi_4(\cdot \mid \theta_2, \tilde{\theta}_2)$.

Algorithm 4.4: Coupling procedure for Theorem 7

As before we use a rough bound on the denominator:

$$\begin{aligned}\mathrm{pr}(\theta_1' = \tilde{\theta}_1' \mid \theta_1 \neq \tilde{\theta}_1) \geq & (1 - 2 \sup_{\varepsilon} \sup_{\theta_1, \tilde{\theta}_1} \|\pi_\varepsilon(\cdot \mid \theta_1) - \pi(\cdot \mid \tilde{\theta}_1)\|_{TV}) \\ \geq & 1 - 2L_0.\end{aligned}$$

For the numerator, we have, with θ_2 and $\tilde{\theta}_2$ the transitory values of the second parameter,

$$\begin{aligned}\mathrm{pr}(\theta_1' \neq \tilde{\theta}_1' \mid \theta_1 = \tilde{\theta}_1) \leq & \mathrm{pr}(\theta_1' \neq \tilde{\theta}_1' \mid \theta_2 = \tilde{\theta}_2)\mathrm{pr}(\theta_2 \neq \tilde{\theta}_2 \mid \theta_1 = \tilde{\theta}_1) \\ & + \mathrm{pr}(\theta_1' \neq \tilde{\theta}_1' \mid \theta_2 \neq \tilde{\theta}_2)\mathrm{pr}(\theta_2 \neq \tilde{\theta}_2 \mid \theta_1 = \tilde{\theta}_1) \\ \leq & \sup_{\theta_2} \mathrm{pr}\{\theta_1 \neq \tilde{\theta}_1 \mid (\theta_1, \tilde{\theta}_1) \sim \xi_4(\cdot \mid \theta_2, \theta_2)\} \\ & + \sup_{\vartheta_1} \mathrm{pr}\{\theta_2 \neq \tilde{\theta}_2 \mid (\theta_2, \tilde{\theta}_2) \sim \xi_3(\cdot \mid \vartheta_1, \vartheta_1)\} \\ \leq & 2L_1(\varepsilon_1) + 2L_2(\varepsilon_2)\end{aligned}$$

Putting together both estimates gives the bound of the theorem.

□

20.4 Counter-example to Theorem 5

In this section, we give a simple example where the assumptions of Theorem 5 are not verified and where ABC-Gibbs fails (whereas Vanilla ABC does not).

Take a single observation from a mixture of two uniforms, with parametrised by (θ_1, θ_2):

$$x \sim \frac{1}{2}\mathcal{U}(\theta_1, \theta_1 + 1) + \frac{1}{2}\mathcal{U}(\theta_2, \theta_2 + 1).$$

For the numerical applications, we shall use the realization $x^\star = 5$. Consider the prior distribution

$$(\theta_1, \theta_2) \sim \mathcal{U}(\mathcal{A}) \qquad \mathcal{A} = \{(\theta_1, \theta_2) : 0 \leq \theta_1, \theta_2 \leq 10 \text{ and } |\theta_1 - \theta_2| > 2\}$$

The exact posterior is uniform over the set

$$(([0,10]\times[x-1,x])\cup([x-1,x]\times[0,10]))\cap\mathcal{A}.$$

The prior and exact posterior are shown in Figure 4.1, as well as the outcome of Vanilla ABC and ABC-Gibbs with $\varepsilon=\varepsilon_1=\varepsilon_2=0.5$. Vanilla ABC leads to a reasonable approximation of the posterior, but ABC-Gibbs misses half of the posterior. Other realizations of ABC-Gibbs lead to the symmetric pseudo-posterior, with the roles of θ_1 and θ_2 swapped. This is a situation where the ABC-Gibbs does not converge to a unique stationary distribution ν_ε (as soon as $\varepsilon_1,\varepsilon_2\leq\frac{1}{2}$).

For Theorem 5 to apply, we would need

$$\sup_{\theta_1,\tilde{\theta}_1}\|\pi_{\varepsilon_2}\{\cdot\mid s_2(x^\star,\theta_1)\}-\pi_{\varepsilon_2}\{\cdot\mid s_2(x^\star,\tilde{\theta}_1)\}\|_{TV}=\kappa<\frac{1}{2}.$$

Consider $\theta_1=1$ and $\tilde{\theta}_1=5$. Then $\pi_{\varepsilon_2}\{\cdot\mid s_2(x^\star,\theta_1)\}$ has support $[3.5,5.5]$ and $\pi_{\varepsilon_2}\{\cdot\mid s_2(x^\star,\tilde{\theta}_1)\}$ has support $[0,3]\cup[7,10]$. Since the two supports are disjoint, the distance in total variation between the two distributions is 1, and Theorem 5 does not apply. Intuitively, the Markov chain does not converge because it is not irreducible.

20.5 Generalities on total variation distance

The main tool in our proofs is the total variation distance used by Nummelin [1978] and Meyn and Tweedie [1993]. Let ν and $\tilde{\nu}$ be two probability distributions over the same space E. A coupling γ between ν and $\tilde{\nu}$ is a probability distribution on $E\times E$ such that $\int\gamma(x,y)\mathrm{d}x=\nu$ and $\int\gamma(x,y)\mathrm{d}y=\tilde{\nu}$. Let $\Gamma(\nu,\tilde{\nu})$ denote the set of all couplings between ν and $\tilde{\nu}$. Then the total variation distance is defined as

$$\|\nu-\tilde{\nu}\|_{TV}=\frac{1}{2}\inf_{\gamma\in\Gamma(\nu,\tilde{\nu})}\mathrm{pr}\{x\neq y\mid(x,y)\sim\gamma\}.$$

To handle this distance, we build an explicit coupling between the distributions: this provides an upper bound on the total variation distance. Note that there always exists an optimal coupling between two distributions, that is a coupling γ_0 such that $\|\nu-\tilde{\nu}\|_{TV}=\frac{1}{2}\mathrm{pr}\{x\neq y\mid(x,y)\sim\gamma_0\}$.

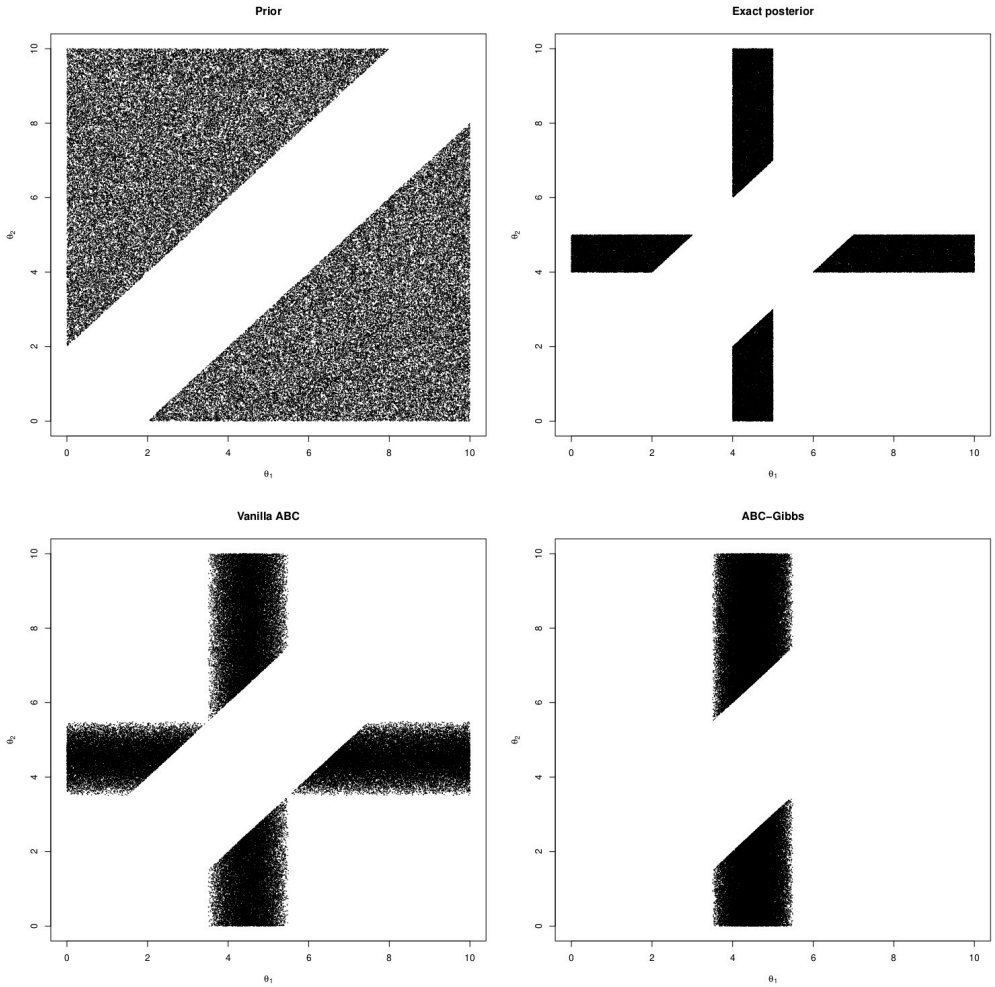


FIGURE 4.1 – Illustration of the mixture of uniforms counter-example from Section 20.4, with $x^\star = 5$ and $\varepsilon = 0.5$. Top left: prior distribution. Top right: Exact posterior. Bottom left: Vanilla ABC posterior. Bottom right: one possible outcome of ABC-Gibbs. The Vanilla ABC is a reasonable approximation of the exact posterior, but the ABC-Gibbs outcome only covers half of the support.

21 Component-wise approximate Bayesian computation: the hierarchical case

21.1 Algorithm and theory

In this section, we focus on the two-stage simple hierarchical model given in (2). This model appears naturally when a hierarchical structure is added to a non-tractable model, see for example Turner and Van Zandt [2013]. Under this model structure, the conditional distributions greatly simplify as

$$\pi(\mu_j \mid x^\star, \alpha, \mu_1, \ldots, \mu_{j-1}, \mu_{j+1}, \ldots, \mu_n) = \pi(\mu_j \mid x_j^\star, \alpha) \text{ and } \pi(\alpha \mid \mu, x^\star) = \pi(\alpha \mid \mu).$$

Algorithm 4.2 then further simplifies and a detailed version in this particular situation is given in Algorithm 4.5. In order to simulate from all or part of the approximate conditional distributions, we might resort to a Metropolis step, using the prior distribution as proposal.

Input: observed dataset $x^\star$, number of iterations N, starting points $\alpha^{(0)}$ and $\mu^{(0)} = (\mu_1^{(0)}, \ldots, \mu_n^{(0)})$, thresholds ε_α and ε_μ, summary statistics s_α and s_μ, and distances d_α and d_μ.
Output: A sample $(\alpha^{(i)}, \mu^{(i)})_{1 \leq i \leq N}$.
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, n$ **do**
    Sample $\mu_j^c \sim \pi(\mu \mid \alpha^{(i-1)})$ and $x_j^c \sim f(\cdot \mid \mu_j^c)$
    while $d\{s_\mu(x_j^c), s_\mu(x_j^\star)\} > \varepsilon_\mu$ **do**
      Sample $\mu_j^c \sim \pi(\mu \mid \alpha^{(i-1)})$ and $x_j^c \sim f(x_j \mid \mu_j^c)$
    $\mu_j^{(i)} \leftarrow \mu_j^c$; // thus $\mu_j^{(i)} \sim \pi_{\varepsilon_\mu}\{\cdot \mid s_\mu(x_j^\star, \alpha^{(i-1)})\}$
  Sample $\alpha^c \sim \pi(\alpha)$ and $\mu^c \sim \pi(\cdot \mid \alpha^c)$,
  while $d\{s_\alpha(\mu^c), s_\alpha(\mu^{(i)})\} > \varepsilon_\alpha$ **do**
    Sample $\alpha^c \sim \pi(\alpha)$ and $\mu^c \sim \pi(\cdot \mid \alpha^c)$,
  $\alpha^{(i)} \leftarrow \alpha^c$; // thus $\alpha^{(i)} \sim \pi_{\varepsilon_\alpha}\{\cdot \mid s_\alpha(\mu^{(i)})\}$

Algorithm 4.5: ABC-Gibbs sampler for hierarchical model (2)

As in Algorithm 4.2, Algorithm 4.5 may bypass the approximation of some conditionals. In particular, if $\pi(\alpha \mid \mu)$ can be simulated from and $\pi(\mu \mid x^\star, \alpha)$ cannot, we prove in the Supplementary Material, Section 21.4, that the limiting distribution of our algorithm is the same as the vanilla Approximate Bayesian computation algorithm. On the other hand, if we can simulate from $\pi(\mu \mid x^\star, \alpha)$ and not from $\pi(\alpha \mid \mu)$, a version of Theorem 6 (Theorem 8) is established under less stringent conditions in the Supplementary Material, Section 21.4.

21.2 Numerical comparison with vanilla ABC

We now compare the ABC-Gibbs, Vanilla ABC and an implementation of the SMC-ABC algorithm (approximate Bayesian computation with sequential Monte Carlo) of Del Moral et al. [2012], with an adaptive proposal and resampling steps, following Toni et al. [2008] in order to avoid degeneracy in the simulation. The example is the toy Normal–Normal model from Gelman et al. [2013]:

$$\mu_j \overset{\text{iid}}{\sim} \mathcal{N}(\alpha, \varsigma^2)\,, \quad x_{j,k} \overset{\text{ind}}{\sim} \mathcal{N}(\mu_j, \sigma^2)\,, \qquad j = 1, \ldots, n, \quad k = 1, \ldots, K \tag{3}$$

with the variances σ^2 and ς^2 known, and a hyperprior $\alpha \sim \mathcal{U}[-4, 4]$. This model is not intractable, which allows us to compare the output with the true posterior in Figure 4.3.

We can check that the assumptions of Theorem 7 apply to the model. We define $\mu_{-i} = (\mu_1, \ldots, \mu_{i-1}, \mu_{i+1}, \ldots, \mu_n)$. By conditional independence of the μ_i given α, and choice of s_μ, we have:

$$\pi_{\varepsilon_\mu}(\cdot \mid s_\mu\{x^\star, \alpha, \mu_{-i}\}) = \pi_{\varepsilon_\mu}\{\cdot \mid s_\mu(x_i^\star, \alpha)\}$$

The assumptions to check can be rewritten as:

(μ_1) $\sup_{\alpha,\tilde{\alpha}} \|\pi_{\varepsilon_\mu}(\cdot \mid s_\mu(x_1^\star), \alpha) - \pi_{\varepsilon_\mu}(\cdot \mid s_\mu(x_1^\star), \tilde{\alpha})\|_{TV} < 1/2$

$\vdots$

(μ_n) $\sup_{\alpha,\tilde{\alpha}} \|\pi_{\varepsilon_\mu}(\cdot \mid s_\mu(x_n^\star), \alpha) - \pi_{\varepsilon_\mu}(\cdot \mid s_\mu(x_n^\star), \tilde{\alpha})\|_{TV} < 1/2$

(α) $\sup_{\mu} \|\pi_{\varepsilon_\alpha}(\cdot \mid s_\alpha(\mu)) - \pi_{\varepsilon_\alpha}\{\cdot \mid s_\alpha(\mu)\}\|_{TV} < 1/2$

To prove the assumption (μ_i), we underline the fact that it is sufficient to check that there exists some subset K of the parameter space, with positive measure for all hyperparameter α, such that $\exists C > 0, \forall \alpha, \forall \mu \in K$, $\pi_{\varepsilon_\mu}(\mu \mid \alpha, s(x^*)) > C$.

We can compute these densities:

$$\pi_\varepsilon(\mu \mid \alpha, s(x^*)) = \frac{\exp(-(\mu-\alpha)^2/(2\tau) \int \exp(-(y-\mu)^2\sqrt{n}/(2\sigma)\mathbf{1}_{|y-\bar{x^\star}|<\varepsilon} \mathrm{d}y}{\int \exp(-(\mu-\alpha)^2/(2\tau) \exp(-(y-\mu)^2\sqrt{n}/(2\sigma)\mathbf{1}_{|y-\bar{x^\star}|<\varepsilon} \mathrm{d}y \mathrm{d}\mu}.$$

As α is compactly supported on $[-4, 4]$, the conditions are verified: we can roughly bound the probabilities by continuity of the expression.

The last condition (α) is always verified as we have by definition of the total variation distance:

$$\sup_{\mu} \|\pi_{\varepsilon_\alpha}(\cdot \mid \mu) - \pi_{\varepsilon_\alpha}(\cdot \mid \mu)\|_{TV} = 0.$$

Recall that in practice the tolerance is provided by an empirical quantile of the distance distribution at each call of an approximate conditional. This means that at each iteration N_α and N_μ simulations are produced from the conditional prior predictives on α and μ, respectively, and that only the simulation associated with the smallest distance is kept. The `R` code used for all simulations can be found at `https://github.com/GClarte/ABCG`.

We strive to provide a fair comparison between ABC-Gibbs and vanilla ABC and hence aim at simulating overall the same number of normal random variables. In ABC, simulating over the hierarchical structures involves $n + nK$ normal variates; taking the best N out of N_V prior predictive simulations thus costs $N_{\text{tot}} = N_V n(1 + K)$. In ABC-Gibbs, each iteration costs $N_\alpha n + N_\mu nK$; if we take $N_\alpha = N_\mu$ the total cost is $N_{\text{tot}} = NnN_\alpha(1 + K)$. We thus take $N = N_V/N_\alpha$ to compare both algorithms.

Figure 4.2 illustrates the result of both algorithms, for $\sigma = 1$, $K = 10$, $n = 20$, by representing the posterior approximation from ABC-Gibbs and Vanilla ABC for the hyperparameter α and the first three parameters, with comparable computational costs. The statistic used at both parameter and hyperparameter levels is the corresponding empirical mean and hence is sufficient. We keep N constant and increase $N_\alpha = N_\mu$.

This toy experiment exhibits a considerable improvement in the parameter estimator when using ABC-Gibbs. This is easily explained by the difficulty for ABC to find a suitable value of $\mu \in \mathbb{R}^{20}$; poor estimation of the parameter ensues. In fact, ABC produces the same output as a non-hierachical model when the μ_j's are integrated out.

This figure exhibits that ABC-Gibbs scales more efficiently with N_μ, that is with the reduction of ε_μ, especially when increasing N_α from 5 to 30, that is a mere 6 time increase in the computational cost. For the same variation in ABC, we do see no noticeable improvement. Hence, for a given computational cost, ABC-Gibbs achieves a smaller threshold ε than ABC, leading to better approximations. The experiment further points out that the choice of the parameters N_α and N_μ may prove delicate. Resorting to a larger ABC table for each update is uselessly costly in that it fails to provide a clear improvement in the result. This is also the case with the classical ABC approach, as shown by Figure 4.2.

In practice, the choice of the parameters N and N_μ may be tricky. For N we would advise to use standard techniques to choose the number of iterations in a Monte Carlo algorithm. For N_μ and N_α, we observe in Figure 4.2 that a moderate value, say $N_\alpha = N_\mu = 30$, seems enough: we expect the optimal value to be problem dependent.

To check the robustness of our method, we represent in Figure 4.3, 10 realisa-

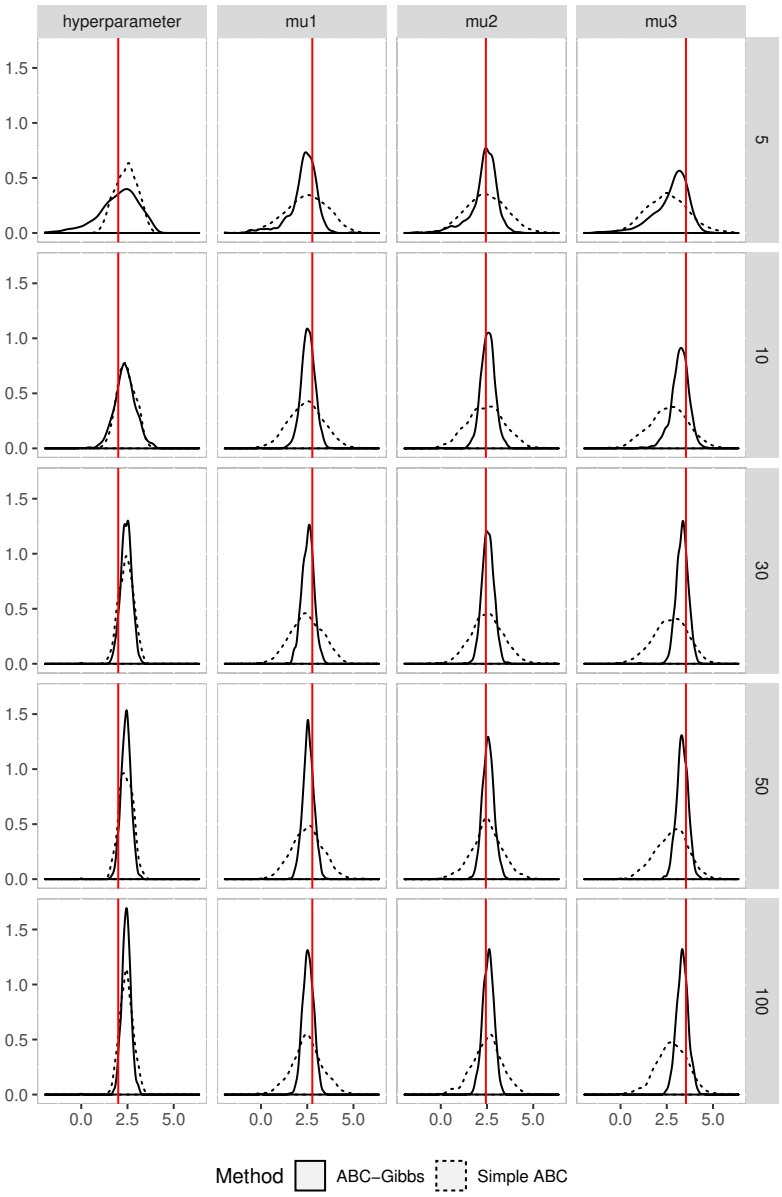


FIGURE 4.2 – Comparison of the posterior density estimates of the hyperparameter and the first three parameter components of the hierarchical model of Equation 3 obtained with ABC and ABC-Gibbs, with identical computational cost. For ABC-Gibbs, results were computed with $N = 1000$ Gibbs iterations; each row is labeled with the number $N_\alpha = N_\mu$ of iterations of the ABC scheme to update one parameter component. Red vertical lines represents the true values used to simulate the data.

tions of the posterior densities, for $N = \lfloor 1000/30 \rfloor$, and $N_\alpha = N_\mu = 30$ (with the first 5 points in ABC-Gibbs removed to account for the small burn in). SMC-ABC does not allow for a fixed limit on the number of simulations, due to the resampling step. We used therefore 10^4 particles, with a target of the smallest possible tolerance for a maximum of 30 steps. In total, SMC-ABC was alloted roughly 60 times more simulations than ABC-Gibbs and ABC. The ABC-Gibbs density is overdispersed compared to the true posterior, albeit closer than the ABC, especially for the parameter μ_1. On the other hand, SMC-ABC fails for this model: due to the difficulties resulting from its high dimension, an adaptive version fails to produce interesting proposals, notwithstanding a consistently larger computational budget. The distribution approximation on α is however better than its ABC counterpart. This fact is supported by numerical experiments in lower dimensions where all three methods lead to suitable approximations, as illustrated in the supplementary material, Section 21.3. The improvement brought by ABC-Gibbs in high dimension occurs consistently over simulations.

This experiment further highlights a striking differenciation between ABC-SMC methods, which require a significant degree of calibration when no package is readily available, and ABC-Gibbs, which relies on a straightforward implementation.

21.3 Comparison in dimension 3

In addition to the results shown in Figure 4.3, we show in Figure 4.4 a comparison of ABC-Gibbs, Vanilla ABC and SMC-ABC for the toy model of Section 21.2 in the low dimension case $n = 2$.

As expected, in this low-dimension setting the results from SMC-ABC and Vanilla ABC are comparable to the approximate posterior provided by ABC-Gibbs for the parameter. ABC-Gibbs however seems to lead to a less stable approximation of the hyperparameter, this can be explained by the lower number of points (as we removed some of the first points as burn-in). This supports the idea that the behaviour of SMC-ABC in Figure 4.3 is caused by the high dimensionality. We believe that in higher dimension, SMC-ABC would require a very large number of particles, and a higher number of iteration each of which would cost much more in resampling, leading to a disastrous computational cost.

21.4 Proofs specific to the hierarchical case

In the hierarchical cases, it is possible to derivate similar results with more practical conditions. They are based on a particular implementation of ABC-Gibbs, presented for $n = 1$ and in the case of an analytically available conditional density

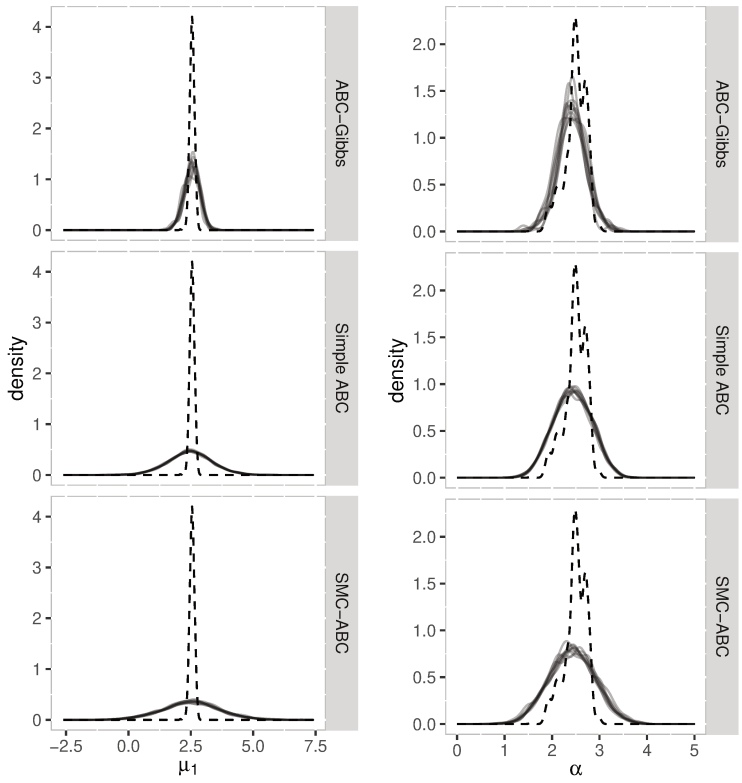


FIGURE 4.3 – Posterior densities for 10 replicas of the algorithms compared to the exact posterior density. The true posterior is represented by the dashed line.

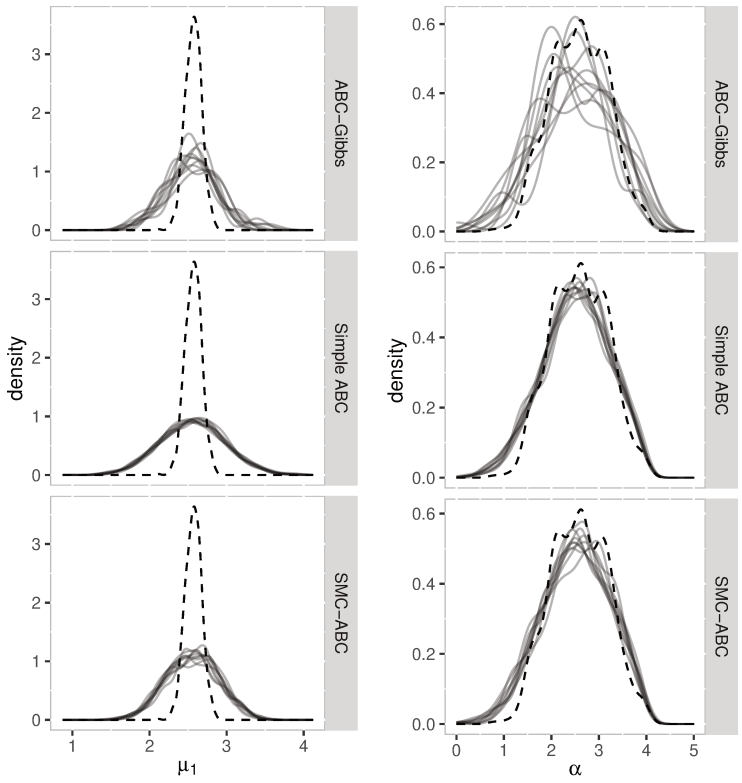


FIGURE 4.4 – Parameter and hyperparameter for the Normal-Normal model with only two parameter and one hyperparameter.

$\pi(\mu \mid \alpha, x^\star)$, in Algorithm 4.6. We will gradually weaken the assumptions to finally prove Theorem 8:

We define $\mathcal{A}$ as the domain of α and $\mathcal{B}$ as the domain of μ.

Theorem 8. *Assume there exists a non-empty convex set C with positive prior measure such that*

$$\begin{aligned}
\kappa_1 &= \inf_{s_\alpha(\mu)\in C} \pi(B_{s_\alpha(\mu),\varepsilon_\alpha/4}) > 0\,,\\
\kappa_2 &= \inf_{\alpha} \inf_{s_\alpha(\mu)\in C} \pi_{\varepsilon_\mu}(B_{s_\alpha(\mu),3\varepsilon_\alpha/2} \mid s_\mu(x^\star,\alpha),\alpha) > 0\,,\\
\kappa_3 &= \inf_{\alpha} \pi_{\varepsilon_\mu}\{s_\alpha(\mu)\in C \mid s_\mu(x^\star,\alpha)\} > 0\,,
\end{aligned}$$

where $B_{z,h}$ denotes the ball of center z and radius h. Then the Markov chain produced by Algorithm 4.5 converges geometrically in total variation distance to a stationary distribution ν_ε, with geometric rate $1-\kappa_1\kappa_2\kappa_3^2$.

The rate in Theorem 8 is uninformative, as it is specific to the selected implementation.

Input: $\alpha^{(0)} \sim \pi(\alpha)$, $\mu^{(0)} \sim \pi(\mu \mid \alpha^{(0)})$.
Output: A sample $(\alpha^{(i)}, \mu^{(i)})_{1\le i\le N}$.
for $i = 1, \ldots, N$ **do**
  $\mu^c \sim \pi(\cdot \mid \alpha^{(i-1)}, x^\star)$
  $\alpha^c \sim \pi$
  $\tilde{\mu} \sim \pi(\cdot \mid \alpha^c)$
  if $\eta\{s_\alpha(\mu), s_\alpha(\mu^c)\} < \varepsilon_\alpha$ **then**
    $\mu^{(i+1)} \leftarrow \mu^c$
    $\alpha^{(i+1)} \leftarrow \alpha^c$
  else
    $\mu^{(i+1)} \leftarrow \mu^{(i)}$
    $\alpha^{(i+1)} \leftarrow \alpha^{(i)}$

Algorithm 4.6: Implementation of ABC-Gibbs used in the proofs.

First we state the most restrictive result:

Theorem 9. *Assume that the following conditions are both satisfied:*

$$\begin{aligned}
\kappa_1 &= \inf_{\mu} \pi(B_{s_\alpha(\mu),\varepsilon_\alpha/4}) > 0\\
\kappa_2 &= \inf_{\alpha}\inf_{\mu} \pi(B_{s_\alpha(\mu),3\varepsilon_\alpha/2} \mid \alpha, x) > 0.
\end{aligned}$$

Then, the Markov chain associated with Algorithm 4.6 enjoys an invariant distribution, and it converges geometrically to this invariant measure with rate $1-\kappa_1\kappa_2$ for the total variation distance.

Proof. The technique of the proof is essentially similar to that of Theorem 5. Let ν and $\tilde{\nu}$ be two distributions over $\mathcal{A}$. We describe the evolution of $\alpha, \tilde{\alpha}$ into $\alpha', \tilde{\alpha}'$, though the kernel $\tilde{Q}$. We denote $\mu, \tilde{\mu}$ the transitory second parameter.

Input: $(\alpha, \tilde{\alpha})$.
Output: $(\alpha', \tilde{\alpha}')$.
if $\alpha \neq \tilde{\alpha}$ **then**
  $(\mu, \tilde{\mu}) \sim \pi(\cdot \mid \alpha, x) \otimes \pi(\cdot \mid \tilde{\alpha}, x)$
else
  $\mu = \tilde{\mu} \sim \pi(\mu \mid \alpha, x)$
$\alpha^c \sim \pi$
$\mu^c \sim \pi(\cdot \mid \alpha^c)$;
if $\eta\{s_\alpha(\mu), s_\alpha(\mu^c)\} \leq \varepsilon_\alpha$ **then**
  $\alpha' \leftarrow \alpha^c$
else
  $\alpha' \leftarrow \alpha$
if $\eta\{s_\alpha(\tilde{\mu}), s_\alpha(\mu^c)\} \leq \varepsilon_\alpha$ **then**
  $\tilde{\alpha}' \leftarrow \alpha^c$
else
  $\tilde{\alpha}' \leftarrow \tilde{\alpha}$

Algorithm 4.7: Coupling procedure

This process defines a transition kernel $\tilde{Q}$ for two coupled chains. As in the previous proofs, if $\alpha = \tilde{\alpha}$ then $\alpha' = \tilde{\alpha}'$.

Let $(\alpha, \tilde{\alpha}) \sim \xi$, an optimal coupling between ν and $\tilde{\nu}$. Then,

$$\begin{aligned}
\|Q\nu - Q\tilde{\nu}\|_{TV} &= \frac{1}{2} \inf_{\gamma \in \Gamma(Q\nu, Q\tilde{\nu})} \mathrm{pr}\{\alpha' \neq \tilde{\alpha}' \mid (\alpha', \tilde{\alpha}') \sim \gamma\} \\
&\leq \frac{1}{2} \mathrm{pr}_\xi\{\alpha' \neq \tilde{\alpha}' \mid \alpha = \tilde{\alpha}, (\alpha', \tilde{\alpha}') \sim \tilde{Q}\xi\} \mathrm{pr}_\xi(\alpha = \tilde{\alpha}) \\
&\quad + \frac{1}{2} \mathrm{pr}_\xi\{\alpha' \neq \tilde{\alpha}' \mid \alpha \neq \tilde{\alpha}, (\alpha', \tilde{\alpha}') \sim \tilde{Q}\xi\} \mathrm{pr}_\xi(\alpha \neq \tilde{\alpha}) \\
&\leq \frac{1}{2} \mathrm{pr}_\xi(\alpha' \neq \tilde{\alpha}' \mid \alpha \neq \tilde{\alpha}, (\alpha', \tilde{\alpha}') \sim \tilde{Q}\xi) \mathrm{pr}_\xi(\alpha \neq \tilde{\alpha}) \\
&\leq \|\nu - \tilde{\nu}\|_{TV} \mathrm{pr}_\xi(\alpha' \neq \tilde{\alpha}' \mid \alpha \neq \tilde{\alpha}, (\alpha', \tilde{\alpha}') \sim \tilde{Q}\xi).
\end{aligned}$$

It is sufficient to find a uniform upper bound on $\mathrm{pr}_{\nu,\tilde{\nu}}(\alpha' \neq \tilde{\alpha}' \mid \alpha \neq \tilde{\alpha}, (\alpha', \tilde{\alpha}') \sim \tilde{Q}\xi) = \int \pi(s_\alpha(\mu^c) \notin B_{s_\alpha(\mu),\varepsilon_\alpha} \cap B_{s_\alpha(\tilde{\mu}),\varepsilon_\alpha}) \pi(\mu \mid \alpha, x) \pi(\tilde{\mu} \mid \tilde{\alpha}, x) \nu(\alpha) \tilde{\nu}(\tilde{\alpha}) \, \mathrm{d}\tilde{\alpha} \mathrm{d}\mu \mathrm{d}\tilde{\mu}$. Notice that we can choose our coupling ξ such that conditionally on $\alpha \neq \tilde{\alpha}$ the

marginals are independent.

$$\begin{aligned}
\mathrm{pr}_{\nu,\tilde{\nu}}(\alpha' \neq \tilde{\alpha}' \mid \alpha \neq \tilde{\alpha}) = &\int \pi\left\{(B_{s_\alpha(\mu),\varepsilon_\alpha} \cap B_{s_\alpha(\tilde{\mu}),\varepsilon_\alpha})^c\right\} \pi(\mu \mid \alpha, x) \\
&\quad \times \pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\tilde{\nu}(\tilde{\alpha})\,\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
= &\int \pi\left\{(B_{s_\alpha(\mu),\varepsilon_\alpha} \cap B_{s_\alpha(\tilde{\mu}),\varepsilon_\alpha})^c\right\} \pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x) \\
&\quad \times \nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}\leq 3\varepsilon_\alpha/2\}}\,\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
&+ \int \pi\left\{(B_{s_\alpha(\mu),\varepsilon_\alpha} \cap B_{s_\alpha(\tilde{\mu}),\varepsilon_\alpha})^c\right\} \pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x) \\
&\quad \times \nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\alpha/2\}}\,\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
= &I_1 + I_2.
\end{aligned}$$

We now bound I_1 and I_2.

$$\begin{aligned}
I_1 = &\int \pi\left\{(B_{s_\alpha(\mu),\varepsilon_\alpha} \cap B_{s_\alpha(\tilde{\mu}),\varepsilon_\alpha})^c\right\} \pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x) \\
&\quad \nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}\leq 3\varepsilon_\alpha/2\}}\mathrm{d}\tilde{\alpha}\mathrm{d}\alpha\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\leq &\int \pi(B^c_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/4})\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}\leq 3\varepsilon_\alpha/2\}} \\
&\quad \nu(\tilde{\alpha})\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\leq &\mathrm{pr}_{\nu,\tilde{\nu}}\{\eta\{s_\alpha(\mu), s_\alpha(\tilde{\mu})\} \leq 3\varepsilon_\alpha/2\} \\
&\quad - \int \pi(B_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/2})\pi(s_\alpha(\tilde{\mu}) \in B_{s_\alpha(\mu),3\varepsilon_\alpha/2} \mid \tilde{\alpha}, x)\pi(\mu \mid \alpha, x) \\
&\quad\quad \nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\leq &\mathrm{pr}_{\nu,\tilde{\nu}}\{\eta\{s_\alpha(\mu), s_\alpha(\tilde{\mu})\} \leq 3\eta/2\} \\
&\quad - \kappa_1 \int \pi(s_\alpha(\tilde{\mu}) \in B_{s_\alpha(\mu),3\varepsilon_\alpha/2} \mid \tilde{\alpha}, x)\pi(\mu \mid \alpha, x)\nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu \\
\leq &\mathrm{pr}_{\nu,\tilde{\nu}}\{\eta\{s_\alpha(\mu), s_\alpha(\tilde{\mu})\} \leq 3\varepsilon_\alpha/2\} - \kappa_1\kappa_2
\end{aligned}$$

$$\begin{aligned}
I_2 = &\int \pi\left\{(B_{s_\alpha(\mu),\varepsilon_\alpha} \cap B_{s_\alpha(\tilde{\mu}),\varepsilon_\alpha})^c\right\} \pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x) \\
&\quad \nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\alpha/2\}}\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\leq &\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\mathbf{1}_{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\eta/2}\tilde{\nu}(\tilde{\alpha})\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\leq &\mathrm{pr}_{\nu,\tilde{\nu}}\{\eta\{s_\alpha(\mu), s_\alpha(\tilde{\mu})\} > 3\varepsilon_\eta/2\}
\end{aligned}$$

Finally, putting both inequalities together, we have $I_1 + I_2 \leq 1 - \kappa_1\kappa_2$, with $\kappa_1\kappa_2 > 0$, and

$$\|Q\nu - Q\tilde{\nu}\|_{TV} \leq (1 - \kappa_1\kappa_2)\|\nu - \tilde{\nu}\|_{TV}.$$

The conclusion is the same as in the proof of Theorem 5.

□

Remark 1. *In the proof, when we describe the coupling kernel, we generate μ and $\tilde{\mu}$ independently if α and $\tilde{\alpha}$ are different, and as a single μ if they are equal. This is a particular coupling of the distributions $\pi(\cdot \mid \alpha, x)$ and $\pi(\cdot \mid \tilde{\alpha}, x)$. Here, the link between Theorem 5 and this one becomes clear, as we make the coupling explicit toward reaching a bound in total variation.*

We now relax the assumptions. First, we remove the assumption that $\mathcal{A}$ is compact: the resulting theorem is Theorem 8.

Proof. With the same notations as before, we merely need to find a lower bound:

$$
\begin{aligned}
I_3 = \int & \operatorname{pr}(\mu^c \in B_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/2})\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\mu(\alpha)\nu(\tilde{\alpha}) \\
& \mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\alpha/2\}}\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu}
\end{aligned}
$$

$$
\begin{aligned}
I_3 \geq & \int \operatorname{pr}(B_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/2})\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\tilde{\nu}(\tilde{\alpha}) \\
& \qquad \times \mathbf{1}_{\{s_\alpha(\mu)+s_\alpha(\tilde{\mu})\}/2\in C\}}\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\alpha/2\}}\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
& + \int \operatorname{pr}(\mu \in B_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/2})\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\tilde{\nu}(\tilde{\alpha}) \\
& \qquad \times \mathbf{1}_{\{s_\alpha(\mu)+s_\alpha(\tilde{\mu})\}/2\notin C\}}\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\alpha/2\}}\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\geq & \int \operatorname{pr}(\mu \in B_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/2})\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\tilde{\nu}(\tilde{\alpha}) \\
& \qquad \times \mathbf{1}_{\{s_\alpha(\mu)\in C\}}\mathbf{1}_{\{s_\alpha(\tilde{\mu})\in C\}}\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\{>3\varepsilon_\alpha/2\}}\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\
\geq & \kappa_1\kappa_2\kappa_3^2
\end{aligned}
$$

as the convexity of C ensures that $\mathbf{1}_{\{\{s_\alpha(\mu)+s_\alpha(\tilde{\mu})\}/2\in C\}} \geq \mathbf{1}_{s_\alpha(\mu)\in C}\mathbf{1}_{s_\alpha(\tilde{\mu})\in C}$. □

We can remove the assumption that $\mathcal{B}$ is compact, by imposing a different assumption:

Theorem 10. *Assume that there exist $\mathcal{H} \subset \mathcal{P}(\mathcal{A})$ stable by Q and $A \subset \mathcal{A}$ and $C \subset s_\alpha(\mathcal{B})$ with finite positive measure such that:*

$$
\begin{aligned}
\kappa_1 &= \inf_{s_\alpha(\mu)\in C} \pi(B_{s_\alpha(\mu),\varepsilon_\alpha/4}) > 0; \\
\kappa_2 &= \inf_{\alpha\in A}\inf_{s_\alpha(\mu)\in C} \pi(B_{s_\alpha(\mu),3\varepsilon_\alpha/2} \mid \alpha, x) > 0; \\
\kappa_3 &= \inf_{\alpha\in A} \pi(s_\alpha(\mu) \in C \mid \alpha, x) > 0; \\
\kappa_4 &= \inf_{\nu\in\mathcal{H}} \nu(A) > 0.
\end{aligned}
$$

Then, the Markov chain associated with Algorithm 4.6 enjoys an invariant distribution, and it converges geometrically to this invariant measure with rate $1 - \kappa_1\kappa_2\kappa_3^2\kappa_4^2$.

Proof. Similarly to previous proofs, we have

$$\begin{aligned} I_3 &\geq \int \mathrm{pr}(\mu \in B_{\frac{s_\alpha(\mu)+s_\alpha(\tilde{\mu})}{2},\varepsilon_\alpha/2})\pi(\mu \mid \alpha, x)\pi(\tilde{\mu} \mid \tilde{\alpha}, x)\nu(\alpha)\tilde{\nu}(\tilde{\alpha}) \\ &\qquad \times \mathbf{1}_{s_\alpha(\mu)\in C}\mathbf{1}_{s_\alpha(\tilde{\mu})\in C}\mathbf{1}_{\{\eta\{s_\alpha(\mu),s_\alpha(\tilde{\mu})\}>3\varepsilon_\alpha/2\}}\mathrm{d}\alpha\mathrm{d}\tilde{\alpha}\mathrm{d}\mu\mathrm{d}\tilde{\mu} \\ &\geq \int \kappa_1\kappa_2\kappa_3^2\nu(\alpha)\tilde{\nu}(\tilde{\alpha})\mathbf{1}_{\alpha\in A}\mathbf{1}_{\tilde{\alpha}\in A} \\ &\geq \kappa_1\kappa_2\kappa_3^2\kappa_4^2 \,. \end{aligned}$$

□

22 Application: hierarchical G & K distribution

The G & K distribution is a notorious example [Prangle, 2017] of intractable distribution. It depends on parameters (μ, B, g, k) and is defined by its inverse cumulative distribution function

$$F^{-1}(x; \mu, B, g, k, c) = \mu + B\left(1 + c\frac{1 - e^{-gz(x)}}{1 + e^{-gz(x)}}\right)\left(1 + z(x)^2\right)^k z(x)$$

where z is the quantile function of the standard normal distribution, and c is a constant typically set to 0.8 [Prangle, 2017]. While the likelihood function is intractable, it is straightforward to simulate realisations of this distribution, making it a favourite benchmark for ABC methods (see for instance Fearnhead and Prangle, 2012).

Here, we analyse two hierarchical versions of this model, both of the form:

$$\mu_i \sim \mathcal{N}(\alpha, 1) \qquad x_i \sim gk(\mu_i, B, g, k) \qquad i = 1, \ldots, n \,. \tag{4}$$

In a first experiment, we assume that the parameters B, g and k are known and we infer the position parameters (μ_i). This leads to the graphical model represented on the left of Figure 4.5. We refer to this model as the simple hierarchical G & K model.

For a hyperprior $\alpha \sim \mathcal{U}[-10, 10]$, the assumptions of Theorem 8 are satisfied. Figure 4.6 compares the results of our algorithm with those of ABC for a similar computational cost in dimension $n = 50$, and ABC-SMC (same as before) for a

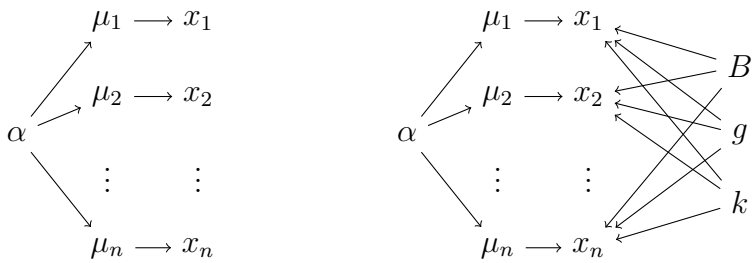


FIGURE 4.5 – Left: Simple hierarchical G & K model; Right: doubly hierarchical G & K model

higher computational cost, with 1000 particles, 500 iterations leading to a computational cost roughly 20 times longer. As in Section 21.2, ABC-Gibbs outperforms both other methods: Vanilla ABC is overdispersed, and carefully calibrated ABC-SMC is either highly peaked at the wrong location or producing results similar with ABC.

As a second experiment, we infer all parameters (B, g, k, α and the μ_i) in Equation 4, with independent hyperpriors $\alpha \sim \mathcal{U}(-10, 10)$ and $B, g, k \sim \mathcal{U}(0, 1)$. This corresponds to the graphical model represented on the right of Figure 4.5, which we refer to as a doubly hierarchical G & K model. The same summary statistic is used at every step of the algorithm, namely the octiles of the observations. Let $q(x, p)$ be the p-th quantile of sample x and take two observations x_1 and x_2; our distance function is

$$d(x_1, x_2) = \sum_{i=0}^{8} \left| q\left(x_1, i/8\right) - q\left(x_2, i/8\right) \right|.$$

It is straightforward to that the assumptions of Theorem 6 are satisfied by this model, when considering the parameters in the order $\alpha, B, g, k, (\mu_i)$.

Figures 4.6, 4.7 and 4.8 compare the output of ABC-Gibbs with Vanilla ABC and ABC-SMC in the same implementation as before, under a fixed budget of $2 \cdot 10^6$ model simulations for ABC-Gibbs and ABC; ABC-SMC is run with 10^3 particles, for 500 steps, resulting in a a grand total computational cost larger than $2.5 \cdot 10^7$ simulations. Note that there exist analytical approximations of the G & K posterior that give better results than ABC methods in the non hierarchical case, however none of these methods can be easily extended to the hierarchical case.

The simple and double hierarchical G & K models lead to comparable results. ABC-Gibbs provides consistently better results (that are more concentrated around the true value), than ABC. The approximation provided by ABC-SMC is less peaked and occasionaly exhibits a bias, if less visible in the hyperaparameter case. Sequential Monte Carlo is supposed to iteratively reduce the threshold

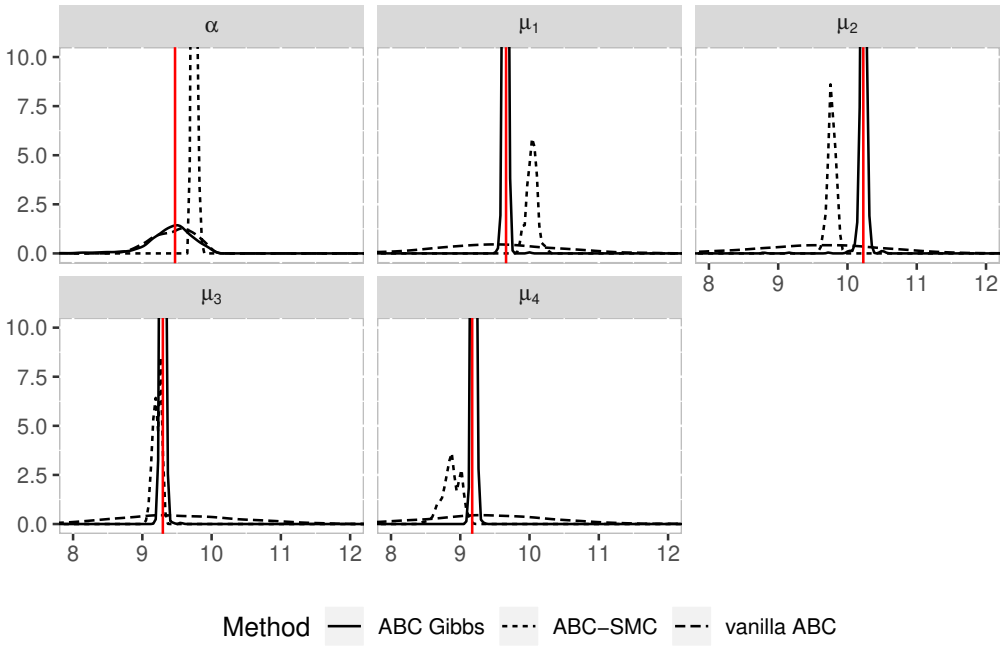


FIGURE 4.6 – Posterior approximations for the simple hierarchical G & K model. The y axis is truncated as the ABC-SMC pseudo-posterior is very peaked. The red vertical lines identify the value of the parameters used in the simulation.

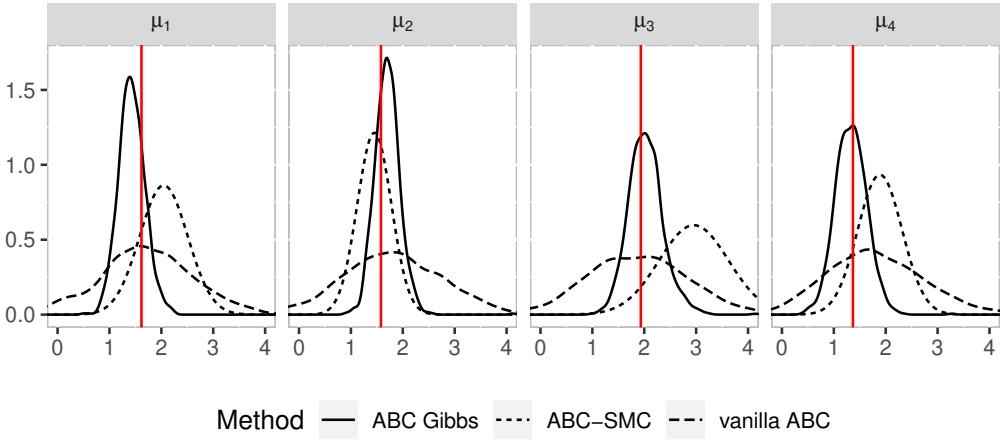


FIGURE 4.7 – Posterior densities for the first four parameters, among 50, $\mu_1, \ldots, \mu_4$ in the doubly hierarchical g & k model.

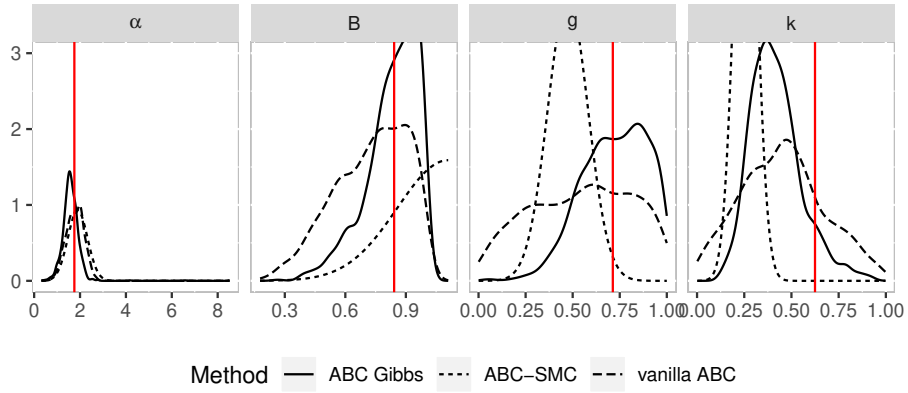


FIGURE 4.8 – Posterior densities for the top-level parameters α, B, g and k in the doubly hierarchical g & k model

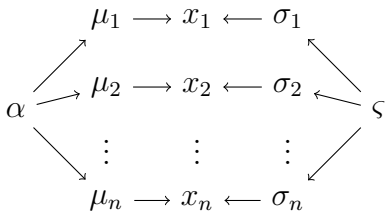


FIGURE 4.9 – Hierarchical dependence structure used in the application of Section 23.3.

of the approximation; however, due to the difficulty of calibrating, the starting points, the reduction is quite slow. It is thus unlikely a further increase in the computational time would lead to higher improvements.

In Section 23 of the Supplementary material, we consider another example (a hierarchical Moving Average model), for which the results are similar.

23 Application: Moving average model

23.1 Model and implementation

In this section, we study a hierarchical moving average model. A graphical representation of the hierarchy is shown in Figure 4.9. We denote $\mathcal{MA}_2(\mu, \sigma^2)$ the distribution of a second order moving average model with parameters $\mu = (\mu_1, \mu_2)$ and σ^2, that is:

$$x(t) = y_t + \mu_1 y_{t-1} + \mu_2 y_{t-2}\,, \qquad \text{with } y_t \sim \mathcal{N}(0, \sigma^2) \text{ for integer } t \geq -1.$$

We consider a hierarchical version of the $\mathcal{MA}_2$ model, consisting of n parallel observed series and $3n+5$ parameters with the following dependencies and prior laws: for $j = 1, \ldots, n$,

$$\begin{aligned} x_j &\sim \mathcal{MA}_2(\mu_j, \sigma_j^2) \\ \sigma_j^2 &\sim \mathcal{IG}(\varsigma_1, \varsigma_2) \\ \mu_j &= (\beta_{j,1} - \beta_{j,2}, 2(\beta_{j,1} + \beta_{j,2}) - 1) = (\mu_{j,1}, \mu_{j,2}) \end{aligned}$$

where $(\beta_{j,1}, \beta_{j,2}, 1-\beta_{j,1}-\beta_{j,2}) \sim \mathcal{D}ir(\alpha_1, \alpha_2, \alpha_3)$, and, if $\mathcal{E}$ denotes the exponential distribution and $\mathcal{C}_+$ the standard half-Cauchy distribution,

$$\alpha = (\alpha_1, \alpha_2, \alpha_3) \sim \mathcal{E}(1)^{\otimes 3}, \quad \varsigma = (\varsigma_1, \varsigma_2) \sim \mathcal{C}_+^{\otimes 2}.$$

We define $w(x_j)$ the distance between the first two autocorrelations of x_j and $x_j^\star$:

$$w^2(x_j) = \{\rho_1(x_j) - \rho_1(x_j^\star)\}^2 + \{\rho_2(x_j) - \rho_2(x_j^\star)\}^2,$$

and

$$\overline{x_j} = \frac{1}{\lfloor T/3 \rfloor} \sum_{t=1}^{\lfloor T/3 \rfloor} x_j(3t)$$

$$v(x_j) = \frac{1}{\lfloor T/3 \rfloor} \left| \sum_{t=1}^{\lfloor T/3 \rfloor} (x_j(3t) - \overline{x_j})^2 - \sum_{t=1}^{\lfloor T/3 \rfloor} (x_j^\star(3t) - \overline{x_j^\star})^2 \right|,$$

where T is the length of the time series. The rationale is that for a $\mathcal{MA}_2$ model $x(t)$ and $x(t+3)$ are independent. Vanilla ABC uses a related single pseudo-distance defined by

$$\delta(x) = \sum_{j=1}^{n} \left\{ \frac{w(x_j)}{q_j} + \frac{v(x_j)}{q_j'} \right\}, \tag{5}$$

where q_j and q_j' are the 0.1% quantiles of $w(x_j)$ and $v(x_j)$, respectively. This choice is constrained by the fact that these quantities appear to have undefined mean and variance.

For the current model, we have the following implementation: First, the μ_j's are updated using the pseudo-distance $d_{\mu_j}(x_j, x_j^*) = w(x_j)$.

Second, the update of α relies on the sufficient statistic associated with Dirichlet distributions:

$$\boldsymbol{\mu} \mapsto \left(\sum_j \log((\mu_{j,2} + 2\mu_{j,1} + 1)/4), \sum_j \log((\mu_{j,2} + 2\mu_{j,1} + 1)/4) - \mu_{j,1}) \right)$$

Third, the σ_j's are updated using the pseudo-distance $d_{\sigma_j}(x_j, x_j^*) = v(x_j)$. And last, ς is updated using the standard sufficient statistic associated with gamma distributions.

The two algorithms output samples from the two pseudo-posteriors. To compare the quality of the two samplers, we simulate new synthetic data from each parameter set in the output, and compute the distance (5) between observed and simulated samples, which is the distance used by ABC. If ABC-Gibbs produces a smaller value than the ABC associated with this distance, this is an indicator of a better fit of the ABC-Gibbs distribution with the true posterior. As in the previous experiment, the total number of simulations of the time series is used as the default measure of the computational cost for the associated algorithm.

23.2 Toy dataset

Consider a synthetic dataset of $n = 5$ times series each with length $T = 100$. Both samplers return samples of size $N = 1000$. The hyperparameters used to produce the true parameters and the simulated observed series are $\alpha = (1, 2, 3)$ and $\varsigma = (1, 1)$. In ABC-Gibbs, the μ_j's are updated based on $N_\mu = 1000$ time series, while the other parameters are updated based on $N_\alpha = N_\sigma = N_\varsigma = 100$ replicas. The overall computational cost for ABC-Gibbs is $N_{\text{tot}} = 5.5 \cdot 10^6$, also used by ABC to run $1.1 \cdot 10^6$ simulations of the whole hierarchy. The computational cost is slightly superior for ABC, as we have to simulate many more Dirichlet and Gamma random variables.

When evaluating the mean of the posterior predictive distance (5), ABC-Gibbs achieves an average of 274.1 ± 2.5, and ABC an average of 436.8 ± 1.6, based on 100 replicates. The sample output by ABC-Gibbs thus offers a noticeably better quality than the one generated by ABC from this perspective. The ABC output barely differs from a simulation from the prior, as shown in Figure 4.10 for the parameter μ_1.

23.3 Stellar flux

We now apply this model to stellar flux data. The 8GHz daily flux emitted by seven stellar objects is analysed in Lazio et al. [2008], and the data were made public by the Naval Research Laboratory: `https://tinyurl.com/yxorvl4u`. Once a few missing observations have been removed, Lazio et al. [2008] suggest that the model described in Section 23.1 may be well suited to these data, with $T = 208$. In ABC-Gibbs, the μ_j's are updated based on $N_\mu = 500$ time series, while the other parameters require $N_\alpha = N_\varsigma = N_\sigma = 100$ replicas. (The overall computing time is the same for the toy and the current datasets; one hour on an Intel Xeon CPU E5-2630 v4 with rate 2.20GHz.)

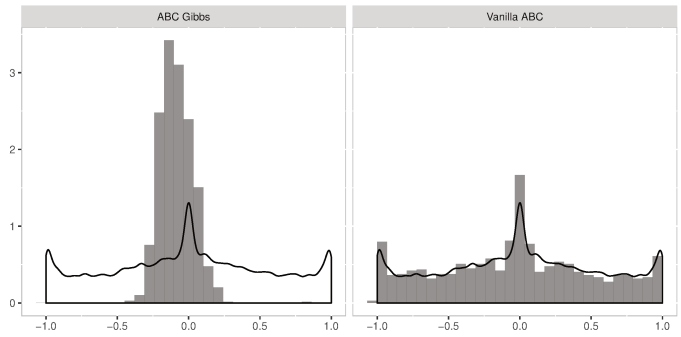


FIGURE 4.10 – For the toy dataset of subsection 23.2, approximate posterior of μ_1 compared with the prior for ABC-Gibbs (left) and ABC (right). The true value was -0.06.

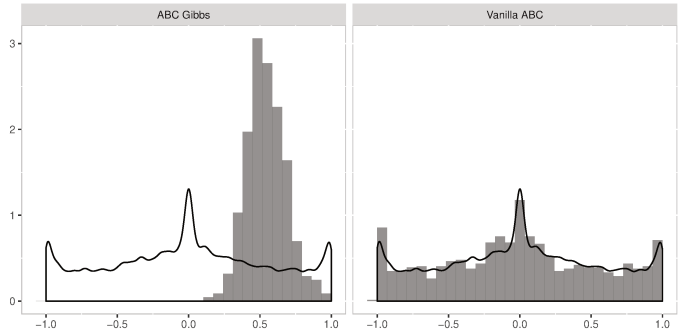


FIGURE 4.11 – For the stellar dataset of subsection 23.3, approximate posterior of μ_1 compared with the prior for ABC-Gibbs (left) and ABC (right)

The average posterior distance to the observed sample is 232.8 ± 1.25 for ABC-Gibbs and 535 ± 0.95 for ABC. The poor fit of the latter is confirmed in Figure 4.11, as it again stays quite close to the prior for the μ's. Since our model differs from the one proposed in Lazio et al. [2008], estimators cannot be directly compared.

24 Application: full dependence model

The concept of ABC-Gibbs is by no means restricted to hierarchical settings. It applies in full generality to any decomposition or completion of the parameter θ into n terms, $(\theta_1, \ldots, \theta_n)$. For simplicity's sake, we only analyse below the case of $n = 2$ parameters, and furthermore assume that θ_1 and θ_2 are a priori independent. The extension to $n \geq 2$ parameters, or non-independent parameters, is straightforward though cumbersome. The generic Algorithm 4.2 and Theorem 5 can thus be adapted to non-hierarchical models where $\theta = (\theta_1, \theta_2)$, such that

the conditional posteriors $\pi(\theta_1 \mid x^\star, \theta_2)$ and $\pi(\theta_2 \mid x^\star, \theta_1)$ depend on the entire dataset $x^\star$ rather than a significantly smaller subset. This setting implies that the approximation steps in ABC-Gibbs will mostly require the simulation of objects of the same size as in ABC.

When the statistics s_1 and s_2 are identical, a single distance can be used, with $\varepsilon_1 = \varepsilon_2$. The resulting stationary distribution is then the same as for ABC, since it is proportional to

$$\int \pi(\theta_1)\pi(\theta_2) f(x \mid \theta_1, \theta_2) \mathbf{1}_{\eta\{s_1(x), s_1(x^\star)\} \leq \varepsilon_1} \, \mathrm{d}x.$$

Formally, these statistics should however be different, since more efficient and smaller-dimension statistics can be calibrated to each parameter.

As an illustration, consider an example inspired by inverse problems [Kaipio and Fox, 2011], in a simplified version. These problems, although deterministic, are difficult to tackle with traditional methods, as the likelihood function is typically extremely expensive to compute [Neal, 2012], requiring the use of surrogate models, and thus approximations. Let y be the solution of the heat equation on a circle defined for $(\tau, z) \in]0, T] \times [0, 1[$ by

$$\partial_\tau y(z, \tau) = \partial_z \left\{ \theta(z) \partial_z y(z, \tau) \right\},$$

with $\theta(z) = \sum_{j=1}^n \theta_j \mathbf{1}_{\{(j-1)/n, j/n\}}(z)$ and with boundary conditions $y(z, 0) = y_0(z)$ and $y(0, \tau) = y(1, \tau)$. We assume y_0 known and the parameter is $\theta = (\theta_1, \ldots, \theta_n)$. The equation is discretised towards its numerical resolution. For this purpose, the first order finite elements method relies on discretisation steps of size $1/n$ for z and Δ for τ. A stepwise approximation of the solution is thus $\hat{y}(z, t) = \sum_{j=1}^n y_{j,t} \phi_j(z)$, where, for $j < n$, $\phi_j(z) = (1 - |nz - j|)\mathbf{1}_{|nz-j|<1}$ and $\phi_n(z) = (1 - nz)\mathbf{1}_{0<z<1/n} + (nz - n + 1)\mathbf{1}_{1-1/n<z<1}$, and with $y_{j,t}$ defined by

$$\frac{y_{j,t+1} - y_{j,t}}{3\Delta} + \frac{y_{j+1,t+1} - y_{j+1,t}}{6\Delta} + \frac{y_{j-1,t+1} - y_{j-1,t}}{6\Delta} \\ = y_{j,t+1}(\theta_{j+1} + \theta_j) - y_{j-1,t+1}\theta_j - y_{j+1,t+1}\theta_{j+1}.$$

We then observe a noisy version of this process, chosen as $x_{j,t} = \mathcal{N}(\hat{y}_{j,t}, \sigma^2)$.

In ABC-Gibbs, each parameter θ_m is updated with summary statistics the observations at locations $m - 2, m - 1, m, m + 1$. ABC relies on the whole data as statistic. In the experiments, $n = 20$ and $\Delta = 0.1$, with a prior $\theta_j \sim \mathcal{U}[0, 1]$, independently. Theorem 6 applies to this setting.

We compared both methods, using as above the same simulation budget and several experiments with various values of N_ε, keeping the total number of simulations constant at $N_{\text{tot}} = N_\varepsilon \cdot N = 8 \cdot 10^6$. As N_ε increases, the size N of the

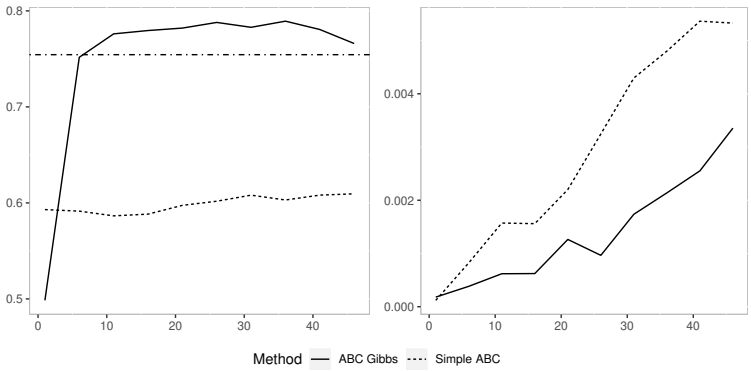


FIGURE 4.12 – For the heat equation model, mean and variance of the ABC and ABC-Gibbs estimators of θ_1 as N_ε increases, selected from among 20 parameters. The horizontal line shows the true value of θ_1.

posterior sample decreases. Figure 4.12 illustrates the estimations of θ_1. The ABC-Gibbs estimate is much closer to the true value of the parameter $\theta_1 = 0.75$, with a smaller variance. We emphasise once more that the choice of the ABC table size is critical, as for a fixed computational budget we must reach a balance between, on the one hand, the quality of the approximations of the conditionals (improved by increasing N_ε), and on the other hand Monte-Carlo error and convergence of the algorithm, (improved by increasing N). In our case, $N_\varepsilon = 10$ was clearly the best choice (low bias and low variance). While we have no systematic rule to choose this parameter, we however advise to choose it so that the approximation of the conditional is significantly different from the prior when run separately.

As in previous instances, ABC-Gibbs is much more efficient than ABC. For instance, Figure 4.13 shows that the posterior sample of θ_1 is more peaked around the true value for ABC-Gibbs. We repeated this experiment for a wide range of values for θ. In all, ABC-Gibbs gives estimates close to the true value, and is never outperformed by ABC. This is confirmed by evaluating the expectation of the posterior predictive distance to the whole data, ABC-Gibbs achieves an average of 39.2 ± 0.002, and ABC reaches an average of 103.8 ± 0.002, based on 100 replicates.

25 Nature of the limiting distribution

When addressing hierarchical models of the form of Equation (2), we gave conditions in Theorem 6 for Algorithm 4.2 to have a limiting distribution ν_ε. However, we did not specify the nature of this limiting distribution. We also showed that as the tolerance parameter ε goes to 0, ν_ε tends to the stationary distribution ν_0

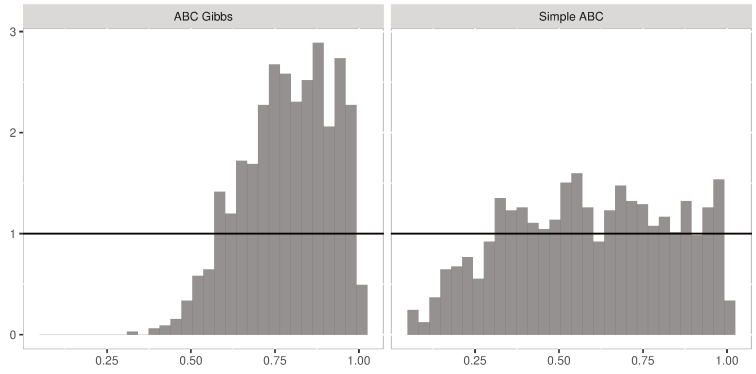


FIGURE 4.13 – For the model of section 24, approximate posterior of θ_1 compared with the uniform prior (black line) for ABC-Gibbs (right) and ABC (left)

of a Gibbs sampler with generators $\pi(\alpha)\pi(s_\alpha(\mu) \mid \alpha)$ and $\pi(\mu \mid \alpha)\pi(s_\mu(x) \mid \mu)$. It is possible that these generators are incompatible, that is, that there is no joint distribution associated with them. In such settings, the stationary distribution ν_0 does not enjoy these generators as conditionals. The incompatibility of conditionals may seem contradictory with the fact that our algorithm does converge to a distribution, but in the case of a compact parameter space there always exists a limiting distribution, the main issue being rather that the limiting distribution has no straightforward Bayesian interpretation.

There are however specific situations where there are theoretical guarantees that the limiting distribution ν_0 is in fact the posterior distribution associated with the summary statistics. According to Arnold and Press [1989] a necessary and sufficient condition for the conditionals to be compatible is the existence of two measurable functions $u(\alpha)$ and $v(\mu)$ such that

$$\frac{\pi(\alpha)\pi(s_\alpha(\mu) \mid \alpha)}{\pi(\mu \mid \alpha)f(s_\mu(x) \mid \mu)} = u(\alpha)v(\mu).$$

In particular, this occurs if s_α is sufficient. (This condition is not necessary, as it is also true for example if s_α is ancillary, although this is of limited interest.)

We have thus proven the following Proposition,

Proposition 25.1. *Under the assumptions of Theorems 5 and 7, a limiting distribution exists and converges, for both ε_μ and ε_α decreasing to 0, to the stationary distribution of a Gibbs sampler with conditionals:*

$$\pi(\alpha)\pi(s_\alpha(\mu) \mid \alpha) \text{ and } \pi(\mu \mid \alpha)f(s_\mu(x) \mid mu).$$

If s_α is sufficient, this limiting distribution is merely $\pi(\alpha, \mu)\pi(s_\mu(x) \mid \mu)$, that is the limiting distribution of ABC with summary statistic s_μ when the tolerance goes to 0.

We can state similar results for non hierarchical models, although each model requires its own proof. For example, for the full dependency model 24 with two parameters θ_1 and θ_2 we have the following Proposition:

Proposition 25.2. *If $\pi(\theta_1, \theta_2) = \pi(\theta_1)\pi(\theta_2)$ and $s_{\theta_1} = s_{\theta_2}$, as the tolerance ε goes to zero and under the assumptions of Theorems 6 and 7, ABC-Gibbs and ABC have the same limiting distribution.*

26 Discussion

The curse of dimensionality remains the major jamming block for the expansion of ABC methodology to more complex models as most of its avatars see their cost rise with the dimensions of the parameter and of the data [Li and Fearnhead, 2018]. This is particularly the case for high-dimensional parameters, since they require summary statistics that are at least of the same dimension and, unless the model under study is amenable to easily computed estimates of these parameters, a much larger collection of statistics is usually unavoidable. Breaking this curse of dimensionality by the Gibbs-like steps is thus as important for ABC methods as it was for Monte Carlo methods [Gelfand and Smith, 1990], as relying on a small number of summary statistics facilitates the derivation of automated or semi-automated approaches [Fearnhead and Prangle, 2012, Raynal et al., 2019] and offers the potential for simulating pseudo-data of much smaller size. In appropriate settings, ABC-Gibbs sampling provides a noticeable improvement of the efficiency of approximate Bayesian computation methods. We have established some sufficient conditions for the convergence of ABC-Gibbs algorithms. Questions remain, from checking such conditions in practice to a better understanding of the limiting distributions from an inferential viewpoint. A Gibbs-like setting could also allow practitioners to embed their model in a higher-dimensional model with auxiliary variables, with compatible conditionals and improved computational tractability. In all cases, constructing or selecting a low-dimension informative summary statistic for the approximation of the conditionals might be an unavoidable challenge to further improve the quality of the results.

Acknowledgements

This work greatly benefited from early discussions with Anthony Ebert, Kerrie Mengersen, and Pierre Pudlo, as well as detailed and helpful suggestions from an anonymous reviewer, to whom we are most grateful. We also acknowledge the Jean Morlet Chair which partly supported meeting in the Centre International de Rencontres Mathématiques in Luminy.

Part 5
Non-linear MCMC through particle methods

This part is a slightly modified version of a submitted with Antoine Diez and Jean Feydy [Clarté et al., 2019a]. The idea of the proofs is due to Antoine Diez, and the GPU implementation is due to Jean Feydy.

27 Introduction

27.1 Background

Monte Carlo methods are designed to estimate the expectation of an observable φ under a probability measure π. They approximate the quantity of interest by an estimator of the form

$$\frac{1}{n}\sum_{i=1}^{n}\varphi(X_i)\ ,$$

where X_i are independent and identically distributed (i.i.d.) random variables with law π. The law of large numbers ensures the convergence as $n \to +\infty$ of this estimator. For complex cases, a now classical procedure consists in constructing a Markov chain with stationary distribution π. Ergodic theory results then ensure that the estimator above still converges, even though the X_i are not independent from each other. The Metropolis-Hastings algorithm [Metropolis and Ulam, 1949, Metropolis et al., 1953, Hastings, 1970] provides a simple construction for such a Markov chain that only requires to evaluate π up to a multiplicative constant. The constructed chain is a random walk biased by an accept-reject step. Its convergence properties have been thoroughly studied, for example in Mengersen and Tweedie [1996].

This well-known procedure has become a building block for more advanced samplers, that are designed to overcome the known flaws of the Metropolis-Hastings algorithm: slow convergence, bad mixing properties for multimodal distributions *etc.* Such extensions include for instance the Wang and Landau algorithm [Bornn et al., 2013], regional MCMC algorithms [Craiu et al., 2009], or non Markovian (adaptive) versions [Haario et al., 2001, 1999, Atchadé and Rosenthal, 2005, Atchadé et al., 2011] where the next proposed state depends on the whole history of the process. The more recent PDMP samplers [Fearnhead et al., 2018, Vanetti et al., 2017] provide an alternative to the discrete time accept-reject scheme, replacing it

by a continuous time non reversible Markov process with random jumps. Finally, more complex algorithms are based on the evaluation of the gradient of π, see for instance the Metropolis-adjusted Langevin [Besag, 1994] and the Hamiltonian Monte Carlo algorithms [Duane et al., 1987].

A non-Markovian alternative to Metropolis-Hastings like methods is given by Importance Sampling algorithms. By drawing i.i.d. samples from an auxiliary distribution q, which is usually simple and called the importance distribution, we can build an estimator using the following identity:

$$\int \varphi(x)\pi(x)\mathrm{d}x = \int \frac{\pi(x)}{q(x)}\varphi(x)q(x)\mathrm{d}x \simeq \frac{1}{n}\sum_{i=1}^{n} w_i\varphi(X_i),$$

where the X_i are i.i.d. with law q and the $w_i \propto \pi(X_i)/q(X_i)$, are called the *importance weights*. The choice of q is critical, as bad choices can lead to a degeneracy of the importance weights. Iterative methods have been developed to sequentially update the choice of the importance distribution, and to update the X_i now interpreted as particles that evolve along iterations. Among these algorithms, we can cite the Sequential Importance Sampling algorithm Gordon et al. [1993], the Population Monte Carlo (PMC) methods [Douc et al., 2007, Cappé et al., 2004, 2008] or the recent Safe Adaptive Importance Sampling (SAIS) algorithm [Delyon and Portier, 2021]. This paradigm is in particular well-suited to the study of filtering problems Gordon et al. [1993], leading to the development of Sequential Monte Carlo (SMC) methods [Del Moral et al., 2006, Doucet et al., 2013]. A review of population-based algorithms and of the SMC method can be found in Jasra et al. [2007].

The SMC methodology has recently been used to design and study *nonlinear* MCMC algorithms [Andrieu et al., 2011]. This framework can be seen as a generalisation of some non-Markovian extensions of the Metropolis-Hastings algorithm (such as the "resampling from the past" procedure Haario et al. [2001], Atchadé and Rosenthal [2005]) but also allows the use of a wider range of algorithmic techniques. Examples are given in Andrieu et al. [2011, 2007] and are often based on the simulation of auxiliary chains. In the present article, we show that an alternative procedure based on the simulation of a swarm of *interacting* particles can also be used to approximate a nonlinear Markov chain. This provides a multi-particle generalisation of the Metropolis-Hastings procedure.

The duality between particle systems and non-linear Markov processes has first been underlined in statistical physics; on the mathematical side, it has been the subject of the pioneering works of McKean [1966, 1967], Kac [1956], Dobrushin [1979]. A key result is the *propagation of chaos* property formalised by Kac [1956], which implies that under an initial *chaoticity* assumption on the law of the particles, the empirical measure of the system at any further time converges towards

a deterministic limit; this type of limit is called *mean-field limit*. In a continuous time framework, this limit classically satisfies a nonlinear Partial Differential Equation (PDE) [Sznitman, 1991, Méléard, 1996]. The original diffusion framework has been extended to jump and jump-diffusion processes in Graham [1992a,b]. We refer the interested reader to Bellomo et al. [2017, 2019] for recent reviews and surveys of the applications of such models. We also mention that this methodology has been used in the analysis of particle methods in filtering problems [Del Moral, 2013, Andrieu et al., 2010].

27.2 Objective and methods

Let π be a target distribution on $E \subset \mathbb{R}^d$, known up to a multiplicative constant. The goal of the present article is to build a *nonlinear* Markov chain $(\overline{X}_t)_t$ on E that samples π efficiently. Given a sample $\overline{X}_t$ at iteration t, we draw $\overline{X}_{t+1}$ according to

$$\overline{X}_{t+1} \sim K_{\mu_t}(\overline{X}_t, \mathrm{d}y) \ ,$$

where the *transition kernel* is defined by:

$$K_{\mu_t}(x, \mathrm{d}y) := \underbrace{h(\alpha_{\mu_t}(x,y))\Theta_{\mu_t}(\mathrm{d}y|x)}_{\text{accept}} + \underbrace{\Big[1 - \int_{z\in E} h(\alpha_{\mu_t}(x,z))\Theta_{\mu_t}(\mathrm{d}z|x)\Big]\delta_x(\mathrm{d}y)}_{\text{reject}} \tag{6}$$

and where for $t \in \mathbb{N}$, $\mu_t \in \mathcal{P}(E)$ is the law of $\overline{X}_t$. In the discrete setting, this method is implemented by Algorithm 5.1, detailed below. It relies on the following quantities:

— The **proposal distribution**, a map

$$\Theta : E \times \mathcal{P}(E) \longrightarrow \mathcal{P}_0^{\mathrm{ac}}(E),$$

where $\mathcal{P}(E)$ denotes the set of probability measures on E and $\mathcal{P}_0^{\mathrm{ac}}(E)$ the subset of non-vanishing absolutely continuous probability measures. For $x \in E$ and $\mu \in \mathcal{P}(E)$, its associated proposal probability density function is denoted by:

$$\Theta(x,\mu)(y)\mathrm{d}y \equiv \Theta_\mu(y|x)\mathrm{d}y.$$

Intuitively, we can understand the probability distribution $\Theta(x,\mu)$ as an approximation of the target π that our method uses to *propose* new samples y in a neighborhood of a point x, relying on information that is provided by the current empirical distribution μ. To ensure a fast convergence of our method, the "model" $\Theta(x,\mu)$ should be both **easy to sample** and close to the target distribution π. In practice, the choice of a good proposal Θ depends on the assumptions that can be made on the distribution π. We discuss several examples in Section 29.

— For $\mu \in \mathcal{P}(E)$ and $x, y \in E$, the **acceptance ratio** is defined by:

$$\alpha_\mu(x, y) := \frac{\Theta_\mu(x|y)}{\Theta_\mu(y|x)} \cdot \frac{\pi(y)}{\pi(x)} \ .$$

This quantity expresses the relative appeals of the transition $x \to y$ for the "model" density $\Theta_\mu(\mathrm{d}y|x)$ and the ground truth target $\pi(\mathrm{d}y)$. Crucially, it can be computed even when the law of π is only known up to a multiplicative constant and allows our method to account for mis-matches between the proposal Θ and the distribution to sample π. Note that in practice, for the sake of numerical stability, the acceptance ratio is often manipulated through its logarithm:

$$\underbrace{\log \alpha_\mu(x, y)}_{\text{"correction"}} := \underbrace{\big[\log \pi(y) - \log \pi(x)\big]}_{\text{appeal of } x \to y \text{ for } \pi} - \underbrace{\big[\log \Theta_\mu(y|x) - \log \Theta_\mu(x|y)\big]}_{\text{appeal of } x \to y \text{ for } \Theta_\mu}.$$

— The **acceptance function** is a non-decreasing Lipschitz map of the form $h : [0, +\infty) \to\ [0, 1]$ which satisfies

$$\forall u \in [0, \infty), \quad u \cdot h(1/u) = h(u) \ .$$

A typical example is $h(u) = \min(1, u)$. As detailed in Algorithm 5.1, we combine the acceptance ratio $\alpha_\mu(x, y)$ and the acceptance function h to **reject** proposed samples y that are much more appealing for $\Theta_\mu(\cdot|x)$ than they are for π.

This **necessary correction** ensures that our method samples the target π instead of the simpler proposal distribution. On the other hand, it can also slow down our method if the proposed samples y keep being rejected. Efficient proposal distributions should keep the acceptance ratio high enough to ensure a renewal of the population of samples X_t^i at every iteration and thus provide good mixing properties.

Non-linearity. We say that the transition kernel is nonlinear due to its dependency on the law of the chain that it generates. When the proposal distribution does not depend on μ_t, the kernel is *linear* and we obtain the general form of the classical Metropolis-Hastings kernel.

Continuous and discrete analyses. We follow Andrieu et al. [2011] and split our analysis in two steps:

1. We show that our non-linear kernel admits π as a stationary distribution and study its asymptotic properties.

Input: An initial population of particles $(X_0^1, \dots, X_0^N) \in E^N$, a maximum time $T \in \mathbb{N}$, a proposal distribution Θ and an acceptance function h

Output: A sample $\left(X_t^i\right)_{1 \le i \le N;\ 1 \le t \le T}$

for $t = 0$ **to** $T - 1$ **do**

  for $i = 1$ **to** N **do**

    (Proposal) Draw $Y_t^i \sim \Theta_{\hat{\mu}_t^N}(\,\cdot\,|X_t^i)$ a proposal for the new state of particle i;

    (Acceptation) Compute $\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i) = \frac{\Theta_{\hat{\mu}_t^N}(X_t^i|Y_t^i)}{\Theta_{\hat{\mu}_t^N}(Y_t^i|X_t^i)} \cdot \frac{\pi(Y_t^i)}{\pi(X_t^i)}$;

    Draw $U_t^i \sim \mathcal{U}([0,1])$;

    if $U_t^i \le h\big(\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i)\big)$ **then**

      Set $X_{t+1}^i = Y_t^i$; `// Accept, probability` $h(\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i))$.

    else

      Set $X_{t+1}^i = X_t^i$; `// Reject, likely if` $\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i) \simeq 0$.

Algorithm 5.1: Collective Monte Carlo (CMC)

2. We present a practical implementation that enables the simulation of this kernel for different choices of the proposal distribution.

Analytical study. Starting from an initial distribution $\mu_0 \in \mathcal{P}(E)$, the law μ_t of the nonlinear chain at the t-th iteration satisfies

$$\mu_{t+1} = \mathcal{T}(\mu_t)$$

where $\mathcal{T} : \mathcal{P}(E) \to \mathcal{P}(E)$ is the **transition operator** defined by duality by:

$$\langle \varphi, \mathcal{T}(\mu) \rangle := \int_E \langle \varphi, K_\mu(x, \cdot) \rangle \mu(\mathrm{d}x) \tag{7}$$

for any test function $\varphi \in C_b(E)$. Thanks to the **micro-reversibility condition**:

$$\pi(x)\Theta_\mu(y|x)h(\alpha_\mu(x,y)) = \pi(y)\Theta_\mu(x|y)h(\alpha_\mu(y,x)), \tag{8}$$

the transition operator can be rewritten:

$$\begin{aligned}\langle \varphi, \mathcal{T}(\mu) \rangle = &\int_E \varphi(x)\mu(\mathrm{d}x) \\ &+ \iint_{E\times E} \pi(x)\Theta_\mu(y|x)h(\alpha_\mu(x,y))\varphi(x)\left(\frac{\mu(\mathrm{d}y)}{\pi(y)}\mathrm{d}x - \frac{\mu(\mathrm{d}x)}{\pi(x)}\mathrm{d}y\right),\end{aligned}$$

from which it can be easily seen that $\mathcal{T}(\pi) = \pi$.

We are going to develop an analytical framework in which the convergence of the sequence of iterations of the transition operator can be analysed. Although it is difficult to get optimal convergence rates, we rely on continuous time entropy methods to prove exponential convergence towards π for a large class of proposal distributions. We show that in an asymptotic regime to be detailed, the rate of convergence depends only on how close from the target is the initial condition. This is in contrast with the (linear) Metropolis-Hastings case where the convergence rate depends on the size of the random walk kernel. As a byproduct, in the linear case, we obtain a convergence result similar to the one obtained in Diaconis et al. [2011].

Efficient implementation. It is not straightforward in general to sample $\overline{X}_t$ from a nonlinear kernel. We therefore rely on a particle method to approximate such samples. Starting from a swarm of N particles $X_t^1, \dots, X_t^N \in E$ at the iteration t, we construct the next iteration by sampling independently for $i \in \{1, \dots, N\}$:

$$X_{t+1}^i \sim K_{\hat{\mu}_t^N}(X_t^i, \mathrm{d}y),$$

where

$$\hat{\mu}_t^N := \frac{1}{N} \sum_{i=1}^N \delta_{X_t^i} \in \mathcal{P}(E),$$

is the empirical measure of the system of particles. Note that the empirical measure process satisfies for all $\varphi \in C_b(E)$:

$$\mathbb{E}[\langle \varphi, \hat{\mu}_{t+1}^N \rangle | X_t^1, \dots X_t^N] = \langle \varphi, \mathcal{T}(\hat{\mu}_t^N) \rangle.$$

We show that as N goes to infinity and for each $t \in \mathbb{N}$, $\hat{\mu}_t^N$ converges towards a deterministic limit which is the t-th iterate of the nonlinear operator $\mathcal{T}$ starting from μ_0. Moreover we show that the N particles are asymptotically, in N, independent thus forming an approximation of a system of N independent nonlinear Markov chains with transition kernel (6).

A drawback of this approach is its high computational cost, that may scale in $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$ for some choices of the proposal Θ. To overcome this difficulty, we propose an implementation based on GPU, more precisely on the techniques developed in the KeOps library [Charlier et al., 2018] by the third author.

Outline. In Section 28 is devoted to the description of the general algorithmic framework and to the statement of convergence results for the particle scheme. Several variants of the main algorithm are presented in Section 29. The links between our algorithm and other related works are discussed in Section 30. The long-time asymptotics of the nonlinear Markov process is studied in Section 31.

The GPU implementation of the different algorithms is detailed in Section 32. Applications to various problems are presented in Section 33.

Throughout this article, we assume that π satisfies the following assumption.

Assumption 1. *The support of π, denoted by E, is a compact subspace of $\mathbb{R}^d$. The target distribution π is smooth and π does not vanish on E:*

$$m_0 := \inf_E \pi > 0 \quad \textit{and} \quad M_0 := \sup_E \pi > 0.$$

Notations. The following notations will be used throughout the article.

$\mathcal{P}(E)$ denotes the set of probability measures on E.

$\mathcal{P}^{\mathrm{ac}}(E)$ denotes the set of probability measures on E which are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$. A probability measure in $\mathcal{P}^{\mathrm{ac}}(E)$ is identified with its associated probability density function: when $f \in \mathcal{P}^{\mathrm{ac}}(E)$ we write indifferently,

$$f(\mathrm{d}x) \equiv f(x)\mathrm{d}x.$$

$\mathcal{P}_0^{\mathrm{ac}}(E) \subset \mathcal{P}^{\mathrm{ac}}(E)$ denotes the subset of continuous probability density functions which do not vanish on E (recall that E is compact).

A test function $\varphi \in C_b(E)$ is a continuous (bounded) function on E. For $\mu \in \mathcal{P}(E)$ we write indifferently

$$\langle \varphi, \mu \rangle \equiv \int_E \varphi(x)\mu(\mathrm{d}x).$$

$X \sim \mu$ means that the law of the random variable $X \in E$ is $\mu \in \mathcal{P}(E)$.

$W^1(\mu, \nu)$ denotes the Wasserstein-1 distance between the two probability measures $\mu, \nu \in \mathcal{P}(E)$.

28 General Framework and Mean-Field Approximation

Our implementation relies on a pathwise approximation of the nonlinear Markov chain $(\overline{X}_t)_t$ defined by the transition kernel (6). For a given proposal distribution Θ, the particle-based simulation is given by Algorithm 5.1.

On the practical side, the complexity of the proposal and acceptation steps highly depends on the proposal density Θ. Many examples of proposal distributions together with their associated sampling procedures are given in Section 29.

On the theoretical side, we can show that the system of particles defined by Algorithm 5.1 satisfies the propagation of chaos property and that the limiting

law is the law of the nonlinear Markov chain with transition kernel satisfying (6). From now on, we assume that the proposal distribution Θ satisfies the following assumptions, discussed in Remark 31.7:

Assumption 2 (Boundedness)**.** *There exist two constants $\kappa^-, \kappa^+ > 0$ such that for all $\mu \in \mathcal{P}(E)$ and for all $x, y \in E$:*

$$\kappa_- \leq \Theta_\mu(y|x) \leq \kappa_+.$$

Assumption 3 (L^∞ Lipschitz)**.** *The map $\Theta : E \times \mathcal{P}(E) \to \mathcal{P}_0^{\mathrm{ac}}(E)$ is uniformly Lipschitz for the L^∞-norm on E: there exists a constant $L > 0$ such that for all $x, y, x', y' \in E$ and for all $(\mu, \nu) \in \mathcal{P}(E)^2$:*

$$\left|\Theta_\mu(y|x) - \Theta_\nu(y'|x')\right| \leq L\Big(W^1(\mu, \nu) + |x - x'| + |y - y'|\Big).$$

Assumption 4 (W^1 non-expansive)**.** *The map $\Theta : E \times \mathcal{P}(E) \to \mathcal{P}_0^{\mathrm{ac}}(E)$ is non-expansive for the Wasserstein-1 distance: for all $x, x' \in E$ and for all $(\mu, \nu) \in \mathcal{P}(E)^2$,*

$$W^1\Big(\Theta_\mu(\mathrm{d}y|x), \Theta_\mu(\mathrm{d}y|x')\Big) \leq W^1(\mu, \nu) + |x - x'|.$$

The main result of this section is the following theorem, that links systems of interacting particles with nonlinear Markov processes and can be interpreted as a law of large numbers.

Theorem 11 (Mean-Field Limit)**.** *Let Θ be a proposal distribution which satisfies Assumptions 2, 3 and 4. Let $(X_0^i)_{i \in \{1,\dots,N\}}$ be N i.i.d. random variables with common law $\mu_0 \in \mathcal{P}(E)$ (*chaoticity assumption*). Let $t \in \mathbb{N}$ and let $(X_t^i)_{i \in \{1,\dots,N\}}$ be the N particles constructed at the t-th iteration of Algorithm 5.1. Then, the empirical measure $\hat{\mu}_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{X_t^i}$ satisfies:*

$$\hat{\mu}_t^N \underset{N \to +\infty}{\longrightarrow} \mu_t$$

*where $\mu_t = \mathcal{T}^{(t)}(\mu_0)$ is the t-th iterate of the transition operator (7) starting from μ_0 and where the convergence is the convergence in law (*i.e. *the weak-$\star$ convergence in the space $\mathcal{P}(\mathcal{P}(E))$).*

The proof of this theorem is based on coupling arguments inspired by Sznitman [1991] and adapted from Diez [2020]. It can be found in Section 31.4. As shown in Sznitman [1991, Proposition 2.2], the convergence in law of the empirical measure process is equivalent to the propagation of chaos property stated in the following corollary.

Corollary 12 (Propagation of Chaos)**.** *Let Θ be a proposal distribution which satisfies Assumptions 2, 3 and 4. Let $t \in \mathbb{N}$, $\ell \in \mathbb{N}$ and let $\mu_t^{\ell,N} \in \mathcal{P}(E^\ell)$ be the joint law at time t of any subset of ℓ particles constructed by Algorithm 5.1, initially i.i.d with initial common distribution $\mu_0 \in \mathcal{P}(E)$. For every ℓ-tuple of continuous bounded functions $\varphi_1, \dots \varphi_\ell$ on E, it holds that:*

$$\int_{E^\ell} \varphi_1(x_1) \dots \varphi_\ell(x_\ell) \mu_t^{\ell,N}(\mathrm{d}x_1, \dots, \mathrm{d}x_\ell) \underset{N \to +\infty}{\longrightarrow} \prod_{k=1}^{\ell} \langle \varphi_k, \mu_t \rangle,$$

where $\mu_t = \mathcal{T}^{(t)}(\mu_0)$ is the t-th iterate of the transition operator (7) starting from μ_0.

The statement of Corollary 12 corresponds to the original formulation of the propagation of chaos introduced by Kac [1956]. From our perspective, it justifies the use of Algorithm 5.1 and ensures that as the number of particles grows to infinity and despite the interactions between the particles, we asymptotically recover an i.i.d. sample. The final MCMC approximation of the expectation of an observable $\varphi \in C_b(E)$ is thus given at the t-th iteration by:

$$\int_E \varphi(x)\pi(\mathrm{d}x) \simeq \frac{1}{N} \sum_{i=1}^{N} \varphi(X_t^i).$$

29 Some Collective Proposal Distributions

The proposal distribution can be fairly general and so far, we have not detailed how to choose it. Several choices of proposal distributions are gathered in this section.

This proposal should use the maximum of information coming from the value of the target π, in order to increase the fitness of Θ_μ to the true distribution. However, the proposal must also allow for some exploration of the parameter space, this problem is addressed for some of the proposals.

Here, we only intend to present some of the proposals possible, each one having pros and cons, on the theoretical and practical side.

In view of Theorem 11, we can see each of the proposal distributions presented in this section either as a specific choice of nonlinear kernel (6) with its associated nonlinear process or as its particle approximation given by Algorithm 5.1. From this second perspective the proposal distribution can be seen as a specific interaction mechanism between the particles. More specifically, it depicts a specific procedure which can be interpreted as "information sharing" between the particles: given the positions of *all* the N particles at a given time, we aim at constructing

the best interaction mechanism which will favour a specific aspect such as acceptance, convergence speed, exploration etc. By analogy with systems of swarming particles which exchange local information (here, the local value of the target distribution) to produce global patterns (here, a globally well distributed sample), we call this class of proposal distributions **collective**. The class of methods introduced will be referred as Collective Monte Carlo methods (CMC). On the contrary, the nonlinear kernels introduced in Andrieu et al. [2007] do not belong to this class as explained in Section 30.1. For each proposal we give an implementation which, starting from population of particles $(X^1, \ldots, X^N) \in E^N$, returns a proposal Y.

Although our theoretical results (Theorem 11 above and Theorem 13 in Section 31) are general enough to encompass almost all of the proposal distributions described here, the validity and numerical efficiency of each of them will be assessed in Section 33 on various examples of target distributions.

29.1 Metropolis-Hastings Proposal (PMH)

Continuous formulation. The classical Metropolis-Hastings algorithm fits into our formalism, with

$$\Theta_\mu(\mathrm{d}y|x) = q(y|x)\mathrm{d}y,$$

where q is a fixed random walk kernel which does not depend on μ.

Particle implementation. In this case, Algorithm 5.1 reduces to the simulation of N independent Metropolis-Hastings chains in parallel.

29.2 Convolution Kernel Proposal (Vanilla CMC)

Continuous formulation. Let us then consider the following proposal distribution given by the convolution:

$$\Theta_\mu(\mathrm{d}y|x) = K \star \mu(y)\mathrm{d}y := \left(\int_E K(y-z)\mu(\mathrm{d}z)\right)\mathrm{d}y,$$

where K is a fixed *interaction kernel*, that is a (smooth) radial function which tends to zero at infinity. Typical examples are $K(x) = \mathcal{N}(0, \sigma^2 Id)$ and $K(x) = \frac{1}{|B_\sigma(0)|}1_{B_\sigma(0)}(x)$, where $\sigma > 0$ is fixed and $B_\sigma(0)$ denotes the ball of radius $\sigma > 0$ centred at 0 in $\mathbb{R}^d$. Note that the proposal distribution does not depend on the starting point x. It may happen that the proposed state falls outside E. In this case it will be rejected since π is equal to zero outside E. One can therefore take equivalently $\Theta_\mu(\mathrm{d}y|x) \propto K \star \mu(y)\mathbf{1}_E(y)\mathrm{d}y$ (the same remark holds for the other collective proposal distributions).

Particle implementation. Each particle chooses another particle and a draw a proposal from K centred at the chosen particle. This "resampling with mutation" procedure is somehow similar to a (genetic) Wright-Fisher model (see for instance Etheridge [2011] for a review of genetic models). Since a "mutation" may or may not be accepted, it can be described as a *biased Wright-Fisher model*.

The collective aspect is twofold: first the proposal distribution allows large scale move on all the domain filled by the N particles; then during the acceptance step, for a particle in X and a proposal in Y the acceptance ratio can be understood as a measure of discrepancy between the target ratio $\frac{\pi(Y)}{\pi(X)}$ and the observed ratio $\frac{K\star\hat{\mu}^N(Y)}{K\star\hat{\mu}^N(X)}$ between the (average) number of particles around Y and the (average) number of particles around X. In the linear Metropolis-Hastings case with a symmetric random-walk kernel, the acceptance ratio only takes into account the target ratio. As a consequence, the acceptance probability of a proposal state depends not only on how "good" it is when looking at the values of π but also on how many particles are (or are not) already around this proposal sate compared to the present state (and therefore on how accepting the proposal would improve the current global state of the system).

Remark 29.1 (Moderate interaction, part 1)**.** When $\sigma \to 0$ we obtain the degenerate proposal distribution $\Theta_\mu(\mathrm{d}y|x) = \mu(\mathrm{d}y)$ (which does not satisfy the assumption $\Theta(\mathrm{d}y|x) \in \mathcal{P}_0^{\mathrm{ac}}(E)$ in general) It would not make sense to take this proposal distribution at the particle level. However, it makes sense to consider the case $\Theta_f(y|x) = f(y)\mathrm{d}y$ in the nonlinear kernel (6) where $f \in \mathcal{P}_0^{\mathrm{ac}}(E)$. In Section 31 we will see that this degenerate proposal distribution still satisfies the assumptions of our convergence result and could lead to a better rate of convergence (see Remark 31.4). It is thus worth mentioning that this degenerate proposal distribution can also be obtained as the many-particle limit of a system of particles under an additional *moderate interaction* assumption Oelschläger [1985], Jourdain and Méléard [1998], Diez [2020]. See Remark 31.6 at the end of the proof of Theorem 11 for additional details.

Draw uniformly $j \in \{1, \dots, N\}$;
Draw $e \sim K$;
Set $Y = X^j + e$;

Algorithm 5.2: Proposal through Convolution Kernel

29.3 Markovian Mixture of Kernels Proposal (MoKA and MoKA-Markov)

Continuous formulation. A limitation of the Convolution Kernel Algorithm 5.2 is the fixed size of the interaction kernel. A remedy is given by the following

collective proposal distribution which is a convolution with a mixture of kernels (with different sizes) with (potentially) nonlinear mixture weights:

$$\Theta_\mu(\mathrm{d}y|x) = \sum_{p=1}^{P} \alpha_p(\mu) K_p \star \mu(y)\mathrm{d}y,$$

A possible choice for the weights is to take a solution of the following minimisation problem:

$$\min_{\alpha \in S^p} \int_E \phi\left(\frac{\sum_p \alpha_p K_p \star \mu(x)}{\pi(x)}\right) \mu(\mathrm{d}x), \tag{9}$$

where S_p denotes the p-simplex and where ϕ is convex non-negative such that $\phi(1) = 0$. Typically $\phi(s) = s \log s - s + 1$. In this case, when $\mu = \pi$ it corresponds to minimising the Kullback-Leibler divegence of the mixture proposal relative to π. In our experiments, we found that optimising the quantity $\int_E |\sum_p \alpha_p K_p \star \mu(x) - \pi(x)| \mu(\mathrm{d}x)$ leads to similar, slightly better results. Moreover, this choice is also numerically more stable so we chose to implement this version, that we call Markovian Mixture of Kernels (MoKA-Markov).

Another choice for the weights, which is **non-Markovian**, is to take α_p proportional to the geometric mean of the acceptance ratio of the particles which have chosen the kernel p at the previous iteration. This method will be referred as Mixture of Kernels Adaptive CMC (MoKA). It shares similarities with the D-kernel algorithm of Douc et al. [2007] and the arguments developed by the authors suggest that the two versions (Markovian and MoKA) may be asymptotically equivalent. The proof is left for future work.

Particle implementation. Same as in Algorithm 5.2 but with an additional step to choose a "mutation kernel" at each proposal step. The computation of the weights of the mixture can be done in a fully Markovian way at the beginning of each iteration before the proposal step or, in MoKA, are computed at the end of the iteration and used at the next iteration.

In Section 33 we will show that this algorithm can favours initial exploration if initially the particles are in an area of low potential.

Compute the weights $\alpha_1, \ldots, \alpha_P$ using (9);
Draw $p \in \{1, \ldots P\}$ with probability $(\alpha_1, \ldots, \alpha_P)$;
Draw uniformly $j \in \{1, \ldots, N\}$;
Draw $e \sim K_p$;
Set $Y = X^j + e$;

Algorithm 5.3: Markovian Mixture of Kernels proposal generation

29.4 Kernelised Importance-by-Deconvolution Sampling (KIDS)

Continuous formulation. Algorithms based on a simple convolution operator (Algorithms 5.2 and 5.3) keep a "blind" resampling step. In order to improve the convergence speed of such algorithms one may want to favour the selection of "good" states. For a fixed interaction kernel K, one can choose a proposal distribution of the form:

$$\Theta_\mu(y|x) = K \star \nu_\mu(y),$$

where ν_μ solves the following deconvolution problem with an absolute continuity constraint:

$$\min_{\nu \ll \mu} \int_E \log\left(\frac{K \star \nu(x)}{\pi(x)}\right) K \star \nu(x) \mathrm{d}x. \tag{10}$$

That is, we are looking for a weight function $w \geq 0$ which satisfies the constraint

$$\int_E w(x)\mu(dx) = 1$$

and such that the measure defined by $\nu_\mu(A) = \int_A w(x)\mu(\mathrm{d}x)$ minimises the KL divergence above. The function w is the Radon-Nikodym derivative of ν with respect to μ. In other words the proposal distribution focuses on the parts of the support of μ which are "closer" to π.

Particle implementation. In the case of an empirical measure $\hat{\mu}^N = \frac{1}{N}\delta_{X^i}$, the Radon-Nikodym weight function w can simply be understood as a vector of N weights $(w^1, \ldots, w^N) \in [0,1]^N$ such that $\sum_i w^i = 1$ and the measure $\nu_{\hat{\mu}^N}$ is thus a weighted empirical measure:

$$\nu_{\hat{\mu}^N} := \sum_{i=1}^N w^i \delta_{X^i}. \tag{11}$$

The deconvolution procedure gives more weight to the set of particles that are already "well distributed" according to π. These particles are thus more often chosen in the Wright-Fisher resampling step (see Algorithm 5.2).

Note that although the weighted empirical measure proposal (11) is very reminiscent of an Importance Sampling procedure, the computation of the weights here follows from a completely different idea.

In practice we solve the deconvolution problem using the Richardson-Lucy algorithm Richardson [1972], Lucy [1974] (also known as the Expectation Maximisation algorithm). See for instance Natterer and Wübbeling [2001, Section 5.3.2] where it is proved that the iterative algorithm below converges towards a minimiser of

the Kullback-Leibler divergence (10) in the case of an empirical measure μ. Note that the computation of the weights (Richardson-Lucy loop) can be done before the resampling step.

Set $w_i^{(0)} = 1$ for all $i \in \{1, \dots, N\}$;
for $s = 0$ *to* $S - 1$ **do**
  For all $i \in \{1, \dots, N\}$, update the weight by:
  $w_i^{(s+1)} = w_i^{(s)} \sum_{j=1}^N \frac{\pi(X_t^j) K(X_t^i - X_t^j)}{\sum_{k=1}^N w_k^{(s)} K(X_t^j - X_t^k)}$;
Set $w_i = w_i^{(S)}$ for all $i \in \{1, \dots, N\}$;
Normalise the weights $(w_1, \dots, w_N)$;
Draw $j \in \{1, \dots, N\}$ with probability $(w_1, \dots, w_N)$;
Draw $e \sim K$;
Set $Y = X^j + e$;

Algorithm 5.4: Kernel Importance-by-Deconvolution Sampling proposal generation

29.5 Bhatnagar-Gross-Krook sampling (BGK)

Continuous formulation. In Algorithms 5.2, 5.3 and 5.4, the proposal distribution is based on a (mixture of) symmetric kernels: this symmetry property is reflected in the proposal distribution and might not well represent the local properties of the target distribution. In dimension $d \geq 2$, we can adopt a different strategy by sampling proposals from a multivariate Gaussian distribution with a covariance matrix that is computed locally. An example is given by the following proposal distribution:

$$\Theta_\mu(\mathrm{d}y|x) = \left(\int_E G_{\widehat{\Sigma}_\mu(z)}(\widehat{m}_\mu(z) - y) \mu(\mathrm{d}z) \right) \mathrm{d}y,$$

where

$$\widehat{m}_\mu(z) = \frac{1}{\int_E K(z - z') \mu(\mathrm{d}z')} \int_E K(z - z') z' \mu(\mathrm{d}z'), \tag{12}$$

and

$$\widehat{\Sigma}_\mu(z) = \frac{1}{\int_E K(z - z') \mu(\mathrm{d}z')} \int_E K(z - z')(z' - \widehat{m}_\mu(z))(z' - \widehat{m}_\mu(z))^{\mathrm{T}} \mu(\mathrm{d}z'), \tag{13}$$

and where K is a fixed interaction kernel. This proposal distribution and the associated transition operator are reminiscent of a Bhatnagar-Gross-Krook (BGK) type operator Bhatnagar et al. [1954].

In the particular case of $K(x,y) \equiv 1$, we have a more simple algorithm which can be interpreted as a Markovian version of the Adaptive Metropolis-Hastings algorithm introduced by Haario et al. [2001]. However such algorithm does not benefit from the appealing properties of local samplers. Indeed, when the target distribution is multimodal, we can take more advantageously as interaction kernel $K(x) \propto \mathbf{1}_{|x|<\sigma}$, where the threshold σ allows the proposal to be adapted to the local mode. Note that moving across the modes is still possible thanks to the choice of another particle at the first step (see proposal algorithm below). The main issue is the choice of σ that must be higher than the size of the modes but smaller than the distance between the modes.

Particle implementation. Each particle, say X_t^i chooses another particle, say X_t^j. Then we compute the local mean and covariance around X_t^j and we draw a proposal Y_t^i for X_t^i according to a Normal law with the locally computed parameters. As before it may be cheaper to compute and store all the local means and covariances before the resampling loop.

Draw $j \in \{1 \ldots, N\}$ uniformly;
Compute $\kappa = \sum_i K(X_t^i, X_t^j)$;
Compute the local mean $\widehat{m}_{X^j} = \frac{1}{\kappa} \sum_i K(X_t^i, X_t^j) X_t^i$;
Compute the local covariance
$\widehat{\Sigma}_{X^j} = \frac{1}{\kappa} \sum_i K(X_t^i, X_t^j)(X^i - \widehat{m}_{X^j})(X^i - \widehat{m}_{X^j})^{\mathrm{T}}$;
Draw $Y \sim \mathcal{N}(\widehat{m}_{X^j}, \widehat{\Sigma}_{X^j})$;

Algorithm 5.5: BGK proposal generation

30 Related Works

30.1 Another Nonlinear MCMC sampler

A nonlinear kernel which does not fit into the "collective proposal" category has been introduced in Andrieu et al. [2011] and is defined by:

$$K_\mu(x, \mathrm{d}y) = (1-\varepsilon) K^{MH}(x, \mathrm{d}y) + \varepsilon Q_\mu(x, \mathrm{d}y),$$

where K^{MH} is the Metropolis-Hastings kernel and

$$Q_\mu(x, \mathrm{d}y) = \left(1 - \int_E \alpha(x,u)\mu(\mathrm{d}u)\right) \delta_x(\mathrm{d}y) + \alpha_\eta(x,y)\mu(\mathrm{d}y).$$

The function α is defined by: $\alpha_\eta(x,y) = \eta(x)\pi(y)/(\eta(y)\pi(x))$, that is, $\alpha_\eta(x,y)$ is the Metropolis ratio associated to **another** distribution $\eta \in \mathcal{P}_0^{\mathrm{ac}}(E)$. In Andrieu

et al. [2011], the authors investigated the case $\eta = \pi^{\tilde{\alpha}}$ for $\tilde{\alpha} \in (0,1)$. This kernel satisfies:

$$\iint_{E\times E} \phi(y) K_\eta(x, \mathrm{d}y)\pi(\mathrm{d}x) = \int_E \phi(y)\pi(\mathrm{d}y).$$

The sampling procedure is therefore quite different as it requires an auxiliary chain to build samples from η first in order to construct a sample from the desired non-linear kernel. More precisely, the authors propose the following iterative procedure to construct a couple of Markov chains (X_t, Y_t):

$$(X_{t+1}, Y_{t+1}) \sim \Big((1-\varepsilon)K^{MH}(X_t, \mathrm{d}x_{t+1}) + \varepsilon Q_{\hat{\mu}_t^Y}(X_t, \mathrm{d}x_{t+1})\Big)P(Y_t, \mathrm{d}y_{t+1}),$$

where P is a (linear) Markov transition kernel with invariant distribution η and $(Y_t)_t$ is a Markov chain with transition kernel P. The empirical measure of this chain is denoted by:

$$\hat{\mu}_t^Y = \frac{1}{t+1}\sum_{s=0}^{t} \delta_{Y_s}.$$

The final MCMC approximation of an observable φ is given in this case by:

$$\int_E \varphi(x)\pi(\mathrm{d}x) \simeq \frac{1}{t+1}\sum_{s=0}^{t} \varphi(X_s). \tag{14}$$

In this empirical sum, the successive iterations of the single chain $(X_t)_t$ are used. In the collective proposal framework introduced in Section 28, the algorithm produces N (asymptotically) independent copies of a nonlinear chain $(X_t^i)_t$, $i \in \{1, \ldots, N\}$ and we have at our disposal a *sequence* of MCMC approximations of the form:

$$\int_E \varphi(x)\pi(\mathrm{d}x) \simeq \frac{1}{N}\sum_{i=1}^{N} \varphi(X_t^i), \tag{15}$$

as $t \to +\infty$. We can therefore interpret the sum (14) as a *time average* and the sum (15) as an *ensemble average*.

30.2 Links with Importance Sampling Based Methods

Even though CMC does not use importance weights, it shares some similarities with importance sampling methods, in particular SMC [Del Moral et al., 2006] and PMC methods Cappé et al. [2004]. As SMC relies on tempering schemes, we will mainly focus on PMC.

According to Cappé et al. [2004], in PMC methods, without the importance correction, a regular acceptance step — as in Metropolis-Hastings — for each

mutation would lead to a simple parallel implementation of N Metropolis-Hastings algorithm. Under the same parallel, we can compare the mutation step in PMC with the proposal step of CMC.

In the first implementation of PMC, at each mutation step, each particle X_t^i is updated independently from the others, according to a kernel $q_{it}(\cdot)$ (that can depend on t and i), the new particle X_{t+1}^i is then associated with a weight proportional to the ratio $\pi(X_{t+1}^i)/q_{it}(X_{t+1}^i)$. In PMC, a mutation therefore occurs according to q_{it}, that is only depends on the position of the ancestor particle. In CMC, the mutation occurs according to $\Theta_{\hat{\mu}_t^N}(\cdot \mid X_t^i)$, the update thus depends on the position of all the particles as Θ depends on the empirical measure of the system $\hat{\mu}_t^N$. This additional dependency is particularly emphasised in the case of Algorithm 5.2. This corresponds to the Rao-Blackwellised version of PMC, in which we integrate over the position of all the particles.

Recently, Delyon and Portier [2021] also proposed to Rao-Blackwellise the mutation kernel in PMC while keeping an importance sampling framework. The resulting algorithm is non-Markovian and does not conserve the number of particles. At each iteration a batch of particles is added to the system according to the previous estimation of the target density. The number of particles N grows with the number of iterations. Once a particle is added, its position does not change, only its weight is updated along the iterations. A particle X_{N+1} is added according to $q_N = \sum_i w_i K_N(\cdot \mid X_i)$, where K_N is a kernel whose bandwidth typically decreases with N and where (w_i) is a vector of normalised importance weights. The weight of X_{N+1} is initially proportional to the standard importance weight associated to q_N and is then updated at each iteration by re-normalisation as new particles are added to the system. The weight of the added particle depends on all the other particles already present in the system from the first iteration. This method shares some similarities with ours, an important difference being that an old particle cannot be improved though time and a "bad" particle will indefinitely remain in the system at the same place, while its weight decreases through time, eventually increasing the computation cost.

Importance sampling based methods output unbiased estimators. For CMC, as stated before, except for the Metropolis-Hastings proposal, each one of the methods previously described is biased. Indeed, for a fixed number N of particles, the algorithm does not converge to the target distribution. For a large number of particles, the algorithm provides however a good approximation of the target density, according to Theorem 11 and Theorem 13. In addition, as a byproduct, we can re-use this approximation to provide an *unbiased* estimator by simply using the (sequence of) collective proposal distributions as importance distributions in any importance sampling based sampler. A more thorough study is left for future work, but some elements can be found in Appendix 30.3.

30.3 Importance Sampling plug-in of CMC

Importance sampling methods are unbiased, by definition, while CMC presents a (usually small) bias depending on the number of particles. To remove this bias, while keeping the original form of the algorithm, we present here a small plug in, without additional cost. We present the algorithm in 5.6.

Input: An initial population of particles $(X_0^1, \dots, X_0^N) \in E^N$, a maximum time $T \in \mathbb{N}$, a proposal distribution Θ and an acceptance function h
Output: T estimators $\hat{\varphi}_t$ of $E_\pi[\varphi(X)]$
for $t = 0$ *to* $T-1$ **do**
  for $i = 1$ *to* N **do**
    Draw $Y_t^i \sim \Theta_{\hat{\mu}_t^N}(\cdot|X_t^i)$ a proposal for the new state of particle i;
    Compute $\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i) = \frac{\Theta_{\hat{\mu}_t^N}(X_t^i|Y_t^i)\pi(Y_t^i)}{\Theta_{\hat{\mu}_t^N}(Y_t^i|X_t^i)\pi(X_t^i)}$;
    Store $W_t^i = \pi(Y_t^i)/\Theta_{\hat{\mu}_t^N}(Y_t^i|X_t^i)$ and Y_t^i;
    Draw $U_t^i \sim \mathcal{U}([0,1])$;
    if $U_t^i \leq h\big(\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i)\big)$ **then**
      Set $X_{t+1}^i = Y_t^i$;
    else
      Set $X_{t+1}^i = X_t^i$;
Set $\hat{\varphi}_t = \sum_i W_t^i \varphi(Y_t^i)$.

Algorithm 5.6: Collective Monte Carlo with IS output

At each step, we can store Y_t^i the proposed state for each particle, and the numerator of the acceptance ratio $W_t^i = \pi(Y_t^i)/\Theta_{\hat{\mu}_t^N}(Y_t^i|X_t^i)$. An estimator of $\mathbb{E}_\pi[\varphi(X)]$ is then $\sum_i W_t^i \varphi(Y_t^i)$. This term corresponds exactly to the importance ratio with π as target and $\Theta_{\hat{\mu}_t^N}$ as proposal distribution. This is similar in a sense to PMC, but the main difference is that *we do not propagate* Y_t^i unless it is accepted, and thus the weight cannot degenerate.

This addition do not interfere with the choice of the interaction Θ, and ensures that the resulting estimator is unbiased. We do not have to use all the points (Y_t^i) as we can only use some of the t. Notice that each of the $(Y_t^i)_i$ are independent by construction, but that $(Y_t^i)_{t,i}$ are not independent.

To measure the quality of a numerical method, it is common to use the Effective Sample Size (ESS) [Robert and Casella, 2013] that represents the equivalent number of i.i.d. samples needed to build an estimator of $\mathbb{E}_\pi[X]$ with same variance — notice that this quantity is mostly informative. In Importance sampling, the ESS is commonly estimated by $ESS(t) = (\sum_i (W_t^i)^2)^{-1}$, Interestingly enough, Theorem 11 and 13, shows that precisely, as the number of particles increase,

$\mathbb{E}[(\sum_i (W_t^i)^2)^{-1}]/N \to_{N\to\infty} 1$, as it tends to an i.i.d. sample distributed according to π. The precise convergence speed would depend on the constants appearing in the theorems.

As usually in importance sampling, this allows to estimate at no cost the normalizing constant — or in a Bayesian framework, the marginal likelihood of a mode: $\int_x \pi(x)\mathrm{d}x$, where we recall that π is non-normalised. Classical Metropolis-Hastings methods cannot estimate this quantity, while importance sampling methods (SMC, PMC) can provide an estimator whose variance may be too large to be usable. Figure 5.1 shows the result for the experiments of Section 33. These are only results on two "simple" targets, further numerical simulations would be needed to confirm the efficiency of our method to estimate this quantity.

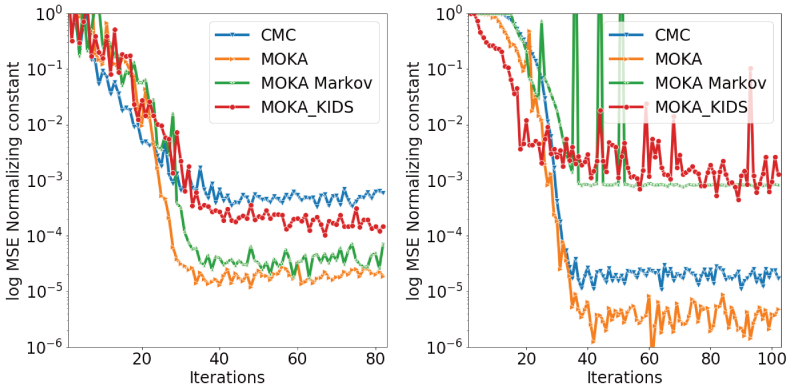


FIGURE 5.1 – Log-MSE of the normalizing constant computed for the banana-shaped distribution (left) and the 8-dimensional Gaussian mixture (right).

31 Convergence of the Nonlinear Process

31.1 Methodology and Main Result

In order to study the convergence as $t \to +\infty$ of the nonlinear process, we first start by a purely artificial technical trick: we look at the continuous time jump process associated to the discrete time nonlinear chain, where the jump times coincide with those of a rate 1 Poisson process [Brémaud, 1991, Chapter 8, Definition 2.2]. In this continuous time framework, starting from an absolutely continuous initial distribution $f_0 \in \mathcal{P}^{\mathrm{ac}}(E)$, the law f_t at time $t \in [0, +\infty)$ satisfies weakly the following nonlinear integro-differential equation:

$$\partial_t f_t(x) = \mathcal{T}(f_t)(x) - f_t(x) = \int_E \pi(x)\Theta_{f_t}(y|x)h(\alpha_{f_t}(x,y))\left(\frac{f_t(y)}{\pi(y)} - \frac{f_t(x)}{\pi(x)}\right)\mathrm{d}y \tag{16}$$

Note that f_t is also absolutely continuous with respect to the Lebesgue measure. The main result of this section is Theorem 13 which states that the solution of the integro-differential equation (16) converges exponentially fast to π as $t \to +\infty$ with a rate which depends only on the initial condition. This result will be valid under the following assumption.

Assumption 5 (Monotonicity)**.** *The proposal distribution Θ satisfies the following monotonicity property: there exists a function $c^- : (0, +\infty) \to (0, 1]$ such that for all $f \in \mathcal{P}_0^{\mathrm{ac}}(E)$,*

$$\inf_{(x,y)\in E} \frac{\Theta_f(y|x)}{\pi(y)} \geq c^- \left(\inf_{x\in E} \frac{f(x)}{\pi(x)} \right).$$

This "monotonicity" assumption allows us to control the balance between the proposal distribution and the target distribution π. Starting from a distribution f, the function c^- measures how the proposal distribution built on f is "far" from π. The best case is attained when $c^- \equiv 1$ since in this case $\Theta_f = \pi$.

This assumption is satisfied for all the "convolution based" methods such as Algorithms 5.2 and 5.3 since for $m > 0$, it holds that:

$$[\forall y \in E, \quad m\pi(y) \leq f(y)] \quad \Longrightarrow \quad [\forall y \in E, \quad mK * \pi(y) \leq K * f(y)],$$

and therefore, if the left-hand side condition holds, we obtain by dividing by $\pi(y)$:

$$\forall y \in E, \quad \frac{K * f(y)}{\pi(y)} \geq m \frac{K * \pi(y)}{\pi(y)}.$$

On the right-hand side of the last inequality the ratio $K * \pi(y)/\pi(y)$ depends only on π and is bounded from below, at least for small interaction kernels K, since $K * \pi$ converges uniformly towards π as $K \to \delta_0$. In the degenerate case $K = \delta_0$, we obtain $c^-(m) = m$ for all $m > 0$ (see Remark 31.4).

For the linear Metropolis-Hastings case the assumption is satisfied with a constant function $c^-(m) = c > 0$ for all $m > 0$, where $c > 0$ is the infimum of the random-walk kernel on the compact set E. This reasoning also applies for Algorithms 5.4 and 5.5 although more sharp results could be expected in specific cases.

We now state the main result of this section.

Theorem 13 (Convergence of the Nonlinear Process)**.** *Let Θ a proposal distribution which satisfies Assumption 5. Let $f_0 \in \mathcal{P}_0^{\mathrm{ac}}(E)$ and let f_t be the solution at time $t \in [0, +\infty)$ of the nonlinear integro-differentatial equation* (16)*. Then for all $t \geq 0$, it holds that:*

$$|f_t - \pi|_{\mathrm{TV}} \leq C_0 \mathrm{e}^{-\lambda t},$$

where $C_0 > 0$ depends only on f_0 and π and where

$$\lambda := c^- \left(\inf_x \frac{f_0(x)}{\pi(x)} \right) h(1) > 0.$$

The proof of this theorem will be based on entropy methods and can be found in Subsection 31.3.

Since Boltzmann's renowned H-Theorem [Villani, 2008], the concept of entropy has played a major role in our understanding of the long-time behaviour of thermodynamic systems. Although it can essentially be understood as a Lyapunov functional approach, it has been shown over last decades that the so-called entropy methods provide powerful and versatile tools to analyse the long-time asymptotics of a wide range of stochastic systems and Partial Differential Equations (PDE) under various metrics. From a PDE perspective, entropy methods can also provide wellposedness results and efficient numerical schemes for both linear and nonlinear equations despite the fact that the analysis of the latter looks generally much more delicate. We refer the interested reader to Schmeiser [2018], Jüngel [2016] for reviews on the subject.

In our case, we show that Equation (16) satisfies an entropy-dissipation relation (Subsection 31.2) which implies a boundedness result (Lemma 31.1) at the core of the proof of Theorem 13 (Subsection 31.3).

31.2 Entropy and Dissipation

The (relative) **entropy** of f_t with respect to π is defined as the following functional:

$$\mathcal{H}_\phi[f_t|\pi] := \int_E \pi(x) \phi \left(\frac{f_t(x)}{\pi(x)} \right) \mathrm{d}x, \tag{17}$$

where $\phi : [0, +\infty) \to [0, +\infty)$ is a convex function such that $\phi(1) = 0$. Typical choices of ϕ are:

$$\phi(s) = \frac{1}{2}|s-1|^2 \quad \text{and} \quad \phi(s) = s \log s - s + 1.$$

In the present article we will mainly focus on the first choice for which the relative entropy is simply equal to a weighted L^2 norm:

$$\mathcal{H}_\phi[f_t|\pi] = \frac{1}{2} \int_E |f_t(x) - \pi(x)|^2 \frac{\mathrm{d}x}{\pi(x)} \equiv \frac{1}{2} |f_t - \pi|^2_{L^2_{1/\pi}}.$$

In the second case $\phi(s) = s \log s - s + 1$, we note that:

$$\mathcal{H}_\phi[f_t|\pi] = \int_E f_t(x) \log \left(\frac{f_t(x)}{\pi(x)} \right) \mathrm{d}x \equiv D_{\mathrm{KL}}(f_t|\pi)$$

is the Kullback-Leibler divergence between f_t and π.

We can therefore see an entropy as a the measure of discrepancy between f_t and π for a metric specified by ϕ.

The continuous time framework allows us to take the time derivative of the entropy. The opposite of this quantity will be called the **dissipation**. Using (16), the dissipation is equal to:

$$\begin{aligned}\mathcal{D}_\phi(f_t|\pi) &= -\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_\phi[f_t|\pi] = -\int_E \phi'\left(\frac{f_t(x)}{\pi(x)}\right)\partial_t f_t(x)\mathrm{d}x \\ &= \frac{1}{2}\iint_{E\times E}\pi(x)W_f(x\to y)\left(\frac{f_t(x)}{\pi(x)}-\frac{f_t(y)}{\pi(y)}\right)\left(\phi'\left(\frac{f_t(x)}{\pi(x)}\right)-\phi'\left(\frac{f_t(y)}{\pi(y)}\right)\right)\mathrm{d}x\mathrm{d}y\end{aligned} \tag{18}$$

where

$$W_f(x\to y) := \Theta_f(y|x)h(\alpha_f(x,y)) \tag{19}$$

and where we have used the micro-reversibility property:

$$\pi(x)W_f(x\to y) = \pi(y)W_f(y\to x).$$

By convexity of ϕ, the dissipation $\mathcal{D}_\phi(f_t|\pi)$ is nonnegative:

$$\mathcal{D}_\phi(f_t|\pi)\geq 0.$$

As a first application of this result, we prove the following lemma which states an important global-in-time boundedness result.

Lemma 31.1. *Let f_t be the solution of the integro-differential equation* (16) *with initial condition $f_0\in\mathcal{P}_0^{\mathrm{ac}}(E)$. Then*

$$\inf_{x\in E}\frac{f_t(x)}{\pi(x)}\geq\inf_{x\in E}\frac{f_0(x)}{\pi(x)}$$

and

$$\sup_{x\in E}\frac{f_t(x)}{\pi(x)}\leq\sup_{x\in E}\frac{f_0(x)}{\pi(x)}.$$

Proof. Let us denote

$$m := \inf_{x\in E}\frac{f_0(x)}{\pi(x)}\quad\text{and}\quad M := \sup_{x\in E}\frac{f_0(x)}{\pi(x)}.$$

Let us take $\phi:[0,+\infty)\to[0,+\infty)$ a convex function such that $\phi\equiv 0$ on the segment $[m,M]$ and $\phi>0$ elsewhere. Note that since f_0 and π are both probability

densities, it holds that $m < 1$ and $M > 1$ and thus $\phi(1) = 0$. The entropy-dissipation relation (18) gives:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_\phi[f_t|\pi] \leq 0$$

and therefore for all $t \geq 0$,

$$\mathcal{H}_\phi[f_t|\pi] \leq \mathcal{H}_\phi[f_0|\pi] = 0$$

by definition of ϕ. As a consequence and since $\phi \geq 0$ and $\pi > 0$ on E, it holds that for all $t \geq 0$ and all $x \in E$,

$$\phi\left(\frac{f_t(x)}{\pi(x)}\right) = 0,$$

which implies that

$$\forall x \in E, \quad m \leq \frac{f_t(x)}{\pi(x)} \leq M.$$

□

To end this subsection we outline the strategy of the proof of Theorem 13. We will follow the classical steps which are detailed for instance in Jüngel [2016, Section 1.3] and can be applied to various linear and nonlinear jump and diffusion processes.

1. Compute the dissipation $\mathcal{D}_\phi(f_t|\pi) = -\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_\phi[f_t|\pi]$.

2. Prove that the dissipation can be bounded from below by a multiple of the entropy: for a constant $\lambda > 0$,

$$\mathcal{D}_\phi(f_t|\pi) \geq \lambda\mathcal{H}_\phi[f_t|\pi].$$

3. Apply Gronwall lemma to the relation $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_\phi[f_t|\pi] \leq \lambda\mathcal{H}_\phi[f_t|\pi]$ to obtain the exponential decay of the entropy:

$$\mathcal{H}_\phi[f_t|\pi] \leq \mathcal{H}_0\mathrm{e}^{-\lambda t}.$$

4. Show that the entropy controls the TV distance and conclude that for some constants $c, C_0 > 0$:

$$|f_t - \pi|_{\mathrm{TV}} \leq c\mathcal{H}_\phi[f_t|\pi] \leq C_0\mathrm{e}^{-\lambda t}.$$

Remark 31.1 (The linear case)**.** In the linear case $\Theta_\mu(y|x) = K_\sigma(x-y)$ where K_σ is a fixed random walk kernel of size $\sigma > 0$ (typically a Gaussian kernel with standard deviation σ). If $K_\sigma(z) \geq c(\sigma) > 0$ for all $z \in E$, then Assumption 5 is satisfied with a constant lower-bound:

$$\forall m > 0, \quad c^-(m) = \frac{c(\sigma)}{M_0}.$$

The result of Theorem 13 still applies but in this case the convergence rate does not depend on the initial condition but rather on the random-walk kernel. In particular, when $\sigma \to 0$, it holds that $c(\sigma) \to 0$. As it can be expected, the convergence is slower for small kernels. This result is similar to the one found in Diaconis et al. [2011].

31.3 Proof of Theorem 13

In the proof of Theorem 13, we will need the following lemma.

Lemma 31.2. *Let $h : [0, +\infty) \to [0, 1]$ be a continuous non-decreasing function which satisfies*

$$\forall u \in (0, +\infty), \quad uh\left(\frac{1}{u}\right) = h(u).$$

Let $[a, b]$ a bounded interval with $a > 0$. Then

$$\inf_{(x,y)\in[a,b]^2} yh\left(\frac{x}{y}\right) = ah(1).$$

Proof. Since h is continuous non-decreasing, for each $y \in [a, b]$, the function $x \in [a, b] \mapsto yh(x/y)$ has a minimum in $x = a$. This shows that the minimum on the function $(x, y) \in [a, b]^2 \mapsto yh(x/y)$ is attained on the segment $\{(a, y), y \in [a, b]\}$. Since $yh(a/y) = ah(y/a)$, the same reasoning shows that this minimum is attained when $y = a$. The conclusion follows. □

We are now ready to prove Theorem 13.

Proof of Theorem 13. Let us consider $\phi(s) = \frac{1}{2}|s-1|^2$.Then it holds that:

$$\mathcal{H}_\phi[f_t|\pi] = \frac{1}{2}\int_E \pi(x)\left|\frac{f(x)}{\pi(x)} - 1\right|^2 \mathrm{d}x = \frac{1}{4}\iint_{E\times E} \pi(x)\pi(y)\left|\frac{f_t(x)}{\pi(x)} - \frac{f_t(y)}{\pi(y)}\right|^2 \mathrm{d}x\mathrm{d}y,$$

and

$$\mathcal{D}_\phi(f_t|\pi) = \frac{1}{2}\iint_{E\times E} W_f(x \to y)\pi(x)\left|\frac{f_t(x)}{\pi(x)} - \frac{f_t(y)}{\pi(y)}\right|^2 \mathrm{d}x\mathrm{d}y.$$

Using Lemma 31.1, it holds that for all $x \in E$,

$$m := \inf_{x' \in E} \frac{f_0(x')}{\pi(x')} \leq \frac{f_t(x)}{\pi(x)} \leq \sup_{x' \in E} \frac{f_0(x')}{\pi(x')} =: M.$$

and therefore, using Assumption 5 and Lemma 31.2 we deduce that:

$$\begin{aligned} W_{f_t}(x \to y) = \Theta_{f_t}(y|x) h(\alpha_{f_t}(x,y)) &= \frac{\Theta_{f_t}(y|x)}{\pi(y)} h\left(\frac{\Theta_{f_t}(y|x)\pi(x)}{\pi(y)\Theta_{f_t}(x|y)}\right)\pi(y) \\ &\geq c^-(m)h(1)\pi(y). \end{aligned}$$

From the entropy-dissipation relation (18), it follows that

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_\phi[f_t|\pi] = -\mathcal{D}_\phi(f_t|\pi) \\ &\leq -\frac{c^-(m)h(1)}{2} \iint_{E\times E} \pi(x)\pi(y) \left|\frac{f_t(x)}{\pi(x)} - \frac{f_t(y)}{\pi(y)}\right|^2 \mathrm{d}x\mathrm{d}y = -2c^-(m)h(1)\mathcal{H}_\phi[f_t|\pi]. \end{aligned}$$

Using Gronwall's inequality we then deduce that:

$$\mathcal{H}_\phi[f_t|\pi] \leq \mathcal{H}_\phi[f_0|\pi]\mathrm{e}^{-2c^-(m)h(1)t}.$$

The conclusion follows from the Cauchy-Schwarz inequality by writing

$$|f_t - \pi|_{\mathrm{TV}} = \int_E |f_t(x) - \pi(x)|\mathrm{d}x = \int_E \sqrt{\pi}\sqrt{\pi}\left|\frac{f_t(x)}{\pi(x)} - 1\right|\mathrm{d}x \leq \sqrt{2\mathcal{H}_\phi[f_t|\pi]},$$

where we have used the fact that the TV norm is equal to the L^1 norm of the probability density functions. □

Remark 31.2. The last inequality between the TV norm and the square root of the relative entropy is a simple form of a Csiszár-Kullback-Pinsker inequality [Jüngel, 2016, Appendix A], [Bolley and Villani, 2005].

Remark 31.3. Another natural choice for ϕ would be $\phi(s) = s\log s - s + 1$. As already noticed before, the relative entropy is in this case equal to the Kullback-Leibler divergence. However, the dissipation term becomes in this case:

$$\begin{aligned} \mathcal{D}_\phi(f_t|\pi) = \frac{1}{2}\iint_{E\times E} W_f(x \to y)\pi(x)\left(\frac{f_t(x)}{\pi(x)} - \frac{f_t(y)}{\pi(y)}\right) \\ \times \left(\log\left(\frac{f_t(x)}{\pi(x)}\right) - \log\left(\frac{f_t(y)}{\pi(y)}\right)\right)\mathrm{d}x\mathrm{d}y, \end{aligned}$$

and it is not clear that it can be bounded from below by the relative entropy. Note that this dissipation functional is very similar to the one obtained in the study of the Boltzmann equation (in this context, the Kullback-Leibler divergence is also called Boltzmann entropy). The long-time asymptotics of this equation is a long-standing problem and the specific question of whether the dissipation controls the entropy is the object of a famous conjecture by Cercignani [Desvillettes et al., 2001, Villani, 2003]. In our case, we know that the Kullback-Leibler divergence is decreasing with time but all this suggests that its exponential decay could be harder to obtain or could hold only in specific cases.

Remark 31.4. In this proof, the convergence rate λ is obtained by a crude estimate of the infimum of the jump rate $W_f(x \to y)$. We do not claim that this rate is optimal. In the degenerate case $\Theta_f(y|x) = f(y)$, the best rate obtained is equal to $h(1)$ (and thus equal to 1 when $h(u) = \min(1, u)$) by taking an initial condition arbitrarily close to π. See also Remark 29.1

31.4 Proof of Theorem 11

Let us start with the following lemma.

Lemma 31.3. *Let Θ be a proposal distribution which satisfies Assumptions 2, 3 and 4. Then the map*

$$\alpha : \mathcal{P}(E) \times E^2 \to \mathbb{R},\ (\mu, x, y) \mapsto \alpha_\mu(x, y) := \frac{\Theta_\mu(x|y)\pi(y)}{\Theta_\mu(y|x)\pi(x)},$$

is Lipschitz in the Wasserstein-1 distance, in the sense that there exists a constant $L_\Theta > 0$ (which depends also on π) such that for all $\mu, \nu \in \mathcal{P}(E)$ and $x, x', y, y' \in E$, it holds that:

$$|\alpha_\mu(x, y) - \alpha_\nu(x', y')| \leq L_\Theta\big(W^1(\mu, \nu) + |x - x'| + |y - y'|\big).$$

Proof. By the triangle inequality, it holds that:

$$|\alpha_\mu(x, y) - \alpha_\nu(x', y')| \leq \frac{\pi(y)}{\pi(x)} \left|\frac{\Theta_\mu(x|y)}{\Theta_\mu(y|x)} - \frac{\Theta_\nu(x'|y')}{\Theta_\nu(y'|x')}\right| + \frac{\Theta_\nu(x'|y')}{\Theta_\nu(y'|x')} \left|\frac{\pi(y)}{\pi(x)} - \frac{\pi(y')}{\pi(x')}\right|.$$

We bound each of the two terms on the right-hand side:

$$\begin{aligned}\frac{\pi(y)}{\pi(x)} \left|\frac{\Theta_\mu(x|y)}{\Theta_\mu(y|x)} - \frac{\Theta_\nu(x'|y')}{\Theta_\nu(y'|x')}\right| &\leq \frac{\pi(y)}{\pi(x)\Theta_\mu(y|x)} |\Theta_\mu(x|y) - \Theta_\nu(x'|y')| \\ &\quad + \frac{\pi(y)\Theta_\nu(x'|y')}{\pi(x)\Theta_\mu(x|y)\Theta_\mu(x'|y')} |\Theta_\mu(y|x) - \Theta_\nu(y'|x')| \\ &\leq \frac{LM_0}{m_0\kappa_-}\left(1 + \frac{\kappa_+}{\kappa_-}\right)\Big(W^1(\mu, \nu) + |x - x'| + |y - y'|\Big).\end{aligned}$$

and

$$\begin{aligned}\frac{\Theta_\nu(x'|y')}{\Theta_\nu(y'|x')}\left|\frac{\pi(y)}{\pi(x)}-\frac{\pi(y')}{\pi(x')}\right| &\leq \frac{\Theta_\nu(x'|y')}{\pi(x)\Theta_\nu(y'|x')}|\pi(y)-\pi(y')| \\ &\quad+\frac{\Theta_\nu(x'|y')\pi(y')}{\Theta_\nu(y'|x')\pi(x)\pi(x')}|\pi(x)-\pi(x')| \\ &\leq \frac{|\pi|_{\mathrm{Lip}}\kappa_+}{m_0\kappa_-}\left(|y-y'|+\frac{M_0}{m_0}|x-x'|\right),\end{aligned}$$

where $|\pi|_{\mathrm{Lip}}$ denotes the Lipschitz norm of π. Gathering everything gives the result with

$$L_\Theta = \frac{M_0}{m_0}\left(\frac{L}{\kappa_-}\left(1+\frac{\kappa_+}{\kappa_-}\right)+\frac{|\pi|_{\mathrm{Lip}}\kappa_+}{m_0\kappa_-}\right).$$

□

The strategy of the proof of Theorem 11 will be based on coupling arguments inspired by Sznitman [1991] and adapted from Diez [2020]. We start by the following trajectorial representation of the nonlinear Markov chain $(\overline{X}_t)_t$ defined by the transition kernel (6).

Definition 31.1 (Nonlinear process)**.** Let $\overline{X}_0 \sim f_0$ be an initial state where $f_0 \in \mathcal{P}(E)$. The state $\overline{X}_t$ at time $t \in \mathbb{N}$, $t \geq 1$, is constructed from $\overline{X}_{t-1}$ and the law of $\overline{X}_{t-1}$ denoted by $f_{t-1} \in \mathcal{P}(E)$ as following:

1. Take a proposal a random variable $\overline{Y}_t \sim \Theta_{f_{t-1}}(\cdot|\overline{X}_{t-1})$
2. Compute the ratio

$$\alpha_{f_{t-1}}(\overline{X}_{t-1},\overline{Y}_t) := \frac{\Theta_{f_{t-1}}(\overline{X}_{t-1}|\overline{Y}_t)\pi(\overline{Y}_t)}{\Theta_{f_{t-1}}(\overline{Y}_t|\overline{X}_{t-1})\pi(\overline{X}_{t-1})}.$$

3. Take $\overline{U}_t \sim \mathcal{U}([0,1])$ and if $\overline{U}_t \leq h\big(\alpha_{f_{t-1}}(\overline{X}_{t-1},\overline{Y}_t)\big)$, then accept the proposal, else reject it:

$$\overline{X}_t = \overline{X}_{t-1}\mathbf{1}_{\overline{U}_t \geq h(\alpha_{f_{t-1}}(\overline{X}_{t-1},\overline{Y}_t))} + \overline{Y}_t\mathbf{1}_{\overline{U}_t \leq h(\alpha_{f_{t-1}}(\overline{X}_{t-1},\overline{Y}_t))}.$$

From now on we consider N independent copies $\big(\overline{X}^i_t\big)_t$, $i \in \{1,\dots,N\}$, of the nonlinear process defined by Definition 31.1 and we define a coupled particle process $(X^i_t)_t$ such that for all $i \in \{1,\dots,N\}$, initially $X^i_0 = \overline{X}^i_0 \sim f_0$ and for each time $t \in \mathbb{N}$ we take:

1. the same jump decision random variables $U^i_t = \overline{U}^i_t \sim \mathcal{U}([0,1])$,

2. optimal proposals of the form $Y_t^i = s(\overline{Y}_t^i)$ where s is an optimal transport map between $\Theta_{f_{t-1}}(\cdot|\overline{X}_{t-1}^i)$ and $\Theta_{\hat{\mu}_{t-1}^N}(\cdot|X_{t-1}^i)$. Since these two probability measures are absolutely continuous with respect to the Lebesgue measure, the existence of such optimal transport map (Monge problem) is proved for instance in Champion et al. [2011] or Caffarelli et al. [2002]. By definition, the pathwise error between the proposals can thus be controlled by

$$\begin{aligned}\mathbb{E}\big[|Y_t^i - \overline{Y}_t^i|\big|\mathcal{F}_{t-1}\big] &= W^1\left(\Theta_{\hat{\mu}_{t-1}^N}(\cdot|X_{t-1}^i), \Theta_{f_{t-1}}(\cdot|\overline{X}_{t-1}^i)\right)\\ &\leq W^1\left(\Theta_{\hat{\mu}_{t-1}^N}(\cdot|X_{t-1}^i), \Theta_{\bar{\mu}_{t-1}^N}(\cdot|\overline{X}_{t-1}^i)\right)\\ &\qquad + W^1\left(\Theta_{\bar{\mu}_{t-1}^N}(\cdot|\overline{X}_{t-1}^i), \Theta_{f_{t-1}}(\cdot|\overline{X}_{t-1}^i)\right)\end{aligned}$$

where $\bar{\mu}_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{\overline{X}_t^i}$ and $\mathcal{F}_t$ is the σ-algebra generated by the processes up to time $t \in \mathbb{N}$. We conclude that:

$$\mathbb{E}\big[|Y_t^i - \overline{Y}_t^i|\big|\mathcal{F}_{t-1}\big] \leq W^1(\hat{\mu}_{t-1}^N, \bar{\mu}_{t-1}^N) + |X_{t-1}^i - \overline{X}_{t-1}^i| + e_t^N \tag{20}$$

where the error term e_t^N only depends on (the laws of) the N independent nonlinear processes $(\overline{X}_t^i)_t$:

$$e_t^N := W^1(\bar{\mu}_{t-1}^N, f_{t-1}^K).$$

Theorem 14. *Let $t \in \mathbb{N}$. The coupled nonlinear and particle processes $\left(\overline{X}_t^i, X_t^i\right)_t$ defined above for $i \in \{1, \ldots, N\}$ satisfy the following pathwise estimate:*

$$\forall i \in \{1, \ldots, N\}, \quad \mathbb{E}[|\overline{X}_t^i - X_t^i|] \leq \beta(N)\mathrm{e}^{tC_\Theta},$$

where $C_\Theta > 0$ is a constant which depends only on π and Θ and where $\beta(N)$ is defined by

$$\beta(N) := \begin{cases} CN^{-1/2} & \text{if } d = 1\\ CN^{-1/2}\log(N) & \text{if } d = 2\\ CN^{-1/d} & \text{if } d > 2\end{cases}, \tag{21}$$

where $C > 0$ is a constant which depends only on π. In particular $\beta(N) \to 0$ as $N \to +\infty$.

Proof. Let $t \in \mathbb{N}$, $t \geq 1$. One has:

$$\begin{aligned}\mathbb{E}\big[|X_t^i - \overline{X}_t^i|\big|\mathcal{F}_{t-1}\big] &= \mathbb{E}\big[|Y_t^i - \overline{Y}_t^i|\mathbf{1}_{U_t^i \leq \min(h_t^i, \overline{h}_t^i)}\big|\mathcal{F}_{t-1}\big]\\ &\quad + |X_{t-1}^i - \overline{X}_{t-1}^i|\mathbb{P}\big(U_t^i \geq \max(h_t^i, \overline{h}_t^i)\big|\mathcal{F}_{t-1}\big)\\ &\quad + \mathbb{E}\big[|X_t^i - \overline{Y}_t^i|\mathbf{1}_{h_t^i \leq U_t^i \leq \overline{h}_t^i}\big|\mathcal{F}_{t-1}\big]\\ &\quad + \mathbb{E}\big[|Y_t^i - \overline{X}_t^i|\mathbf{1}_{\overline{h}_t^i \leq U_t^i \leq h_t^i}\big|\mathcal{F}_{t-1}\big]\end{aligned}$$

where we write for short:

$$h_t^i \equiv h\big(\alpha_{\hat{\mu}_t^N}(X_t^i, Y_t^i)\big) \quad \text{and} \quad \overline{h}_t^i \equiv h\big(\alpha_{f_t}(\overline{X}_t^i, \overline{Y}_t^i)\big).$$

we deduce that:

$$\begin{aligned}\mathbb{E}\big[|X_t^i - \overline{X}_t^i|\big|\mathcal{F}_{t-1}\big] \leq\ & W^1(\hat{\mu}_{t-1}^N, \bar{\mu}_{t-1}^N) + 2|X_{t-1}^i - \overline{X}_{t-1}^i| \\ & + 2M_0(\mathbb{P}(h_t^i \leq U_t^i \leq \overline{h}_t^i\big|\mathcal{F}_{t-1}) + \mathbb{P}(\overline{h}_t^i \leq U_t^i \leq h_t^i\big|\mathcal{F}_{t-1})) + e_t^N \quad (22)\end{aligned}$$

The last two probabilities are bounded by $\mathbb{E}\big[|h_t^i - \overline{h}_t^i|\big|\mathcal{F}_{t-1}\big]$. Assuming that h is L_h-Lipschitz for a constant $L_h > 0$, it holds that:

$$|h_t^i - \overline{h}_t^i| \leq L_h\Big|\alpha_{\hat{\mu}_{t-1}}(X_{t-1}^i, Y_t^i) - \alpha_{f_{t-1}}(\overline{X}_{t-1}^i, \overline{Y}_t^i)\Big|.$$

Let $\bar{\mu}_t^N$ be the empirical measure of the N nonlinear Markov processes $\overline{X}_t^i$ at time t. It holds that:

$$\begin{aligned}|h_t^i - \overline{h}_t^i| \leq\ & L_h\Big|\alpha_{\hat{\mu}_{t-1}}(X_{t-1}^i, Y_t^i) - \alpha_{\bar{\mu}_{t-1}}(\overline{X}_{t-1}^i, \overline{Y}_t^i)\Big| \\ & + L_h\Big|\alpha_{\bar{\mu}_{t-1}}(\overline{X}_{t-1}^i, \overline{Y}_t^i) - \alpha_{f_{t-1}}(\overline{X}_{t-1}^i, \overline{Y}_t^i)\Big|\end{aligned}$$

Using Lemma 31.3 we get:

$$|h_t^i - \overline{h}_t^i| \leq C_\Theta\left(W^1(\hat{\mu}_{t-1}^N, \bar{\mu}_{t-1}^N) + |X_{t-1}^i - \overline{X}_{t-1}^i| + |Y_t^i - \overline{Y}_t^i| + e_t^N\right),$$

with $C_\Theta := L_h L_\Theta$. Therefore:

$$\mathbb{E}\big[|X_t^i - \overline{X}_t^i|\big|\mathcal{F}_{t-1}\big] \leq (1 + C_\Theta)\left(|X_{t-1}^i - \overline{X}_{t-1}^i| + W^1(\hat{\mu}_{t-1}^N + \bar{\mu}_{t-1}^N)\right) + (1 + 2C_\Theta)e_t^N.$$

Let us define:

$$S_t := \frac{1}{N}\sum_{i=1}^N |X_t^i - \overline{X}_t^i|.$$

Summing the previous expression for i from 1 to N and dividing by N gives the following estimate for S_t:

$$\mathbb{E}[S_t|\mathcal{F}_{t-1}] \leq 2\,(1 + C_\Theta)\,S_{t-1} + (1 + 2C_\Theta)e_t^N \quad (23)$$

where we have used the fact that

$$W^1(\hat{\mu}_{t-1}^N, \bar{\mu}_{t-1}^N) \leq \frac{1}{N}S_t.$$

Taking the expectation in (23), we deduce that:

$$\mathbb{E}[S_t] \leq (1 + C_\Theta)\, \mathbb{E}[S_{t-1}] + C_\Theta \mathbb{E}[e_t^N] \tag{24}$$

where the value of the constant C_Θ has been updated by $C_\Theta \leftarrow 1 + 2C_\Theta$.

The error term can be controlled uniformly on t using Carmona and Delarue [2018, Theorem 5.8] or Fournier and Guillin [2015]. In particular, since π is a smooth probability density function on a compact set, it has finite moments of all order and therefore it follows from Fournier and Guillin [2015, Theorem 1] that:

$$\forall t \in \mathbb{N},\ \mathbb{E}[e_t^N] \leq \beta(N) \tag{25}$$

where $\beta(N)$ is defined by (21).

One can easily prove by induction that:

$$\mathbb{E}[S_t] \leq C_\Theta \beta(N) \sum_{s=0}^{t-1} e^{C_\Theta s} = \beta(N) \frac{C_\Theta}{\mathrm{e}^{C_\Theta} - 1} \mathrm{e}^{tC_\Theta} \tag{26}$$

By symmetry of the processes, all the quantities $\mathbb{E}[|X_t^i - \overline{X}_t^i|]$ are equal and their common value is $\mathbb{E}[S_t]$. The result follows. □

Proof of Theorem 11. Theorem 11 and Corollary 12 follow directly from Theorem 14 and Sznitman [1991, Proposition 2.2]. □

Remark 31.5 (Continuous time)**.** In a continuous time framework, a similar estimate can be derived. Let us assume that the time between two jumps is exponentially distributed with parameter 1 and that the inter-jump times are independent. Let us denote by $(T_n)_n$ the sequence of jump times. Let $n \in \mathbb{N}$. It holds that

$$\mathbb{E}[S_{T_n} | \mathcal{F}_{T_n^-}] \leq 2\,(1 + C_\Theta)\, S_{T_{n-1}} + C_\Theta \beta(N).$$

Taking the conditional expectation with respect to $\mathcal{G}_n = \sigma(T_1, T_2 - T_1, \ldots, T_n - T_{n-1})$, we obtain:

$$\mathbb{E}[S_{T_n} | \mathcal{G}_n] \leq 2\,(1 + C_\Theta)\, \mathbb{E}[S_{T_{n-1}} | \mathcal{G}_{n-1}] + C_\Theta \beta(N).$$

And thus, for all $n \in \mathbb{N}$,

$$\mathbb{E}[S_{T_n} | \mathcal{G}_n] \leq \beta(N) \frac{C_\Theta}{\mathrm{e}^{C_\Theta} - 1} \mathrm{e}^{C_\Theta n}.$$

For $t \in \mathbb{R}_+$, let us define $N_t = \sup\{n \in \mathbb{N},\quad T_n \leq t\}$. The random variable N_t is the index of the last jump before t. It is well known that N_t follows a Poisson law

with parameter t. It holds that:

$$\begin{aligned} E[S_{T_n}\mathbf{1}_{N_t=n}] &= \mathbb{E}[\mathbb{E}[S_{T_n}\mathbf{1}_{N_t=n}|\mathcal{G}_{n+1}]] \\ &= \mathbb{E}[\mathbf{1}_{N_t=n}\mathbb{E}[S_{T_n}|\mathcal{G}_n] \\ &\leq \beta(N)\frac{C_\Theta}{\mathrm{e}^{C_\Theta}-1}\mathrm{e}^{C_\Theta n}\mathbb{P}(N_t=n), \end{aligned}$$

where the second inequality comes from the fact that the event $\{N_t = n\}$ is $\mathcal{G}_{n+1}$ measurable and the fact that S_{T_n} is independent from $T_{n+1}-T_n$. As a consequence, since $S_t = S_{T_{N_t}}$, we conclude that:

$$\mathbb{E}[S_t] = \mathbb{E}[S_{T_{N_t}}] = \sum_{n=0}^{+\infty} E[S_{T_n}\mathbf{1}_{N_t=n}] \leq \beta(N)\frac{C_\Theta}{\mathrm{e}^{C_\Theta}-1}\exp(t(\mathrm{e}^{C_\Theta}-1)).$$

Remark 31.6 (Moderate interaction, part 2)**.** The result of Theorem 14 is stronger than the conclusion of Theorem 11 as it provides an explicit convergence rate in terms of N. This could be used to understand more precisely the moderate interaction assumption mentioned in Remark 29.1. In the case $\Theta_\mu(\mathrm{d}y|x) = K \star \mu(y)\mathrm{d}y$, one can take at the particle level (*i.e* in Algorithm 5.1) an interaction kernel $K \equiv K^N$ which depends on N and such that its size $\sigma_N \to 0$ as $N \to +\infty$ (and thus $K^N \to \delta_0$). As a consequence the constant $C_\Theta \equiv C_\Theta^N$ in Theorem 14 would depend on N. Since we have a precise control on $\beta(N)$ we can choose σ_N such that the following convergence still holds:

$$\beta(N)\mathrm{e}^{tC_\Theta^N} \underset{N\to+\infty}{\longrightarrow} 0.$$

In particular, σ_N should not converges to zero too fast, justifying the moderate interaction terminology introduced in Oelschläger [1985]. In the limit $N \to +\infty$ we then obtain that the emprirical measure $\hat{\mu}_t^N$ converges towards the t-th iterate of the transition operator (7) with the degenerate choice of proposal distribution $\Theta_f(\mathrm{d}y|x) = f(y)\mathrm{d}y$ which makes sense as soon as $f \in \mathcal{P}_0^{\mathrm{ac}}(E)$. We refer the reader to Jourdain and Méléard [1998] and Diez [2020] for two examples of propagation of chaos results under a moderate interaction assumption. Note that this result is mainly of theoretical interest as it does not give sharp estimates on how slow σ_N should decrease to zero.

Remark 31.7 (About the assumptions)**.** In order to prove a propagation of chaos property, it is usually assumed that the parameters of the problem are Lipschitz [Sznitman, 1991, Méléard, 1996]. This corresponds to the two Lipschitz assumptions 3 and 4. Propagation of chaos in non-Lipschitz settings is a more difficult problem (see for instance Jabin and Wang [2016] for a recent result).

Assumption 1 and Assumption 2 should be understood as technical assumptions. In the proof of Theorem 11, we use the fact that the acceptance ratio is Lipschitz (Lemma 31.3) which follows directly from Assumptions 1, 2 and 3. However, we could relax the compactness assumption 1 and keep the same Lipschitz property by replacing Assumptions 1, 2 and 3 by the following assumption.

Assumption 6. *The target distribution π does not vanish on E and the map*

$$\mathcal{P}(E) \times E^2 \to \mathbb{R}, \quad (\mu, x, y) \mapsto g_\mu(y|x) := \frac{\Theta_\mu(y|x)}{\pi(y)}$$

satisfies the two following properties.

— ***(Boundedness).*** *There exists two constants $\kappa_- > 0$ and $\kappa_+ > 0$ such that*

$$\forall (x, y) \in E^2, \quad \kappa_- \leq g(y|x) \leq \kappa_+.$$

— ***(Lipschitz).*** *There exists a constant $L > 0$ such that*

$$\forall (\mu, x, y), (\nu, x', y') \in \mathcal{P}(E) \times E^2,$$
$$|g_\mu(y|x) - g_\nu(y'|x')| \leq \Big(W^1(\mu, \nu) + |x - x'| + |y - y'| \Big).$$

In practice, this would necessitate a precise control of the tails of π and of the proposal distribution. It seems easier for us to check the compactness and boundedness assumptions 1, 2 and 3 (possibly up to truncating the support of π and replacing it by a compact set).

32 GPU implementation

Historically, a major bottleneck for the development of collective samplers based on interacting particles has been the computational cost associated to the simulation of N^2 pair-wise interactions between particles at every time step. To overcome this issue, many methods have been proposed throughout the years: let us cite for instance the Verlet and cell-list methods for short-ranged interactions [Sigurgeirsson et al., 2001], the Fast Multipole Method [Rokhlin, 1985, Greengard and Rokhlin, 1987] for long-ranged interactions or the more recent Random Batch Method introduced by Jin et al. [2020].

Most importantly, the last decade has seen the introduction of massively parallel GPU chips. Beyond the training of convolutional neural networks, GPUs can now be used to simulate generic particle systems at ever faster rates. The

present paper discusses the consequences of this hardware revolution on Monte Carlo sampling.

From the evaluation of kernel densities to the computation of importance weights in the Richardson-Lucy loop (Algorithm 5.4), the bottleneck of the Collective Proposal framework presented in Section 29 is the computation of an off-grid convolution of the form:

$$a_i = \sum_{j=1}^{N} K(X_t^i, X_t^j)\, b_j \tag{27}$$

for all i between 1 and N, with arbitrary weights b_j.

All CMC-related methods have $\mathcal{O}(N^2)$ time complexity, with a constant that is directly related to the efficiency of the underlying implementation of the "kernel sum" above.

In the machine learning literature, this fundamental operation is often understood as a matrix-vector product between an N-by-N *kernel matrix* $(K(X_t^i, X_t^j))$ and a vector (b_j) of size N. Common implementations generally rely on linear algebra routines provided e.g. by the PyTorch library Paszke et al. [2017a] and have a quadratic memory footprint: even on modern hardware, this prevents them from handling populations of more than $N = 10^4$ to 10^5 samples without making approximations or relying on specific sub-sampling schemes Yang et al. [2012].

Fortunately though, over the last few years, efficient GPU routines have been developed to tackle computations in the mould of (27) with maximum efficiency. These methods can be accessed through the KeOps extension for PyTorch Paszke et al. [2017b], NumPy Van Der Walt et al. [2011], R Team et al. [2013] and Matlab Mat [2017], that is developed, among others, by Jean Feydy, Charlier et al. [2018], Feydy [2020], Feydy et al. [2020] and freely available at `https://www.kernel-operations.io`. In practice, this library supports arbitrary kernels on the GPU with a linear memory footprint, log-domain implementations for the sake of numerical stability and outperforms baseline PyTorch GPU implementations by one to two orders of magnitude. Computing the convolution of (27) with a cloud of $N = 10^6$ points in dimension 3 takes no more than 1s on a modern chip. Pairwise interactions between populations of $N = 10^5$ samples may also be simulated in no more than 10ms, without making any approximation.

As detailed in our documentation, we run all the tests of Section 33 on a single gaming GPU, the Nvidia GeForce RTX 2080 Ti. With 10^4 to 10^6 particles handled at any given time, our simulations run in a handful of seconds at most, with performances that enable the real-time sampling of large populations of *interacting* samples.

33 Numerical experiments

We now run experiments on several target distributions in low and moderately large dimension. We always clip distributions on the unit (hyper)-cube $[0,1]^d$. We compare our method with the Safe Adaptive Importance sampling (SAIS) from Delyon and Portier [2021], which is one of the state-of-the-art importance sampling based methods (see Section 30.2). We also include as baseline a parallel implementation of Metropolis-Hastings (PMH) with a number of parallel runs that is equal to the number of particles in our method. For all the methods we chose a poor initialisation with N particles independently distributed according to $X_0^i = (0.9, \dots, 0.9)^T + 0.1 U_0^i$ where $U_0^i \sim \mathcal{U}([0,1]^d)$.

Among the variants of our method, we show results for vanilla CMC, MoKA-CMC, MoKA-Markov-CMC and MoKA-KIDS-CMC. Please note that we do not include the BGK and KIDS samplers in these experiments: although we believe that the ideas behind these methods are interesting enough to justify their presentation, we observe that they generally do not perform as well as the other CMC variants and leave them aside for the sake of clarity.

We compare the results in term of Energy distance Rizzo and Székely [2016] between a true sample generated by rejection and the population of particles at each step. As baseline, we show the average Energy distance between two iid exact samples of size N, and a 90% prediction interval for this quantity. For two independent samples X and Y of size n and m respectively, the energy distance is defined as:

$$\mathcal{E}(X,Y) = \frac{2}{nm}\sum_{i,j}\|X_i - Y_j\| - \frac{1}{n^2}\sum_{i,j}\|X_i - X_j\| - \frac{1}{m^2}\|Y_i - Y_j\|,$$

where $\|\cdot\|$ denotes the standard Euclidean norm.

Our code and its documentation are available online on KeOps website: `http://www.kernel-operations.io/monaco/`. All our experiments can be run on Google Colaboratory (`colab.research.google.com`) with a GPU within a few seconds to a few minutes, depending on the method and number of independent runs. We note that MoKA-KIDS is a slightly heavier method, as it relies on the Richardson-Lucy iterations to optimise the deconvolution weights. We also note that unlike our Markovian methods, the memory cost of SAIS significantly increases with the number of iterations – to the best of our knowledge, no procedure has yet been proposed to remove particles with time. We have not implemented the batch sampler as proposed in Delyon and Portier [2021], as the computation time is not a problem with our fast GPU implementation.

Another example of target distribution is included in Appendix 33.3 and we include results for the estimation of the normalizing constant of the target distributions below in Appendix 30.3. Experiments in non-Euclidean spaces such a

the Poincaré hyperbolic plane and the group of 3D rotation matrices are available online in our documentation.

33.1 Banana shaped distribution

Our first target is inspired by the Banana example from Delyon and Portier [2021], that is made up of three Gaussian bells with variance 0.2 and of a Banana shaped distribution. We represent a typical run of our method and the level sets of the distribution in Figure 5.2.

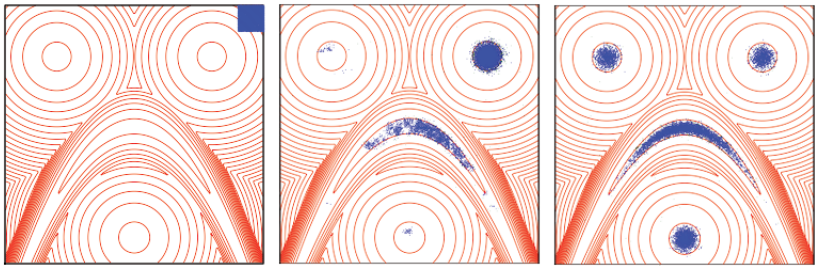


FIGURE 5.2 – Banana shaped distribution. Level sets (red) and particles (blue) from left to right at iterations 0, 10 and 50 for the MoKA-Markov sampler.

This distribution is renowned for being difficult to sample from, especially because of its geometry. Our proposals are uniform proposals in balls of various diameters depending on the method, that is not adapted to the distribution. We increase the difficulty of the problem with a starting particle swarm that is situated outside the modes of our target.

We run each method for 80 iterations with $N = 10^4$ particles. To promote the discovery of all modes, we run each method with an *exploration proposal*: a large Gaussian sample with standard deviation 0.30 that is selected with probability 0.01 and complements the adaptive CMC, PMH or SAIS proposal (that is selected with probability 0.99). To generate perturbations, the vanilla CMC, PMH and SAIS methods rely on a uniform proposal on a ball of radius 0.10. The other methods are based on kernel selection and rely on uniform proposals on balls of four different radii: 0.01, 0.03, 0.10 and 0.30.

In Figure 5.3, we present the energy distance through iterations for each of our methods. All the methods except PMH reach the minimal distance, that is the average distance between two exact iid samples from the target. SAIS seems to be initially the fastest method, but its convergence speed decreases along the iterations and it reaches a lower final distance compared to our methods. CMC is the slower of our methods, because the kernel is not adapted in size, while the other methods are comparable with each other. The straight line for Vanilla CMC

confirms the exponential speed of the convergence that is proved in Section 31. The variance of the Energy Distance is given in Table 5.1. SAIS seems to have a high variance, as we observed that in some of the runs, the algorithm could not explore all the modes — we excluded these runs from the Energy distance plot — while CMC and MoKAs have lower variances.

Th computational time for PMH, vanilla CMC, Kids and SAIS was 1.6s, 3.2s, 58s and 16.5s respectively for a 50 iteration run of the algorithm with 10^4 particles.

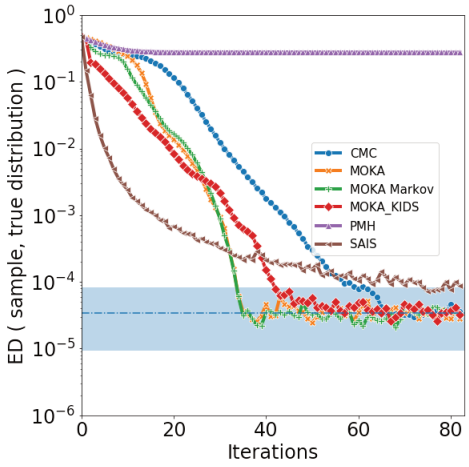


FIGURE 5.3 – Mean energy distance to a true sample on 10 repetitions of the algorithm for the Banana-shaped distribution. The dotted line represents the mean distance between two iid exact samples of size N from the target, computed over 100 independent realisations. The coloured area is the corresponding 90% prediction interval.

In Figure 5.4 we present a more in-depth analysis of the simulations. The acceptance rate of the vanilla CMC sampler is close to the one of PMH, but adaptive methods perform significantly better. This suggests that the proposal distribution that is created through our *advanced* methods is indeed closer to the target distribution, which should reflect positively on the variance of estimators. Although both MoKAs method have a very similar behaviour in terms of convergence speed, we note that they select very different weights. We believe that this is related to the L^1-like energy that is optimised in our MoKA-Markov implementation, and expect other criteria to exhibit different behaviours. Finally, we note that SAIS presents a large variance (unlike our method which produces consistent results across the runs). To remain fair with a method that can sometimes fail to converge, we only show 10 of the best among 50 runs for this algorithm.

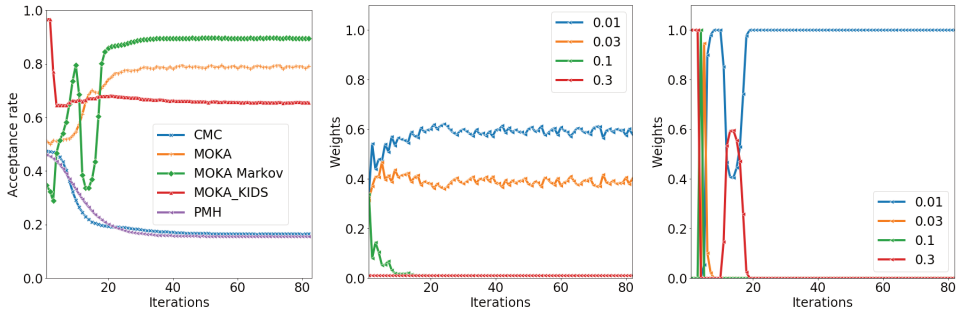


FIGURE 5.4 – Convergence analysis for the Banana-shaped distribution. From left to right: acceptance rate, evolution of weights in MoKA and in MoKA-Markov.

33.2 Moderately large dimension

Our second target is constituted of two Gaussian distributions in dimension 8, with mean $(1/2 - 1/4\sqrt{8}, 1/2 + 1/4\sqrt{8}, \ldots, 1/2 + 1/4\sqrt{8})$ and $(1/2 + 1/4\sqrt{8}, 1/2 - 1/4\sqrt{8}, \ldots, 1/2 - 1/4\sqrt{8})$, and the same variance: $\sqrt{0.05}/2$. This example is inspired from the classical example introduced in Cappé et al. [2008], Delyon and Portier [2021].

As the dimension increases, we also increase the number of particles to $N = 10^5$. We do not include an exploration proposal. We use vanilla CMC, PMH and SAIS with a single uniform proposal on a ball of radius 0.20, while the other methods based on kernel selection are given four different radii: 0.10, 0.16, 0.24 and 0.30. Due to the curse of dimensionality, the volumes of the balls that are induced by these radii are relatively smaller than in the previous experiment in dimension 2: this should reduce the exploration efficiency. We also slightly increase the number of iterations to 100.

In Figure 5.5, we present the evolution of the Energy Distance. In this example, only CMC seems to reach the smallest possible distance. MoKA-KIDS first converges quickly towards a plateau as it visits the first mode, before finding the second one. The dimension of the problem may explain the relative failure of the adaptive methods, as they rely on the proximity of particles that are more isolated from each other in higher dimensional scenarios. As before (see Figure 5.6) MoKA-Markov and MoKA do not seem to converge to the same limit: MoKA chooses a mixture of kernels while our sparsity-inducing MoKA-Markov energy promotes the use of a single kernel during each *phase* of the convergence. In this example, we remark that SAIS seems more stable than in the previous experiment.

	CMC	MoKA	MoKA-Markov	MoKA-KIDS	PMH	SAIS
Banana-shaped	2.41e-05	2.25e-05	**1.91e-05**	2.04e-05	6.75e-04	6,07e-02
Gaussian	**1.74e-06**	2.32e-06	1.75e-06	5.29e-06	3.24e-04	1.03e-05
Cauchy mixture	8.01e-05	**6.59e-05**	6.86e-05	7.40e-02	3.31e-4	8.66e-05

TABLE 5.1 – Variance of the Energy distance at the last iteration for the targets.

33.3 Numerical experiments on a Cauchy mixture

Our last target is a simple mixture of two Cauchy distributions with mean $(0.2, 0.8)$ and $(0.8, 0.2)$ and the same scale parameter 0.01. We represent in Figure 5.7 a sample and the level sets of the target distribution. This distribution has "heavy tails" which should reduce the efficiency of our uniform proposals on balls. For this example, we choose 0.1 for the radius of the unique proposals, and $0.01, 0.05, 0.1, 0.3$ for the MoKAs. The results presented in Figure 5.8 show that none of the method reaches the minimal value possible for the Energy distance. Furthermore, MoKA-KIDS dramatically fails, this is probably because of the underlying assumptions of the deconvolution algorithm we use. This last example confirms the reliability of MoKA-Markov as an efficient and secure method. The variance of the energy distance is given in Tabular 5.1.

34 Conclusion

Nonlinear MCMC samplers are appealing. They generalise more traditional methods and overcome many of their flaws. Getting back to the historical development of mathematical kinetic theory, we can advantageously simulate nonlinear Markov processes using systems of interacting particles. This versatility enables the development of a wide variety of algorithms that can tackle difficult sampling problems while remaining in a traditional Markovian framework. Although the implementation may, at first sight, seem computationally demanding, we have shown that modern GPU hardware can now enable the use of interacting particles for Monte Carlo sampling at scale.

Alongside its variants, the CMC algorithm can be implemented efficiently and leads to striking reductions in global convergence times. It relies on pairwise interactions to best leverage the information that is present in any given sample swarm, and thus make the most of each evaluation of the target distribution $\pi(x)$. CMC avoids the mixing issues of classical "one particle" methods such as Metropolis-Hastings, with a notable improvement of the convergence and mixing speed. In practice, we thus expect that the benefits of this improved "sample efficiency" will outweigh the (small) computational overhead of our method for most applications.

We note that the present contribution shares similarities with some well-known

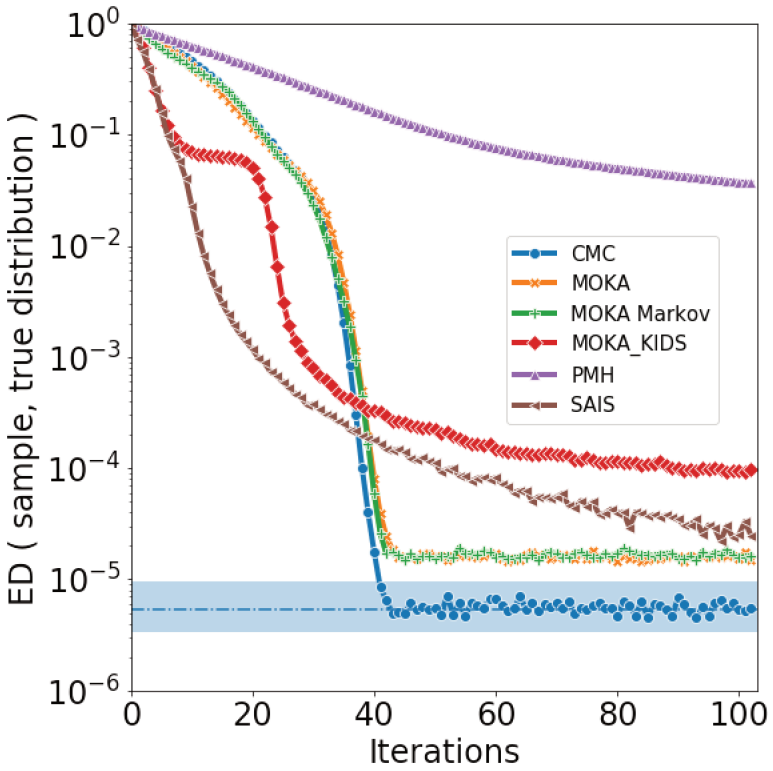


FIGURE 5.5 – Mean energy distance to a true sample on 10 repetitions of the algorithm for the eight-dimensional example. The dotted line represents the mean distance between two iid exact samples of size N from the target, computed over 100 independent realisations. The coloured area is the corresponding 90% prediction interval.

and recent nonlinear samplers, that are often based on non-Markovian importance sampling techniques. In the future, the joint development of Markovian and non-Markovian methods is likely to benefit both approaches: we may for instance improve the importance weights in SAIS-like methods as in the KIDS algorithm, or construct better proposal distributions in CMC which incorporate knowledge of (part of) the past. The theoretical study of such hybrid methods would however be challenging and require the development of new analytical tools.

Finally, one may think of extending the theoretical framework introduced here to other MCMC samplers, such as *nonlinear* PDMP samplers or *nonlinear* Langevin dynamics. This could open new problems in nonlinear analysis and statistics, both on the theoretical and computational sides.

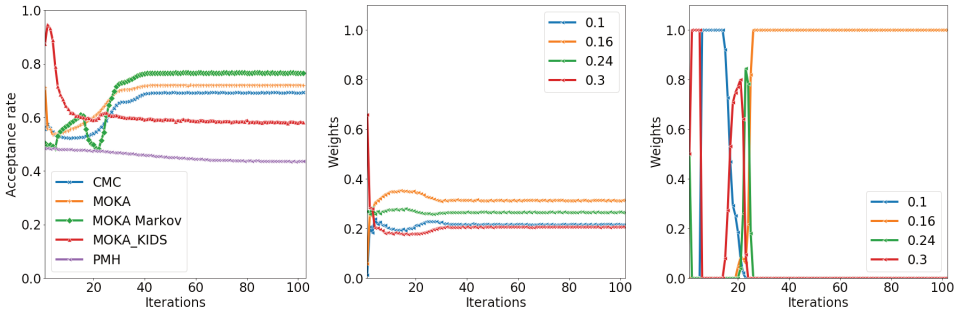


FIGURE 5.6 – Convergence analysis for the 8-dimensional Gaussian mixture. From left to right: acceptance rate, evolution of weights in MoKA and in MoKA-Markov.

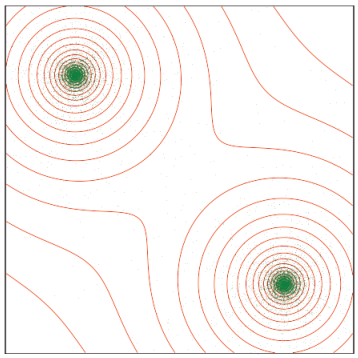


FIGURE 5.7 – True sample and levels of the mixture of Cauchy target distribution

Acknowledgments

The authors wish to thank Pierre Degond, Robin Ryder and Christian Robert for their support and useful advice. A.D. acknowledges the hospitality of the CEREMADE, Université Paris-Dauphine where part of this research was carried out. G.C. acknowledges the hospitality of the Mathematics Department at Imperial College London where part of this research was carried out.

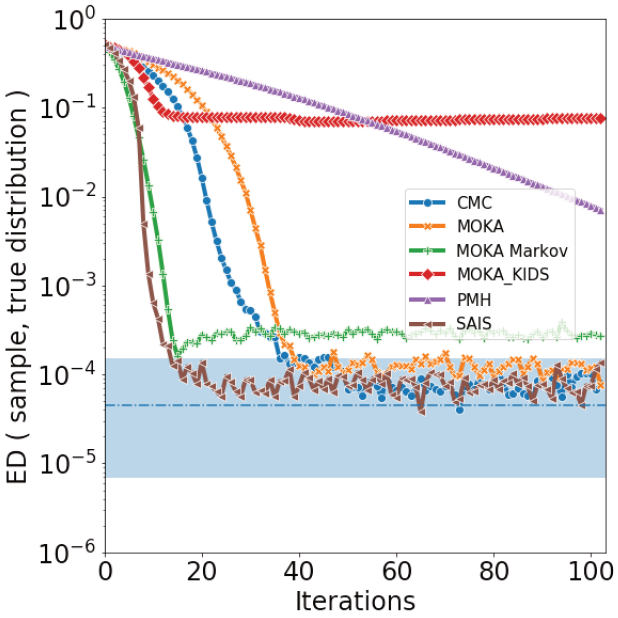


FIGURE 5.8 – Mean energy distance to a true sample on 10 repetitions of the algorithm, Cauchy mixture. The dotted line represents the mean distance between two iid exact samples, computed over 100 independent realisations, and the coloured area is the corresponding 90% prediction interval.

Index of acronyms

ABC Approximate Bayesian Computation [Marin et al., 2012];

ABC-Gibbs Approximate Bayesian Computation with Gibbs steps [Clarté et al., 2019b];

ABC-MCMC Approximate Bayesian Computation Monte Carlo Markov Chain [Marjoram et al., 2003];

BGK Bhatnagar-Gross-Krook , name of a linear operator appearing in some versions of Boltzmann equation;

BSL British Sign Language;

CMC Collective Monte Carlo;

CPU Central Processing Unit;

ESS Effective sample size, in the MCMC and SMC context [Robert and Casella, 2004];

G& K probability distribution, common example of intractable likelihood distribution;

GPU Graphical Processing Unit;

IE Indo-European;

IPA International Phonological Alphabet;

KeOps Kernel Operation, library for computing efficiently kernel interactions on GPU, using symbolic computations [Charlier et al., 2018];

KIDS Kernelised Interpolation by Deconvolution Sampling, one of the variations of CMC proposed in this work;

LSF Langue des Signes Française, main Sign language in France and some other francophone countries;

MCMC Monte Carlo by Markov Chain, general name of numerical methods that uses Markov chains to generate samples from a given distribution;

MH Metropolis Hastings algorithm Metropolis et al. [1953];

MoKA Mixture of Kernel Adaptive version, one of the CMC methods we study here;

NZSL New Zealand Sign Language;

OldLSF partially known ancestral Sign language spoken in France during the early 18th century;

PC Proto-Celtic;

PDE Partial Differential Equation;

PMC Particle Monte Carlo [Cappé et al., 2004];

PMH Parallel Metropolis Hastings, parallel implementation of several MH;

SAIS Safe Adaptive Importance Sampling [Delyon and Portier, 2021];

SMC Sequential Monte Carlo [Chopin and Papaspiliopoulos, 2020];

SMC-ABC Sequential Monte Carlo Approximate Bayesian Computation [Toni et al., 2008].

Part 6
Index of Tables and Figures

List of Figures

List of Tables

List of Algorithms

Bibliography

Natasha Abner, Carlo Geraci, Shi Yu, Jessica Lettieri, Justine Mertz, and Anah Salgat. Getting the upper hand on sign language families: Historical analysis and annotationmethods. *FEAST. Formal and Experimental Advances in Sign language Theory*, 3:17–29, 11 2020.

Christophe Andrieu, Ajay Jasra, Arnaud Doucet, and Pierre Del Moral. Non-linear Markov Chain Monte Carlo. *ESAIM: Proc.*, 19:79–84, 2007.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov Chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 72(3): 269–342, 2010.

Christophe Andrieu, Ajay Jasra, Arnaud Doucet, and Pierre Del Moral. On non-linear Markov Chain Monte Carlo. *Bernoulli*, 17(3):987–1014, 08 2011.

Barry C Arnold and S James Press. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156, 1989.

Yves Atchadé, Gersende Fort, Eric Moulines, and Pierre Priouret. Adaptive Markov Chain Monte Carlo: Theory and Methods. In David Barber, Ali Taylan Cemgil, and Silvia Chiappa, editors, *Bayesian Time Series Models*, pages 32–51. Cambridge Univ. Press., 2011.

Yves F. Atchadé and Jeffrey S. Rosenthal. On adaptive Markov Chain Monte Carlo algorithms. *Bernoulli*, 11(5):815–828, 10 2005.

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian Computation in Population Genetics. *Genetics*, 162(4):2025–2035, 2002.

Mark A. Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 2009. doi: 10.1093/biomet/asp052.

Nicola Bellomo, Pierre Degond, and Eitan Tadmor, editors. *Active Particles, Volume 1: Advances in Theory, Models, and Applications*. Birkhäuser, 2017.

Nicola Bellomo, Pierre Degond, and Eitan Tadmor, editors. *Active Particles, Volume 2: Advances in Theory, Models, and Applications*. Birkhäuser, 2019.

J. Besag. Comments on “Representations of knowledge in complex systems" by U. Grenander and MI Miller. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 56:591–592, 1994.

Prabhu Lal Bhatnagar, Eugene P Gross, and Max Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511, 1954.

François Bolley and Cédric Villani. Weighted Csiszár-Kullback-Pinsker inequalities and applications to transportation inequalities. *Ann. Fac. Sci. Toulouse Math.*, 14(3):331–352, 2005.

Luke Bornn, Pierre E Jacob, Pierre Del Moral, and Arnaud Doucet. An adaptive interacting Wang–Landau algorithm for automatic density exploration. *J. Comput. Graph. Statist.*, 22(3):749–773, 2013.

Alexandre Bouchard-Côté. Sequential Monte Carlo (SMC) for Bayesian phylogenetics. In M.-H. Chen, L. Kuo, and P. O. (eds.) Lewis, editors, *Bayesian phylogenetics: methods, algorithms, and applications*, pages 163–186. Chapman & Hall, CRC, 2014.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 110(11):4224–4229, 2013. ISSN 0027-8424. doi: 10.1073/pnas.1204678110.

Remco R Bouckaert, Claire Bowern, and Quentin D Atkinson. The origin and expansion of pama–nyungan languages across australia. *Nature ecology & evolution*, 2(4):741–749, 2018.

Claire Bowern and Quentin Atkinson. Computational phylogenetics and the internal structure of pama-nyungan. *Language*, pages 817–845, 2012.

Pierre Brémaud. *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues* . Springer, 1991.

Luis Caffarelli, Mikhail Feldman, and Robert McCann. Constructing optimal maps for Monge's transport problem as a limit of strictly convex costs. *J. Amer. Math. Soc.*, 15(1):1–26, 2002.

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P Robert. Population Monte Carlo. *J. Comput. Graph. Statist.*, 13(4):907–929, 2004.

Olivier Cappé, Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P Robert. Adaptive importance sampling in general mixture classes. *Stat. Comput.*, 18:447–459, 2008.

B.P. Carlin and T.A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman and Hall, London, 1996.

René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications I.* Springer, 2018.

Thierry Champion, Luigi De Pascale, et al. The Monge problem in $\mathbb{R}^d$. *Duke Math. J.*, 157(3):551–572, 2011.

Benjamin Charlier, Jean Feydy, and Joan Glaunes. Kernel operations on the GPU, with autodiff, without memory overflows. `http://www.kernel-operations.io`, 2018.

Nicolas Chopin and Omiros Papaspiliopoulos. *An introduction to sequential Monte Carlo.* Springer, 2020.

Grégoire Clarté, Antoine Diez, and Jean Feydy. Collective proposal distributions for nonlinear mcmc samplers: Mean-field theory and fast implementation. *arXiv preprint arXiv:1909.08988*, 2019a.

Grégoire Clarté, Christian P Robert, Robin Ryder, and Julien Stoehr. Component-wise approximate bayesian computation via gibbs-like steps. *arXiv preprint arXiv:1905.13599*, 2019b.

Neville Edgar Collinge. *The laws of Indo-european*, volume 35. John Benjamins Publishing, 1985.

Radu V Craiu, Jeffrey Rosenthal, and Chao Yang. Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *J. Amer. Statist. Assoc.*, 104(488): 1454–1466, 2009.

Katalin Csilléry, Michael G. B. Blum, Oscar E. Gaggiotti, and Olivier François. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

Bart de Boer. Self Organisation in Vowel Systems through Imitation, 1997.

Ferdinand De Saussure. *Cours de linguistique générale*, volume 1. Otto Harrassowitz Verlag, 1989.

Pierre Del Moral. Feynman-kac formulae. In *Feynman-Kac Formulae*, pages 47–93. Springer, 2004.

Pierre Del Moral. *Mean Field Simulation for Monte Carlo Integration.* Chapman and Hall/CRC, May 2013. ISBN 9781466504172.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential Mmonte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

Bernard Delyon and François Portier. Safe adaptive importance sampling: A mixture approach. *The Annals of Statistics*, 49(2):885–917, 2021.

Laurent Desvillettes, Clément Mouhot, and Cédric Villani. Celebrating Cercignani's conjecture for the Boltzmann equation. *Kinet. Relat. Models*, 4(1):277–294, 2001.

Persi Diaconis, Gilles Lebeau, and Laurent Michel. Geometric analysis for the Metropolis algorithm on Lipschitz domains. *Invent. Math.*, 185(2):239–281, 2011.

Jared Diamond and Peter Bellwood. Farmers and their languages: The first expansions. *Science (New York, N.Y.)*, 300:597–603, 05 2003. doi: 10.1126/science.1078208.

Antoine Diez. Propagation of chaos and moderate interaction for a piecewise deterministic system of geometrically enriched particles. *Electron. J. Probab.*, 25:1–38, 2020. ISSN 1083-6489. doi: 10.1214/20-EJP496.

Roland L'vovich Dobrushin. Vlasov equations. *Funct. Anal. Appl.*, 13(2):115–123, 1979.

Louis Dollo. Les lois de l'évolution. In *Bulletin de la société Belge de géologie, de paléontologie et d'hydrographie*. 1887.

R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Convergence of adaptive mixtures of importance sampling schemes. *Ann. Statist.*, 35(1):420–448, 02 2007.

Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Science & Business Media, 2013.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195(2):216–222, Sep 1987.

Ewan Dunbar and Emmanuel Dupoux. Geometric Constraints on Human Speech Sound Inventories. *Frontiers in Psychology*, 7:1061, 2016. ISSN 1664-1078. doi: 10.3389/fpsyg.2016.01061. URL `https://www.frontiersin.org/article/10.3389/fpsyg.2016.01061`.

Alison Etheridge. *Some Mathematical Models from Population Genetics*, volume 2012 of *École D'Été de Probabilités de Saint-Flour*. Springer Science & Business Media, 2011.

Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 74(3):419–474, 2012.

Paul Fearnhead, Joris Bierkens, Murray Pollock, Gareth O Roberts, et al. Piecewise deterministic Markov processes for continuous-time Monte Carlo. *Statist. Sci.*, 33(3):386–412, 2018.

Joseph Felsenstein. *Inferring phylogenies*. Sinauer associates Sunderland, MA, 1981.

Jean Feydy. *Geometric data analysis, beyond convolutions*. PhD thesis, Université Paris-Saclay, 2020.

Jean Feydy, Joan Glaunès, Benjamin Charlier, and Michael Bronstein. Fast geometric learning with symbolic matrices. *Proc. NeurIPS*, 2(4):6, 2020.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probab. Theory Related Fields*, 162(3-4): 707–738, 2015.

D. Frazier, G.M. Martin, C.P. Robert, and J. Rousseau. Asymptotic properties of approximate Bayesian computation. *Biometrika*, 105(3):593–607, 09 2018.

Alan E. Gelfand and Adrian F. M. Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85 (410):398–409, 1990.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013. ISBN 9781439840955. URL `https://books.google.fr/books?id=ZXL6AQAAQBAJ`.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6: 721–741, 1984.

Charles J Geyer. Markov chain monte carlo maximum likelihood. 1991.

Subhashis Ghosal, Jayanta K. Ghosh, and Aad W. van der Vaart. Convergence rates of posterior distributions. *Ann. Statist.*, 28(2):500–531, 04 2000.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

Carl Graham. McKean–Vlasov Itō-Skorohod equations, and nonlinear diffusions with discrete jump sets. *Stochastic Process. Appl.*, 40(1):69–82, 1992a.

Carl Graham. Nonlinear diffusion with jumps. *Ann. Inst. Henri Poincaré Probab. Stat.*, 28(3):393–402, 1992b.

R. D. Gray, A. J. Drummond, and S. J. Greenhill. Language phylogenies reveal expansion pulses and pauses in pacific settlement. *Science*, 323(5913):479–483, 2009. ISSN 0036-8075. doi: 10.1126/science.1166858.

Russell Gray and Quentin Atkinson. Language-Tree Divergence Times Support the Anatolian Theory of Indo-European Origin. *Journal of the Royal Statistical Society, series B*, 426:435–9, 12 2003.

L Greengard and V Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325 – 348, 1987.

Heikki Haario, Eero Saksman, and Johanna Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Comput. Statist.*, 14(3):375–396, 1999.

Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Hans Henrich Hock. *Principles of Historical Linguistics*. De Gruyter, 2 edition, 1991. ISBN 0899258514. doi: 10.1515/9783110219135.

Pierre-Emmanuel Jabin and Zhenfu Wang. Mean field limit and propagation of chaos for Vlasov systems with bounded forces. *J. Funct. Anal.*, 271(12):3588–3627, 2016.

Ajay Jasra, David A Stephens, and Christopher C Holmes. On population-based simulation for static inference. *Stat. Comput.*, 17(3):263–279, 2007.

Shi Jin, Lei Li, and Jian-Guo Liu. Random batch methods (RBM) for interacting particle systems. *J. Comput. Phys.*, 400:108877, 2020.

Geneviève Joly. *Précis de phonétique historique du français*. Armand Colin, 1995.

Benjamin Jourdain and Sylvie Méléard. Propagation of chaos and fluctuations for a moderate model with smooth initial data. *Ann. Inst. Henri Poincaré Probab. Stat.*, 34(6):727–766, 1998.

Ansgar Jüngel. *Entropy Methods for Diffusive Partial Differential Equations*. Springer, 2016.

Mark Kac. Foundations of kinetic theory. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 3, pages 171–197. University of California Press Berkeley and Los Angeles, California, 1956.

Jari P. Kaipio and Colin Fox. The Bayesian Framework for Inverse Problems in Heat Transfer. *Heat Transfer Engineering*, 32(9):718–753, 2011.

Avshalom Karasik and Uzy Smilansky. Computerized morphological classification of ceramics. *Journal of Archaeological Science*, 38(10):2644–2657, 2011.

L. J. Kelly and G. K. Nicholls. Lateral transfer in Stochastic Dollo models. *Annals of Applied Statistics*, 11(2):1146–1168, 2017. doi: 10.1214/17-AOAS1040.

Luke Kelly. *A stochastic Dollo model for lateral transfer*. PhD thesis, University of Oxford, 2016.

Vishnupriya Kolipakam, Fiona M. Jordan, Michael Dunn, Simon J. Greenhill, Remco Bouckaert, Russell D. Gray, and Annemarie Verkerk. A bayesian phylogenetic study of the dravidian language family. *Royal Society Open Science*, 5 (3):171504, 2018. doi: 10.1098/rsos.171504.

Athanasios Kousathanas, Christoph Leuenberger, Jonas Helfer, Mathieu Quinodoz, Matthieu Foll, and Daniel Wegmann. Likelihood-free inference in high-dimensional models. *Genetics*, 203(2):893–904, 2016. ISSN 0016-6731. doi: 10.1534/genetics.116.187567. URL `https://www.genetics.org/content/203/2/893`.

René-Joseph Lavie. Exemplar theory in linguistics: a perspective for the cognitive subject. Communication to the 11th Congress of Cognitive Linguistics, Bordeaux, 19-21 May 2005. Proceedings in press., April 2007. URL `https://halshs.archives-ouvertes.fr/halshs-00142394`.

T. Joseph W. Lazio, E B. Waltman, F D. Ghigo, Richard Fiedler, R S. Foster, and and K. J. Johnston. A Dual-Frequency, Multiyear Monitoring Program of Compact Radio Sources. *The Astrophysical Journal Supplement Series*, 136:265, December 2008. doi: 10.1086/322531.

Ryan Lepic, Carl Börstell, Gal Belsitzman, and Wendy Sandler. Taking meaning in hand: Iconic motivations in two-handed signs. *Sign Language &; Linguistics*, 19(1):37–81, 2016. ISSN 1387-9316.

Paul Lewis. A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic biology*, 50:913–25, 11 2001. doi: 10.1080/106351501753462876.

Wentao Li and Paul Fearnhead. On the asymptotic efficiency of approximate Bayesian computation estimators. *Biometrika*, 105(2):285–299, 01 2018.

D.V. Lindley and A.F.M. Smith. Bayes estimates for the linear model. *Journal of the Royal Statistical Society*, 34:1–41, 1972.

Johann-Mattis List, Philippe Lopez, and Eric Bapteste. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In *Proceedings of the Association of Computational Linguistics 2016 (Volume 2: Short Papers)*, pages 599–605, Berlin, 2016. URL `http://anthology.aclweb.org/P16-2097`.

Johann-Mattis List, Simon J. Greenhill, and Russell D. Gray. The potential of automatic word comparison for historical linguistics. *PLOS ONE*, 12(1):1–18, 01 2017. doi: 10.1371/journal.pone.0170046.

Leon B Lucy. An iterative technique for the rectification of observed distributions. *Astron. J.*, 79:745, 1974.

D.J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter. *The BUGS Book: A Practical Introduction to Bayesian Analysis*. Chapman & Hall/CRC Press, New York, 2010.

Jean-Michel Marin, Pierre Pudlo, Christian P. Robert, and Robin J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6): 1167–1180, November 2012. ISSN 1573-1375. doi: 10.1007/s11222-011-9288-2. URL `https://doi.org/10.1007/s11222-011-9288-2`.

Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

André Martinet. *Économie des changements phonétiques: traité de phonologie diachronique*, volume 10. editions A. Francke, 1970.

MATLAB. The Mathworks, Inc., Natick, Massachusetts, 2017.

Henry P McKean. A class of Markov processes associated with nonlinear parabolic equations. *Proc. Nat. Acad. Sci.*, 56(6):1907, 1966.

Henry P McKean. Propagation of chaos for a class of non-linear parabolic equations. *Stochastic Differential Equations (Lecture Series in Differential Equations, Session 7, Catholic Univ., 1967)*, pages 41–57, 1967.

Sylvie Méléard. Asymptotic behaviour of some interacting particle systems; McKean–Vlasov and Boltzmann models. In *Probabilistic models for nonlinear partial differential equations*, pages 42–95. Springer, 1996.

Kerrie L Mengersen and Richard L Tweedie. Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Statist.*, 24(1):101–121, 1996.

Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo method. *J. Amer. Statist. Assoc.*, 44(247):335–341, 1949.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114.

Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993.

Matthew T. Moores, Christopher C. Drovandi, Kerrie Mengersen, and Christian P. Robert. Pre-processing for approximate Bayesian computation in image analysis. *Statistics and Computing*, 25(1):23–33, 2015.

Frank Natterer and Frank Wübbeling. *Mathematical Methods in Image Reconstruction*, volume 5 of *SIAM Monographs on Mathematical Modeling and Computation*. Society for Industrial and Applied Mathematics, 2001.

Peter Neal. Efficient likelihood-free Bayesian Computation for household epidemics. *Statistics and Computing*, 22(6):1239–1256, Nov 2012.

Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

G. K. Nicholls and R. D. Gray. Dated ancestral trees from binary trait data and its application to the diversification of languages. *ArXiv e-prints*, November 2007.

E. Nummelin. A splitting technique for Harris recurrent chains. *Zeit. Warsch. Verv. Gebiete*, 43:309–318, 1978.

Karl Oelschläger. A law of large numbers for moderately interacting diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 69 (2):279–322, 1985.

Mark Pagel. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 255(1342):37–45, 1994.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Proceedings of Neural Information Processing Systems*, 2017a.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017b.

Porphyrios. *Isagoge*. c. 268.

Justin M. Power, Guido Grimm, and Johann-Mattis List. Evolutionary dynamics in the dispersal of sign languages. *Royal Society Open Science*, 7(1):1–30, 2020. doi: https://doi.org/10.1098/rsos.191100.

Dennis Prangle. gk: An R package for the g-and-k and generalised g-and-h distributions. *arXiv preprint arXiv:1706.06889*, 2017.

Taraka Rama. Three tree priors and five datasets: A study of indo-european phylogenetics. *Language Dynamics and Change*, 8:182–218, 01 2018a. doi: 10.1163/22105832-00802005.

Taraka Rama. Similarity dependent Chinese restaurant process for cognate identification in multilingual wordlists. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 271–281, Brussels, Belgium, October 2018b. Association for Computational Linguistics. doi: 10.18653/v1/K18-1027.

Louis Raynal, Jean-Michel Marin, Pierre Pudlo, Mathieu Ribatet, Christian P. Robert, and Arnaud Estoup. ABC random forests for Bayesian parameter inference. *Bioinformatics*, 35(10):1720–1728, 2019. doi: 10.1093/bioinformatics/bty867. URL `http://dx.doi.org/10.1093/bioinformatics/bty867`.

William Hadley Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62(1):55–59, Jan 1972.

Don Ringe, Tandy Warnow, Ann Taylor, James Clackson, Sean Crist, Joe Eska, Henry Hoenigswald, John Penney, and Anthony Warner. Indoeuropean and computational cladistics. *Transactions of the Philological Society Volume*, 1001, 01 2002.

Maria L Rizzo and Gábor J Székely. Energy distance. *Wiley interdisciplinary reviews: Computational Statistics*, 8(1):27–38, 2016.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Christian P. Robert. *The Bayesian Choice*. Spinger, 2007.

Christian P Robert and George Casella. The metropolis—hastings algorithm. In *Monte Carlo Statistical Methods*, pages 231–283. Springer, 1999.

Christian P Robert and George Casella. The metropolis—hastings algorithm. In *Monte Carlo statistical methods*, pages 267–320. Springer, 2004.

Gareth O Roberts and Jeffrey S Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

GS Rodrigues, David J Nott, and SA Sisson. Likelihood-free approximate gibbs sampling. *arXiv preprint arXiv:1906.04347*, 2019.

V Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Computat. Phys.*, 60(2):187 – 207, 1985.

Malcolm Ross et al. *Proto Oceanic and the Austronesian languages of western Melanesia*. Dept. of Linguistics, Research School of Pacific Studies, The Australian . . . , 1988.

Robin J. Ryder. *Phylogenetic Models of Language Diversification*. PhD thesis, Oxford University, 2010.

Robin J. Ryder and Geoff K. Nicholls. Missing data in a stochastic Dollo model for cognate data, and its application to the dating of Proto-Indo-European. *ArXiv e-prints*, August 2009.

Laurent Sagart, Guillaume Jacques, Yunfan Lai, Robin J. Ryder, Valentin Thouzeau, Simon J. Greenhill, and Johann-Mattis List. Dated language phylogenies shed light on the ancestry of sino-tibetan. *Proceedings of the National Academy of Sciences*, 116(21):10317–10322, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1817972116.

Christian Schmeiser. *Entropy methods*, 2018. `https://homepage.univie.ac.at/christian.schmeiser/Entropy-course.pdf`.

Hersir Sigurgeirsson, Andrew Stuart, and Wing-Lok Wan. Algorithms for particle-field simulations with collisions. *J. Comput. Phys.*, 172(2):766 – 807, 2001.

Scott Sisson, Y. Fan, and Marc Beaumont, editors. *Handbook of Approximate Bayesian Computation*. New York: Chapman & Hall/CRC, 2018.

Morris Swadesh. Lexico-statistic dating of prehistoric ethnic contacts: with special reference to north american indians and eskimos. *Proceedings of the American philosophical society*, 96(4):452–463, 1952.

Robert H Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Physical review letters*, 57(21):2607, 1986.

Saifuddin Syed, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Non-reversible parallel tempering: A scalable highly parallel mcmc scheme. *arXiv preprint arXiv:1905.02939*, 2019.

Alain-Sol Sznitman. Topics in propagation of chaos. In *Éc. Été Probab. St.-Flour XIX—1989*, pages 165–251. Springer, 1991.

Simon Tavaré, David J. Balding, Robert C. Griffiths, and Peter Donnelly. Inferring Coalescence Times From DNA Sequence Data. *Genetics*, 145(2):505–518, 1997.

R Core Team et al. R: A language and environment for statistical computing. 2013.

Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P. H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6 (31):187–202, 2008.

Brandon Turner and Trisha Van Zandt. Hierarchical approximate bayesian computation. *Psychometrika*, 79, 12 2013. doi: 10.1007/s11336-013-9381-x.

Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30, 2011.

Paul Vanetti, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Piecewise-Deterministic Markov Chain Monte Carlo. *arXiv preprint arXiv:1707.05296*, 2017.

Cédric Villani. Cercignani's conjecture is sometimes true and always almost true. *Comm. Math. Phys.*, 234(3):455–490, 2003.

Cédric Villani. H-Theorem and beyond: Boltzmann's entropy in today's mathematics. In Giovanni Gallavotti, Wolfgang l. Reiter, and Jakob Yngvason, editors, *Boltzmann's Legacy*, pages 129–144. European Mathematical Society, 2008.

Liangliang Wang, Alexandre Bouchard-Côté, and Arnaud Doucet. Bayesian phylogenetic inference using a combinatorial sequential monte carlo method. *Journal of the American Statistical Association*, 110(512):1362–1374, 2015.

Liangliang Wang, Shijia Wang, and Alexandre Bouchard-Côté. An Annealed Sequential Monte Carlo Method for Bayesian Phylogenetics. *Systematic Biology*, 69(1):155–183, 06 2019. ISSN 1063-5157. doi: 10.1093/sysbio/syz028. URL `https://doi.org/10.1093/sysbio/syz028`.

Tandy Warnow. *Computational phylogenetics: an introduction to designing methods for phylogeny estimation*. Cambridge University Press, 2017.

Richard Wilkinson, Michael Steiper, Christophe Soligo, Robert Martin, Ziheng Yang, and Simon Tavaré. Dating primate divergences through an integrated analysis of palaeontological and molecular data. *Syst. Biol.*, 60(1):16–31, 2011. doi: 10.1093/sysbio/syq054.

James Woodward. Explanation and invariance in the special sciences. *British Journal for the Philosophy of Science*, 51(2):197–254, 2000. doi: 10.1093/bjps/51.2.197.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.

Ziheng Yang. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods. *Journal of Molecular evolution*, 39(3):306–314, 1994.

Gaston Zink. *Phonétique historique du français*. Presses universitaires de France, 1986.

RÉSUMÉ

Ce travail est la concaténation de trois parties, ayant pour point commun de porter sur les statistiques bayesiennes.
La première partie concerne les méthodes bayesiennes d'inférence de phylogénies, avec une application à l'histoire des langues des Signes. Nous développons un modèle pour des données matricielles, dont lignes et colonnes sont corrélées ; ces données peuvent représenter des traits socio-culturels, phénotypiques, ou, comme dans notre cas, des données lexicales. Nous montrons comment calculer la vraisemblance de ce modèle et proposons des méthodes numériques pour échantillonner depuis le posterior associé, basées sur un Monte Carlo sequentiel associé à un tempering exotique. Les résultats sur données simulées sont plus que satisfaisants, tandis que les résultats sur données réelles apportent des éléments de réponses aux questions des linguistes.
La deuxième partie traite des méthodes bayésiennes approchées. Ces méthodes s'utilisent lorsque les vraisemblances sont intractables, elles sont, hélas, particulièrement sensibles au fléau de la dimension, requerrant des ressources exponentiellement élevées à mesure que la dimension croit. Pour résoudre ce problème, nous explorons une version à la Gibbs des méthodes ABC traditionnelles, où l'on met à jour sequentiellement les coordonnées des paramètres selon des lois conditionnelles approchées reposant sur des statistiques résumées de dimension moindre. Bien qu'il ne soit pas possible d'utiliser des méthodes classiques pour étudier cette méthode, nous avons été capables de montrer sa convergence vers une mesure stationnaire dépourvue de forme explicite. Les expériences démontrent une efficacité particulière par rapport aux méthodes standard.
La troisième partie est dédiée aux méthodes numériques particulaires. Au cours des dernières décénies, des méthodes MCMC *non linéaires* ont été dévelopées ; bien qu'attirantes par leur vitesse de convergence et leur efficacité, leur implémentation et étude théorique reste problématique. Nous introduisons une large classe de méthodes non linéaires qu'il est possible d'étudier à l'aide de limites champ-moyen de particules en interaction. L'implémentation que l'on propose repose sur le calcul parallèle sur GPU.

MOTS CLÉS

Statistiques Bayesiennes, Méthodes Bayesiennes approchées, Sequential Monte Carlo, Phylogénies

ABSTRACT

This work is the concatenation of three papers, all revolving around Bayesian statistics.
The first one concerns Bayesian phylogenetical inference with application to historical linguistics of Sign Languages. We develop a model for matricial datasets where lines and columns evolve jointly, this can represent vocabulary datasets or even socio-cultural traits. We are able to compute the likelihood associated with this model, and to sample from the posterior by using Sequential Monte Carlo methods with exotic tempering. The results on simulated datasets are quite satisfactory and the results on real dataset confront the hypothesis of the linguists.
The second deals with approximate Bayesian computation. These methods are useful for generative models with intractable likelihoods. These methods are however sensitive to the dimension of the parameter space, requiring exponentially increasing resources as this dimension grows. To tackle this difficulty, we explore a Gibbs version of the ABC approach that runs component-wise approximate Bayesian computation steps aimed at the corresponding conditional posterior distributions. Each of these ABS is based on summary statistics of reduced dimension. While lacking the standard justifications for the Gibbs sampler, the resulting Markov chain is shown to converge in distribution under some partial independence conditions. The associated stationary distribution can further be shown to be close to the true posterior distribution and some hierarchical versions of the proposed mechanism enjoy a closed form limiting distribution. Experiments also demonstrate the gain in efficiency brought by the Gibbs version over the standard solutions.
The third is dedicated to interacting particle methods. Over the last decades, various ``non-linear'' MCMC methods have arisen. While appealing for their convergence speed and efficiency, their practical implementation and theoretical study remain challenging. We introduce a large class of non-linear samplers that can be studied and simulated as the mean-field limit of a system of interacting particles. The practical implementation we propose leverages the computational power of modern hardware (GPU).

KEYWORDS

Bayesian Statstics, Approximate Bayesian Computation, Sequential Monte Carlo, Phylogenies