Point process models of retinal ganglion cell spike trains

Cyrille Rossant

Master's internship

from March to August 2008,
under the supervision of Michael Berry,
in collaboration with Kolia Sadeghi,
Department of Molecular Biology,
Princeton University, Princeton, USA.

Contents

Fiche de synthèse

Le contexte général

La *rétine* est un tissu nerveux au fond de l'œil des vertébrés, qui convertit le signal lumineux en brèves impulsions électriques stéréotypées appelées *potentiels d'action*. Ces derniers sont ensuite transmis au cerveau par l'intermédiaire du nerf optique. Le projet de recherche de l'équipe dirigée par Michael Berry est de comprendre les processus algorithmiques implémentés par la rétine qui permettent l'encodage de l'information visuelle. Pour cela, l'équipe a développé une technique expérimentale permettant d'enregistrer l'activité électrique de la rétine tout en projetant des images à sa surface.

Les travaux effectués jusqu'à présent sur la rétine ont montré que, bien qu'étant la toute première étape de la vision biologique, des traitements complexes sur l'information visuelle y sont effectués. De plus, une structure statistique complexe est présente dans les signaux de sortie de la rétine, et porte probablement beaucoup d'information sur l'image. Comprendre cette structure et son rôle est l'un des enjeux du projet de recherche.

Le problème étudié

Ce stage, réalisé en collaboration avec Kolia Sadeghi (doctorant dans l'équipe), a consisté à appliquer la théorie des *processus ponctuels* à l'étude des propriétés statistiques des trains de potentiels d'action des cellules de sortie de la rétine (*cellules ganglionnaires*). L'intérêt de cette étude vient du fait que la structure statistique observée dans ces signaux (en particulier les *corrélations* entre les cellules) a vraisemblablement un rôle important dans l'encodage de l'information.

Ce problème a été beaucoup étudié au cours des dernières années. En particulier, des outils issus de la physique statistique (modèle d'Ising par exemple) ont été appliqués avec succès à l'étude statistique du code neural. Il a ainsi été montré que les corrélations des paires de cellules capturaient une très grande partie de la structure statistique de ces signaux.

La théorie des processus ponctuels, qui est la formalisation mathématique des trains de potentiels d'action, a quant à elle été encore peu utilisée. Les modèles développés jusqu'à présent ne l'étaient que dans un cadre restreint (principalement des processus de renouvellement). Durant ce stage, nous avons proposé une autre manière d'appliquer cette théorie au problème posé, qui permet d'exprimer de manière élégante une très grande variété de modèles.

La contribution proposée

Au cours de stage, j'ai 1/ étudié un outil théorique pour définir et manipuler des modèles multidimensionnels de processus ponctuels, 2/ élaboré et implémenté informatiquement un outil statistique pour comparer les modèles aux données, 3/ trouvé un modèle intéressant à l'aide de ces outils.

L'outil choisi pour manipuler des processus ponctuels est la *fonction génératrice*. À notre connaissance, cet outil n'a encore jamais été utilisé pour de telles applications, malgré de nombreux avantages. L'outil statistique utilisé pour l'inférence des modèles à partir des données expérimentales n'est pas l'algorithme du maximum de vraisemblance pour des raisons de limitations en puissance de calcul. Nous avons plutôt choisi une variante heuristique appelée *algorithme du maximum de pseudo-vraisemblance*. Ce dernier se place dans un cadre discret, contrairement à la définition du modèle qui se déroule dans un cadre continu. Le passage du continu au discret s'effectue simplement grâce aux fonctions génératrices. J'ai implémenté cet algorithme en *Matlab* pour tester les modèles vis-à-vis des données. Les outils ainsi développés sont tout à fait généraux et peuvent potentiellement admettre des applications en dehors du cadre des neurosciences computationnelles.

Dans un second temps, nous avons défini à l'aide de ces outils un modèle particulier de processus ponctuel multidimensionnel pour les trains de potentiels d'action des cellules ganglionnaires de la rétine. Les données expérimentales que nous avions provenaient du laboratoire. La démarche utilisée pour obtenir ce modèle a d'abord été de trouver, dans un cadre discret, une distribution de probabilité intéressante pour les données. Nous l'avons alors généralisée au cadre continu, puis améliorée pour obtenir de meilleurs résultats.

Les arguments en faveur de sa validité

Pour tester notre modèle, nous avons inféré ses paramètres à partir des données, à l'aide de l'algorithme du maximum de pseudo-vraisemblance. Nous avons ensuite vérifié que différentes quantités découlant du modèle étaient proches des valeurs expérimentales : distributions de probabilité unidimensionnelles (marginales), covariances des paires de cellules... Les premiers résultats obtenus montrent que pour une certaine partie des cellules, les marginales et les covariances sont très bien capturées. Les résultats sont cependant un peu moins bons pour les autres cellules. Une des raisons principales est que le modèle actuel ne capture pas les corrélations négatives. Ce défaut devra être résolu après la fin de ce stage.

Il reste encore beaucoup à faire pour étudier l'influence de tous les paramètres du modèle sur les résultats et pour améliorer le modèle. Il faudrait aussi confronter le modèle avec d'autres données expérimentales. De plus, les résultats sont à considérer en tenant compte du fait que le passage du continu au discret implique une perte d'information, et que l'algorithme d'inférence n'est pas parfait. Améliorer cet outil statistique pourrait nous aider à obtenir des résultats plus intéressants.

Le bilan et les perspectives

Le travail réalisé montre que la fonction génératrice, un outil simple, puissant et peu connu dans le domaine, pouvait être appliqué avec succès à l'élaboration de modèles de processus ponctuels. D'autre part, nous avons pu trouver à l'aide de cette méthode un modèle intéressant pour l'étude des propriétés statistiques des trains de potentiels d'action des cellules ganglionnaires de la rétine. Ce travail constitue donc une première étape pour l'application de cet outil au problème posé, et fournit une approche puissante et différente de celles déjà explorées.

Il s'agit maintenant de s'aider de cet outil et du modèle trouvé pour tenter de comprendre les implications computationnelles de la structure statistique du code neural de la rétine. Autrement dit, ce modèle permet-il d'expliquer comment le corps genouillé latéral et le cortex visuel utilisent les propriétés statistiques des signaux rétiniens pour en déduire des propriétés sur l'image ?

Résumé en français

Remarques sur ce rapport

Ce rapport est en anglais du fait que ce stage s'est déroulé aux États-Unis et que mon directeur de stage est américain. De plus, le contenu utile de ce rapport (sans les annexes, la page de garde et les divers compléments) dépasse légèrement la limite imposée. Cela vient du fait que je me suis plus longuement attardé sur la présentation du contexte scientifique. La raison de ce choix est que je souhaitais rendre compte non seulement de mon travail sur le plan mathématique, mais aussi de ses motivations et de son intérêt en neurosciences computationnelles. Il m'a donc semblé important d'en donner brièvement les fondations pour faciliter le travail des lecteurs étrangers à ce domaine de recherche.

Dans ce résumé en français, le contexte est introduit très brièvement, et les travaux précédents sont rapidement mentionnés. La suite de ce résumé contient l'essentiel de mon travail sur les outils théoriques pour étudier les trains de potentiels d'action, et leur utilisation dans la recherche d'un bon modèle pour des données provenant de la rétine.

Introduction

Le laboratoire

Ce stage s'est déroulé dans le département de biologie moléculaire à l'Université de Princeton, aux États-Unis. J'étais intégré dans l'équipe de recherche dirigée par Michael Berry, un physicien de formation qui s'intéresse aux neurosciences computationnelles, et plus particulièrement aux processus algorithmiques intervenant dans la rétine.

La rétine

La rétine est un tissu nerveux au fond de l'oeil des vertébrés, qui convertit le signal lumineux (convergeant sur la surface de la rétine grâce au cristallin) en brèves impulsions électriques stéréotypées (*potentiels d'action*, ou *spikes*). Ces potentiels d'action, considérés instantanés dans le cadre de ce stage, sont ensuite transmis au cerveau par l'intermédiaire du nerf optique. La rétine est donc la toute première étape de la vision biologique. Le reste du traitement de l'information visuelle se fait dans le thalamus et surtout dans le cortex cérébral. Le projet général de recherche du laboratoire est donc de comprendre les processus algorithmiques qui sont implémentés par la rétine et qui permettent l'encodage de l'information visuelle.

Le protocole expérimental

Les chercheurs de l'équipe mènent des expériences qui consistent à projeter des images ou des vidéos sur des rétines extraites de salamandres, tout en enregistrant l'activité électrique simultanée de dizaines de cellules de sortie de la rétine (*cellules ganglionnaires*) à l'aide d'une grille multi-électrodes. Un travail statistique et de modélisation s'appuyant sur les données est alors conduit pour tenter de comprendre le fonctionnement de la rétine.

Statistiques des trains de potentiels d'action des cellules ganglionnaires

Les potentiels d'action émis par les cellules ganglionnaires présentent expérimentalement une part d'aléatoire (bruit) : le cadre théorique d'étude est donc probabiliste. Il a été constaté que les signaux de sortie de la rétine présentent une structure statistique complexe, qui joue vraisemblablement un rôle important dans l'encodage. Il a ainsi été montré que les corrélations entre cellules sont essentielles dans le code neural. Le sujet particulier sur lequel j'ai travaillé était donc l'élaboration de modèles statistiques des trains de potentiels d'action des cellules ganglionnaires capturant autant que possible les corrélations entre cellules.

Contributions

Ce stage, réalisé en collaboration avec Kolia Sadeghi (doctorant dans l'équipe), a consisté à appliquer la théorie des *processus ponctuels* à l'étude statistique des trains de potentiels d'action des cellules ganglionnaires de la rétine. J'ai plus particulièrement 1/ étudié un outil théorique pour définir et manipuler des modèles multidimensionnels de processus ponctuels, 2/ élaboré et implémenté informatiquement un outil statistique pour comparer les modèles aux données, 3/ trouvé un modèle intéressant à l'aide de ces outils.

L'étude statistique des trains de potentiels d'action à l'aide de la théorie des processus ponctuels

Processus ponctuels

Un *processus ponctuel réel* est une suite aléatoire croissante d'instants $\{t_i\} \in \mathbb{R}$. C'est la formalisation mathématique des trains de potentiels d'action. Les travaux précédents utilisant la théorie des processus ponctuels pour l'étude statistique du code neural utilisent l'*intensité conditionnelle* et la distribution des intervalles inter-spikes. Les modèles utilisés sont surtout des processus de renouvellement. L'inférence statistique utilise la méthode du maximum de vraisemblance, puisqu'un modèle de processus ponctuel donné par son intensité conditionnelle admet une expression analytique pour sa fonction de vraisemblance. La qualité statistique (*goodness-of-fit*) d'un modèle est donnée visuellement par un graphique de Kolmogorov-Smirnov, ou quantitativement par le coefficient d'Akaike. Un exemple de modèle intéressant est celui qui admet une distribution gaussienne inverse comme distribution d'inter-spikes.

Intérêt d'une autre approche

L'inconvénient de cette approche est qu'elle ne permet d'exprimer principalement que des processus de renouvellement. Ces processus ne forment qu'une classe restreinte de modèles, et ne permettent pas de capturer efficacement les corrélations à travers le temps et les cellules. Durant ce stage, nous avons utilisé une autre approche, permettant de définir des processus ponctuels en utilisant la *fonction génératrice.* Cet outil permet de définir de manière simple et naturelle une très grande variété de modèles. Nous n'avons pas connaissance de travaux antérieurs appliquant cet outil a des questions de neurosciences.

Fonction génératrice d'un processus ponctuel

La fonction génératrice d'un processus ponctuel $\{t_i\}$ est définie par (h est une fonction régulière $\mathbb{R} \to [\,0, 1\,]$, valant 1 en dehors d'un intervalle borné) :

$$G(h) = \mathbb{E}\left(\prod_i h(t_i)\right).$$

C'est la généralisation "continue" de la fonction génératrice d'une distribution de probabilité discrète. En effet, la fonction génératrice appliquée à une fonction h constante par morceaux donne lieu à la fonction génératrice de la distribution discrète multidimensionnelle du nombre de potentiels d'action dans les différents intervalles sur lesquels h est constante. La fonction génératrice est intéressante pour des applications pratiques parce que des modèles probabilistes complexes peuvent admettre des fonctions génératrices relativement simples. De plus, il est facile d'obtenir des moments et de marginaliser à partir de la fonction génératrice.

Il est possible d'obtenir numériquement une approximation des N^D premières valeurs d'une distribution discrète de dimension D à partir de la fonction génératrice, à l'aide d'une transformée discrète de Fourier. Cela est utile pour l'inférence des paramètres d'un modèle de processus ponctuel défini par sa fonction génératrice.

Algorithme du maximum de pseudo-vraisemblance

Dans ce stage, l'élaboration d'un modèle s'est faite dans le cadre continu, à l'aide des fonctions génératrices des processus ponctuels. En revanche, l'inférence et la confrontation aux données s'est faite dans un cadre discret, en considérant les distributions de probabilité du nombre de potentiels d'action dans des fenêtres temporelles fixées. Le passage du continu au discret s'effectue très facilement à l'aide des fonctions génératrices (voir le paragraphe précédent).

Les distributions discrètes multidimensionnelles considérées sont donc le nombre de potentiels d'action dans différentes fenêtres temporelles de plusieurs cellules. Le nombre de dimension D de ces distributions (le nombre total de fenêtres) est assez important, ce qui complique l'inférence. En particulier, l'algorithme du maximum de vraisemblance peut être implémenté avec un modèle défini par sa fonction génératrice, mais sa complexité est importante à cause du calcul de la fonction de vraisemblance. Cette dernière fait intervenir le calcul des N^D premières valeurs de la distribution à partir de la fonction génératrice : c'est essentiellement une transformée de Fourier rapide en D dimensions. Or, pour les applications qui nous intéressent, D peut être grand (de l'ordre de la centaine), rendant impraticable l'algorithme.

Une variante pour l'inférence a donc été imaginée : il s'agit de maximiser une fonction de *pseudo-vraisemblance*, qui est la somme des fonctions de vraisemblance de plusieurs marginales unidimensionnelles (ou bidimensionnelles). Une telle marginale est en pratique la somme sur un sous-ensemble des composantes de la distribution. Autrement dit, c'est le nombre de potentiels d'action dans la *réunion* de plusieurs fenêtres temporelles de différentes cellules. Le calcul de la fonction de pseudo-vraisemblance est plus rapide, puisqu'il n'implique des FFT qu'en 1 ou 2 dimensions.

Cet algorithme a été implémenté en *Matlab* durant ce stage : il prend en entrée un modèle de distribution de probabilité multidimensionnelle défini par sa fonction génératrice, les différentes marginales choisies, ainsi que les données. Il retourne les paramètres du modèle qui minimisent la fonction de pseudo-vraisemblance. Il utilise une méthode classique d'optimisation implémentée d'origine dans *Matlab*.

Le modèle PNMP : élaboration et inférence

Les outils décrits précédemment nous ont servi à élaborer un modèle particulier pour les trains de potentiels d'action des cellules ganglionnaires de la rétine, et à le confronter aux données expérimentales obtenues dans le laboratoire.

Méthodologie

Le parcours utilisé pour trouver un modèle satisfaisant était le suivant. D'abord, les distributions unidimensionnelles expérimentales du nombre de potentiels d'action dans des réunions de fenêtres temporelles sur plusieurs cellules étaient observées graphiquement. Nous avons tenté d'appliquer différentes distributions de probabilité classiques à ces données, à l'aide de la méthode des moments ou du maximum de vraisemblance (c'est faisable rapidement car le cadre est unidimensionnel à cette étape).

Une fois une bonne distribution trouvée, nous l'avons généralisée en un processus ponctuel multidimensionnel (que nous avons appelé le *modèle NM*). Nous avons alors lancé l'algorithme du maximum de pseudo-vraisemblance pour inférer les paramètres du modèle à partir des données. Les marginales unidimensionnelles choisies étaient soit le nombre total de potentiels d'action dans une fenêtre étroite sur un grand nombre de cellules, soit le nombre de potentiels d'action dans une large fenêtre temporelle sur des cellules individuelles. Dans le premier cas, les résultats étaient très intéressants, laissant penser que le modèle parvenait bien à capturer les corrélations entre cellules. Dans le second cas en revanche, les distributions théoriques et expérimentales ne correspondaient pas du tout.

Nous avons alors amélioré le modèle à l'aide de la technique des *cluster processes*, conduisant au modèle PNMP. C'est ce modèle que nous avons retenu pour conduire différents tests à l'aide de l'algorithme d'inférence. Nous avons en particulier regardé si le modèle parvenait à capturer non seulement les marginales unidimensionnelles du nombre de potentiels d'action dans différentes fenêtres, mais aussi les corrélations entre cellules.

La distribution de Poisson

La première distribution unidimensionnelle que nous avons tenté d'appliquer aux distributions du nombre de potentiels d'action dans différentes fenêtres était la distribution de Poisson. Le processus ponctuel correspondant est le processus de Poisson, qui est le plus simple et le mieux compris. C'est aussi celui qui est utilisé en majorité pour modéliser des trains de potentiels d'action. C'est un processus totalement aléatoire : seule la *fréquence de décharge* (densité locale moyenne du nombre de potentiels d'action, c'est une fonction déterministe) est fournie au modèle. Les émissions de potentiels d'action respectent cette moyenne mais sont totalement aléatoires en dehors de cette contrainte. La légitimité de ce modèle pour représenter des trains de potentiels d'action dans certains cas est contestée par les chercheurs, car le processus de Poisson ne permet pas de capturer les corrélations et la synchronisation qui sont souvent observées expérimentalement, et qui ont vraisemblablement un rôle computationnel important.

Les résultats obtenus montrent clairement que le processus de Poisson n'est pas du tout un bon modèle pour nos données. Cela confirme le fait qu'il est réellement nécessaire de trouver un meilleur modèle de processus ponctuel.

La distribution *negative multinomial*

La distribution géométrique, et surtout sa généralisation, la distribution *negative multinomial*, donnent des résultats beaucoup plus intéressants. La fonction génératrice de cette dernière est, en une dimension :

$$G(x) = \left(\frac{1}{1 + K(1-x)} \right)^{\kappa},$$

où $K, \kappa > 0$ sont des paramètres. Cette distribution capture très bien le nombre total de potentiels d'action dans une fenêtre étroite de temps (environ 20 ms), sur un grand nombre de cellules simultanément.

Le processus NM

Nous avons donc défini, à l'aide des fonctions génératrices, un processus ponctuel multidimensionnel généralisant cette distribution. Le processus ainsi obtenu, que nous avons appelé *processus NM*, est défini par la fonction génératrice suivante :

$$G(h_1, \dots, h_N) = \left(\frac{1}{1 + \sum_i \int_{\mathbb{R}} K_i(t)(1 - h_i(t))\,\mathrm{d}t} \right)^{\kappa},$$

où les $K_i : \mathbb{R} \to [\,0, 1\,]$ sont des fonctions régulières. Ce modèle donne cependant de mauvais résultats sur une large fenêtre temporelle (environ 100 ms), avec une seule cellule.

Le processus PNMP

Nous avons essayé d'utiliser la technique des *cluster processes* pour améliorer le modèle. Cette technique consiste à composer deux processus ponctuels, le premier définissant des *clusters*, le second générant les potentiels d'action au sein de chaque *cluster*. Cette opération s'effectue facilement avec les fonctions génératrices. En composant le modèle NM avec deux processus de Poisson, nous avons obtenu et implémenté le modèle PNMP. Sa fonction génératrice est :

$$G(h_1, \dots, h_n) = \exp\left(\lambda \int \left(\left(\frac{1}{1 + \sum_i \int K_i(t) \left(1 - \exp\left(\int \mu_i(u)(h_i(u - t - v) - 1)\,\mathrm{d}u\right)\right)\,\mathrm{d}t} \right)^{\kappa} - 1 \right) \mathrm{d}v \right).$$

C'est ce modèle qui nous a semblé intéressant et sur lequel nous avons travaillé pour le comparer aux données, et pour voir s'il capturait les corrélations dans le temps et entre cellules.

Résultats et conclusion

Protocole

Nous avons lancé l'algorithme du maximum de pseudo-vraisemblance sur toutes les paires et tous les triplets de cellules. Ayant 40 cellules dans les enregistrements expérimentaux mis à notre disposition, il y avait donc 780 paires et 9880 triplets à tester. Les marginales choisies pour l'algorithme étaient le nombre de potentiels d'action pour chaque cellule dans toute la grille (de 100 ms). Nous y avons ajouté les distributions jointes des paires de ces marginales pour espérer capturer les corrélations entre cellules.

Résultats

Les résultats sont très variables selon les paires et les triplets. Nous avons tenté de les trier selon un score général rendant compte de l'adéquation du modèle aux données (le maximum des erreurs relatives des variances et covariances des marginales de chaque cellule). D'autres score plus appropriés pourraient bien sûr être envisagés.

Pour une petite partie des paires et des triplets, les résultats sont très satisfaisants. Les marginales unidimensionnelles (nombre de potentiels d'action dans une fenêtre de 100 ms) empiriques et théoriques sont visuellement assez proches, tandis que les covariances sont très bien capturées. Pour les autres paires, les résultats sont moins bons, surtout en ce qui concerne les covariances. Il semble que cela soit surtout dû à des covariances empiriques négatives que le modèle n'est pas capable de capturer. Il s'agit là d'une faille important du modèle PNMP qu'il faudrait résoudre dans le futur.

Conclusion

Au final, ce modèle donne de bon résultats dans une partie non négligeable de cas et semble être un bon point de départ pour trouver un bon modèle. Plusieurs améliorations sont envisageables, et elles continueront à être explorées après la fin de ce stage. De plus, il serait possible d'essayer des modèles différents dans le cadre développé durant ce stage, à savoir à l'aide des fonctions génératrices et de la discrétisation aux distributions du nombre de potentiels d'action dans plusieurs fenêtres temporelles. Les premiers résultats obtenus ici montrent que cette nouvelle approche est potentiellement aussi puissante que celles déjà existantes.

Acknowledgements

I would like to express my gratitude to Kolia for his very substantial help and our fruitful discussions. I would also like to thank Michael Berry for allowing me to do my internship in his lab, and the Princeton University staff (especially Rebecca and Elena) for offering me a few months in a very nice place. Many thanks also to Romain Brette and Rava da Silveira for their guidance concerning the choice of my internship, and Alain Berthoz for his generous help. Finally, I am also grateful to Donna, Bill, my family and Claire.

Introduction

General introduction

The brain is an organ of bewildering complexity, perhaps the most complex in the universe. In order to understand its functioning, experimentations involving electrophysiological data gathering are essential. Recently, new experimental devices like multi-electrode arrays have allowed neuroscientists to record *simultaneous* activity of many neurons. In the past, our technology allowed us to record only the activity of single neurons. This technological advance offers us a new promising ability to understand how the functioning of the brain *emerges* from interactions between many neurons. Yet, a full comprehension of the functioning of the brain critically needs statistical tools and theoretical modeling in addition of data gathering.

The vertebrate retina is a part of the brain simple enough to be studied quantitatively, but complex enough to be interesting. Experimental and theoretical studies on the retina may allow us to understand processes occurring within the whole brain. Therefore, I've worked during this internship on probabilistic and statistical tools for understanding the complex statistical structure of the neural activity of the retina. They offer a new approach to the analysis of any electrophysiological data and a better comprehension of the neural code.

This thesis presents first the broad context of this internship : the early visual system, computational neuroscience and statistical analysis of electrophysiological data. Then, my work on the theoretical tools and on their application to retinal activity modeling are presented in the second part of this thesis.

Contributions

I've made the following contributions during my internship.

1. I've written a short mathematical introduction to the theory of point processes on the real line. It contains basic definitions and results needed for applications to spike train data analysis (see Appendix C). Almost all presented results are in classical reference textbooks [5, 6]. Those precise statements are required to define properly particular point process models (see Theorems 13 and 24).

2. I've written a *Matlab* implementation of the maximum pseudo-likelihood algorithm. It can be used for any point process model given in terms of a probability generating functional (see Section 6.3 for details).

3. This program relies on a simple algorithm for inverse-transforming probability generating functions using a *Fast Fourier Transform.* I proved the convergence of this algorithm (see Appendix B).

4. I've performed statistical tests to show that, according to our experimental data, ganglion cell spike trains are not Poisson.

5. I've defined with Kolia Sadeghi a point process model for retinal ganglion cell spike trains (see Section 7), called the PNMP model.

6. I've used the Matlab program to infer the PNMP model's parameters from the data, and to compare model predictions with experimental observations.

Overview of this thesis

This thesis is organized as follows.

- **Scientific context**. This part presents the general scientific context this internship is in line with. It is intended to those who have no knowledge in biology or in computational neuroscience. Reading this whole part is not required to understand the mathematical aspects of this internship. However, it can be useful to refer to it in order to understand the biological motivations and implications of this work. Besides, previous work about the statistical study of the electrical activity of the retina, which my internship forms the continuation, are presented.

  ◦ *The Berry Lab* in Section 1. It introduces the research project and the experimental protocol in the Berry Lab, in which I did my internship.

  ◦ *The visual system* in Section 2. It gives the bases needed to understand the biological object under study : the retina and the visual system.

  ◦ *Computational neuroscience* in Section 3. It is an introduction to computational neuroscience, which the general research project of the Berry Lab is a subdomain. While the goal of computational neuroscience is to understand the algorithmic processes occurring in the brain, the goal of the Berry Lab is to focus on computations in the retina (which is part of the brain).

  ◦ *Statistical exploration of spike train data* in Section 4. It presents some theoretical aspects of the exploration of electrophysiological data, as well as previous work on statistical tools and properties of retinal ganglion cell spike trains.

- **Point process models of spike trains**. This part details my actual work on probabilistic and statistical tools for spike train data analysis, and their use in the research of a satisfying point process model of retinal ganglion cell spike trains.

  ◦ *Theory of point processes* in Section 5. This section presents the bases of point processes theory, and some tools to define and manipulate particular models.

  ◦ *Modeling spike count statistics of groups of cells* in Section 6. A statistical tool to fit models on data is presented in this section. It is called the *maximum pseudo-likelihood algorithm.* It lies in a discrete framework, contrary to probabilistic tools lying in a continuous framework.

  ◦ *A path toward an efficient model for spike trains* in Section 7. In this section, we use the probabilistic and statistical tools presented before to design a particular model for ganglion cell spike trains. The path we took to find this model is detailed in this section.

    - *Results of the PNMP model* in Section 8. Some results of this model are presented in this section.

- **Mathematical appendices** contain precise mathematical definitions, theorems and proofs that are used throughout this report.

    - Appendix A contains the definition of probability generating functions and marginals.
    - Appendix B contains the proof of an algorithm for inverse-transforming probability generating functions, used in the statistical tool presented in Section 6.
    - Appendix C is a self-contained mathematical introduction to the theory of point processes on the real line. It contains rigorous definitions and theorems used during this internship.

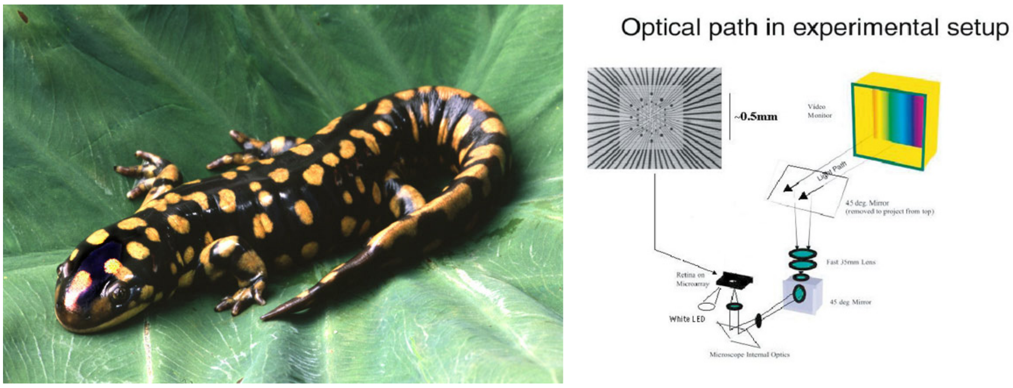
Optical path in experimental setup
~0.5mm
Video Monitor
Light Path
45 deg. Mirror (removed to project from top)
Fast 35mm Lens
Retina on Microarray
45 deg Mirror
White LED
Microscope Internal Optics

Figure 1: Left : a Tiger Salamander. Right : experimental setup in the Berry Lab.

Scientific context

1 The Berry Lab

1.1 Presentation

General presentation of the lab

The *Berry Lab* is located in the Department of Molecular Biology, in Princeton University. It is headed by Michael Berry, who is a physicist with an interest in computational neuroscience. Researchers in the lab are mostly physicists or biologists. There is also one mathematician (PhD candidate), Kolia Sadeghi, who has a background in computer science and applied mathematics. I've been working in collaboration with him during my internship.

Research project

The *vertebrate retina* is a thin sheet of neural tissue in the eye, realizing the first step of the vision process. It converts the visual information (the *stimulus*) into electrical signals (the *response*) called *spike trains*, which are transmitted to the brain. The research project of the lab is to understand the *computations* performed in the retina, that is, the algorithmic processes converting images into spike trains, and the nature of the output signals of the retina.

Experimental protocol

In order to study the functioning of the retina, experiments are done in the lab. They consist in recording the electrical activity of *ganglion cells* in the retina (the output, generating electrical signals for the brain) while projecting a visual flow to the retina. Retinas used for experiments are extracted from *Tiger Salamanders* (a particular species of amphibians, see Figure 1, left). An optical device projects a stimulus chosen by the experimentalist onto the retina. A multi-electrode array set on the retina is then able to record simultaneously the activity of hundreds of ganglion cells (see Figure 1, right).

1.2 Research directions

Spike sorting

Several specific issues interest the researchers in the lab. Before all, the multi-electrode array technique requires a post-processing algorithm, called *spike sorting*, to extract spikes from raw data. It is briefly mentioned in Section 4.2. It involves algorithmic and statistical tools. Several spike sorting algorithms already exist, but they are not well-adapted to the experimental protocol in the lab. Researchers in the lab try to improve these algorithms for the specific case of the retina.

Encoding

The *encoding* issue consists in developing mathematical models for the transformation process performed in the retina, which maps the visual information into a set of spike trains. These models often involve several layers, each one modeling a particular neural layer in the retina. Linear and non-linear filters are then combined to model the whole process.

Decoding

The *decoding* issue is the inverse of encoding. It consists in developing algorithms to recover, from the examination of spike trains, the stimulus which generated them. An efficient decoding algorithm might help us to understand the encoding process in the retina, and the way higher areas in the visual cortex deal with spike trains coming from the retina to perform cognitive visual tasks.

Motion prediction

A biological issue of interest is the reaction of the retina to a moving visual stimulus. Indeed, it has been shown that the retina is able to detect regularities in the motion, and predict future motion of an uniform motion, for example. It allows an efficient way for coding information. In the lab, simple mathematical models of the encoding process performed in the retina are designed to capture that predictive feature.

Reliability, redundancy and compression

An other topic of interest in the lab concerns reliability, redundancy and compression. The visual stimulus carries a huge amount of information, which must be transmitted and processed very rapidly by the brain. An efficient *compression* task is performed by the retina, taking into account any spatio-temporal statistical regularity in the natural world. On the other hand, the neurons being very noisy by nature, a certain amount of *redundancy* during the coding process is needed in order to yield a *reliable* signal. How this trade-off between compression and redundancy is performed in the retina is a very interesting theoretical question about the retina and the nervous system in general. Exploring these questions requires a certain amount of theoretical tools, especially from information theory.

Role of correlations

Researchers also study the statistical structure of the output signals of the retina (spike trains emitted by a large number of ganglion cells). The complex statistical structure observed in this set of spike trains is believed to contain important information about the stimulus. It is the concept of *population coding.* In particular, it was recently shown that correlations between cells are responsible for a very important part of this structure. Quantifying the correlations between spike trains of different cells and understanding their role in the neural code are fundamental issues in computational neuroscience.

Point processes approach to spike trains statistical analysis

Finally, a last research direction in the lab consists in using the mathematical theory of *point processes* to study multiple ganglion cell spike trains. The goal is to find specific point process models describing the firing statistics of ganglion cells, and to give a framework for investigating the role of correlations between different cells. That is the specific issue I've been working on during my internship, in collaboration with Kolia Sadeghi.

2 The visual system

In this section, we give an overview of the biological object of study in the Berry Lab : the visual system, and especially the retina. The main reference is the PhD thesis of Adrien Wohrer [16].

2.1 Overview of the visual system

Introduction

Visual perception is the ability to interpret information from visible light reaching the eyes. This task is extremely complex, and is entirely performed in the cortex. Before the information reaches the cortex, it is processed through several layers which are part of the *early visual system.* The latter comprises the *eye* (including the *retina*), the *lateral geniculate nucleus* (LGN), and the *primary visual cortex.*

The eye

The eye (see Figure 2, left) contains two successive lenses, the *cornea* and the *crystalline lens*, which project incoming light on the retina, at the back of the eye (see Figure 2, left). The *iris* (responsible for the color of the eyes) is a circular membrane at the front of the eye and plays the role of an optic diaphragm. It surrounds a variable aperture : the *pupil*, which provides a first adaptation to the incoming levels of light in the eye.

The *retina* is a thin sheet of neural tissue at the back of the eye. It converts light into electrical activity. Its anatomy and physiology are detailed in the next subsection. Retinal output axons are bundled in the *optic nerve*, which links the retina to the thalamus and the cortex.

Beyond the eye : the LGN and the cortex

More specifically, the optic nerve projects onto the *lateral geniculate nucleus* (LGN), which is part of the thalamus, and plays the role of a relay station between the retina and the cortex. From the LGN, the visual information is then transmitted to the *primary visual cortex* (V1) which is a two-dimensional, layered sheet of neurons. It's located in the *occipital lobe*, at

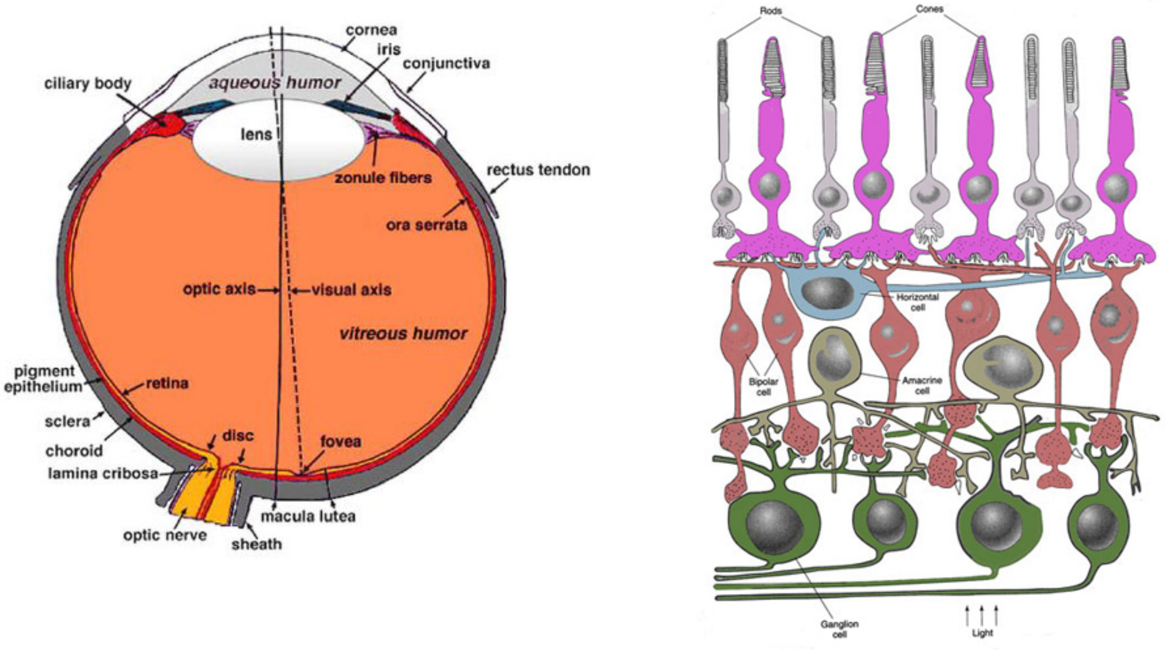

Figure 2: Left : anatomy of the eye (from [16]). Right : different types of cells in the retina (from [9]).

the rear of the brain, and it is the first cortical processing step during visual perception. It performs, among others, edge detections and segmentation. Finally, higher-level areas in the brain (V2, V3, ..., V8) are responsible for advanced processing of the visual information.

2.2 The retina

An organization in layers

There are five different types of cells in the retina, which are organized along five different layers (see Figure 2, right). Each type of cell has a specific function in the general process of transforming light into electrical activity suitable for the brain.

Photoreceptive cells

Incoming light goes through the whole thickness of the retina and reaches *photoreceptive cells* at the back (on the top of the figure). They convert light into electrical activity during the *phototransduction process*. There are two types of photoreceptive cells : *cones* and *rods*. Cones are basically activated during daytime, whereas rods are useful especially during the night. The density of photoreceptive cells is much higher in the *fovea* (in a way, the center of the field of view) than in the remaining of the retina. It implies that vision is precise only on the very center of the field of view.

Horizontal, bipolar, amacrine and ganglion cells

Photoreceptive cells transmit signals via synaptic connections to *bipolar cells*. *Horizontal cells* are involved in inhibitory modulation of those transmissions. Then, bipolar cells transmit signals to *ganglion cells*. *Amacrine cells* are involved in inhibitory modulation of those transmissions. Ganglion cells perform the last processing step in the retina. Their axons are bundled in the optic nerve ; they emit action potentials (*spikes*) which propagate to the LGN and the cortex.

Experimental advantages of ganglion cells

In the lab experiments, multi-electrode arrays record extracellular currents of ganglion cells. There are mainly three reasons to record the activity of ganglion cells instead of other types of cells in the retina [16]. First, the functional importance of the retina lies in its output : spike trains generated by ganglion cells. They are the only signals the cortex receives about the visual information. Second, ganglion cells are much larger than other retinal cells, which small size make experimental measures difficult. It implies that recording electrical activity is easier for ganglion cells. Finally, ganglion cells fire spikes, which can be recorded by extracellular recordings. Other retinal cells, on the contrary, can only be studied by intracellular techniques, since they don't fire any kind of spikes.

3 Introduction to computational neuroscience

3.1 The lab project : a subdomain of computational neuroscience

Computations in the retina and in the brain

The research project in the Berry Lab is to understand the theoretical functioning of the retina, i.e. discover the algorithmic processes occurring in the retina. Computations occur not only in the retina, but also within the whole brain. Understanding them is the goal of *computational neuroscience*. An analogy with computers might help to understand the nature and the role of those computations.

In a computer, information is encoded as series of *binary digits*, and transmitted by *transistors*. In the brain, information about the external world is encoded as a temporal series of pulse-like stereotyped electrical signals called *spikes*. The latter are emitted and transmitted by *neurons*. The conversion between analogue signals from the external world (light, sound...) and spike trains is performed by particular neurons called *receptive cells* (like in the retina). Finding the rules behind the theoretical mapping between stimuli and spike trains is the main goal of computational neuroscience.

Most of the work in the lab concern the particular case of the retina, but some of them are general enough to be applied to the whole brain. For example, theoretical tools for analyzing spike train data are quite general. The computational study of the retina should thus be considered as a subdomain of computational neuroscience, since a better comprehension of computations in the retina can lead to a better comprehension of the computational functioning of the brain.

Advantages of studying the retina in computational neuroscience

Why choosing the retina in computational neuroscience, instead of another part of the brain ? First, the retina is an important part of the brain since it is the entry point in the vision process, and vision is very important among primates and humans. Moreover, the retina is a relatively simple part of the brain, but its functioning is nonetheless quite interesting and non-trivial [11]. A more pragmatic reason is that it can be removed from the eye without damage to internal connections, and continues to function in vitro for several hours. Also, the general function of the retina is well known. It converts the visual information into neural signals suitable for the brain, and transmits the result through the optic nerve to the brain.

Additionally, inputs and outputs are well identified, and can be respectively controlled and recorded. The input is the visual flow which hits photoreceptive cells, and can be easily controlled during experiments. The output is a pattern of neural activity generated by the ganglion cells, and can be recorded with a multi-electrode array. Very few other parts of the brain have these advantages of admitting well-defined inputs and outputs. Furthermore, unlike most areas of the brain, there are no backward projections from higher parts in the brain onto the retina. It means that the only input in the retina is the visual flow. It is a very rare property in the brain, since, particularly in the cortex, there is a large amount of recurrent connections between different areas. It implies that retina processing can be studied as a closed problem involving solely the retina itself [16].

3.2 Biological foundations

In the following paragraphs, we present the biological and theoretical foundations of computational neuroscience.

Neurons

Neurons are electrically excitable cells which form the basic components of information processing in the central nervous system. They are essentially made up of three different parts (see Figure 3, left) : the dendrites, the soma, and the axon. Retinal ganglion cells are particular types of neurons.

Input electrical signals (waves of voltage between the inside and the outside of the membrane cells) come from hundreds of neural cells connected to the neuron's *dendrites.* The junctions between neurons are called *synapses* : they relay the information via the release of *neurotransmitters* during the *synaptic transmission* process. Dendrites conduct the electrical stimulation to the cell body, or *soma.* The incoming electrical signals are then added up, and if the resulting voltage exceeds a threshold, a new signal is transmitted through the *axon* cable to an output neuron's dendrite.

Spikes

Input electrical signals coming from the synapses are added during the *integration process.* If the resulting voltage of the cell's interior relative to the cell's exterior is raised above some threshold, an *action potential*, or *spike*, is generated at the beginning of the axon. It is a very brief pulse-like stereotyped wave of voltage of high intensity (see Figure 3, right). It travels along the axon until its end, at a synapse with another neuron, where it usually causes a new synaptic transmission.

Just after a spike has been emitted, the cell can't emit any other spike during an interval of a few milliseconds, called the *refractory period.* The very brief duration of spikes (a few milliseconds) allows us to consider them as instantaneous events. Therefore, it is now assumed that neurons communicate with a *discrete* code, made of a temporal series of spikes called *spike trains.*

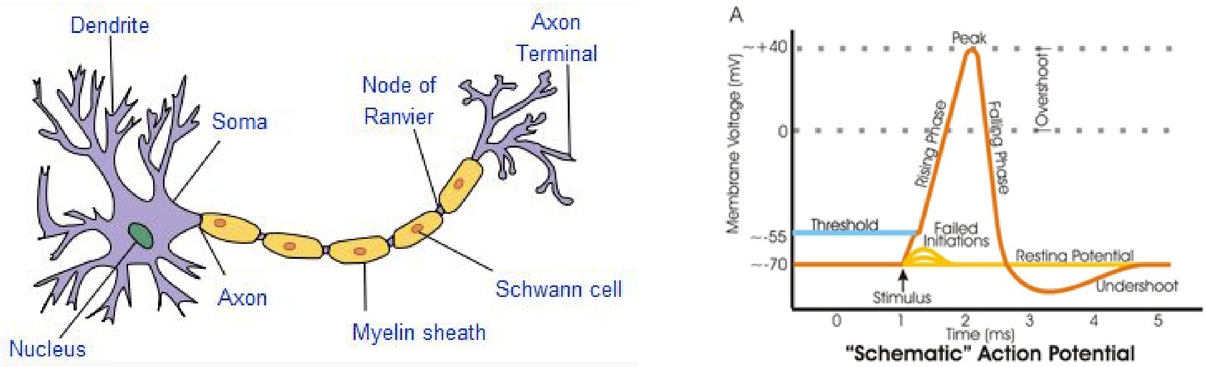

Figure 3: Left : anatomy of a neuron (from [15]). Right : an action potential, or spike (from [14]).

3.3 Some theoretical issues

The stochastic nature of neuronal activity

When a stimulus is presented several successive times to a sensory neuron, the resulting successive spike trains are similar, but not identical : one can observe significant trial-to-trial variability. It shows the stochastic nature of neuronal activity. Despite that noise, information encoded by receptive cells is often very reliable.

That very important observation implies that theoretical models for studying the neural code should be developed in a probabilistic context. For example, the mathematical process of encoding, transforming a physical signal to a pattern response of neural activity, should not be a classical function, but rather a conditional *probability distribution* of observing a response given the stimulus. Regarding data exploration tools, such noise requires the use of a statistical framework.

The neural code

Neurons in the brain communicate using spike trains. For example, *sensory receptors* are neurons which convert physical stimuli to spike trains, allowing the brain to process information about them. Cells in the retina, olfactory neurons or taste neurons are some examples of sensory receptors. Each different stimulus which is accessible to our sensory system is represented in the brain as a pattern of spikes trains : that is the notion of *neural code*.

There are basically two different ways of considering neural coding from a theoretical point of view. The first is linked to the concept of *firing rate* : it is the *average* local density of spikes as a function of time. *Rate coding* assumes that only the firing rate is relevant as regards the neural code, spikes being thought as independent realizations of Poisson (random) processes. *Spike coding*, on the other hand, assumes that precise spike timing is relevant, and that spike trains are not totally independent. Rate coding models are theoretically simpler since they involves continuous functions. Spike coding models involve discrete stochastic processes (emissions of spikes), and classical mathematical tools are not available anymore.

The specific question whether rate coding or spike coding is actually used throughout the brain was highly controversial during the past decades. It is nevertheless globally assumed nowadays that only spike coding can account for every complex feature experimentally observed in the brain (such as correlations between cells or synchronization).

Modeling neuron activity

Theoretical studies in computational neuroscience involve design of neuron models. There are basically two sorts of models : *biological models* and *computational models* [2]. Biological models aim at describing biological, physical or chemical phenomena in action during neuronal activity. The famous Hogdkin-Huxley model [7] is an example of biological model : it describes the emission of an action potential with four non-linear differential equations modeling the dynamics of ionic channels in the cell membrane. There are a lot of other biological models for both subthreshold dynamics and spike emission (the integrate-and-fire model [1] for example).

Computational models describe networks of neurons, in order to understand how neurons interact to perform computations all together. Cellular mechanisms involved in neural activity are not addressed in these models : each neuron is considered as an idealized mathematical object, performing a simple computation on its inputs, and giving the result in its output. Understanding the emergence of complex behaviors in a network of simple components (neurons) is the fundamental issue in these models. McCulloch and Pitts neurons [10] or Hopfield networks [8] are some examples of computational models.

4 Statistical exploration of spike train data

4.1 Overview

In order to understand the neural code, it is essential to perform experiments and collect spike train data. Then, statistical analysis of the data is an essential complement to data gathering. It is closely linked to modeling neuron activity. Statistical studies of the data allow to discover properties of the biological objects. These properties are then reflected in the models. Finally, statistics allow to compare models with the data.

In this section, we give a few different aspects of spike train data analysis. First, we present the very first step of spike train data analysis, *spike sorting*, which consists in extracting discrete emissions of spikes from continuous extracellular recordings. Then we describe two previous work which this internship is based on. The first concerns a particular set of theoretical tools used in spike train data analysis. The second is a deep result about the statistical structure of the retinal code.

4.2 Spike sorting

For a few decades now, we have been able to record the electrical activity of a large number of neurons simultaneously thanks to multi-electrode arrays. For the particular case of the retina, ganglion cells' extracellular activity is recorded while the visual stimulus is experimentally controlled [11]. Discrete spike emissions can be recovered from continuous extracellular recordings thanks to *spike-sorting* algorithms. They are basically based on statistical learning algorithms. That is the first step in any spike train data analysis.

Intracellular recordings

In order to record the electrical activity of neurons, two methods can be used. Intracellular recordings require the insertion of a sharp micro-electrode inside the cell which records the voltage between the inside and the outside of the membrane cell. The intracellular recording

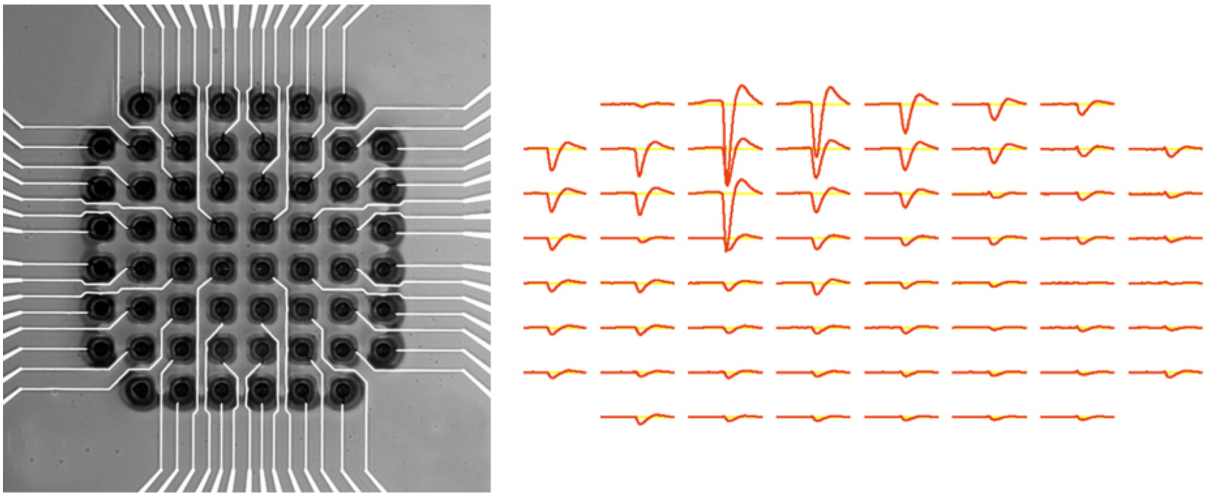

Figure 4: Left : a multi-electrode array. Right : a multi-waveform.

technique of the *voltage clamp*, invented in the 1940's, allowed Hodgkin and Huxley (Nobel Price in 1963) to design a mathematical model of the generation of an action potential [7]. Those techniques are difficult to perform in-vivo.

Extracellular recordings

Extracellular recordings involve the introduction of an electrode into the neural tissue, or the positioning of a multi-electrode array (see Figure 4, left) on the neural tissue (like the retina). In the latter case, electrodes record voltages generated in the extracellular matrix by the current fields outside the cells in the local region when they generate action potentials. Thus, each electrode records the activity of several cells. The intensity of the signal decreases when the distance to the electrode increases. Furthermore, signals recorded during emissions of action potentials are smaller than with intracellular recordings. Figure 4 (right) shows signals recorded by each electrode during a small interval of time. One can observe a spike generated by a cell close to an electrode in the top-left of the array. Signals recorded by farther electrodes are smaller with distance.

Goals of ***spike sorting***

Signals recorded by extracellular electrodes are continuous, whereas spike emissions are discrete. Therefore, experiments involving recordings by a multi-electrode array need a post-processing algorithm for extracting occurrences of spikes from electrode signals. It is a sort of shape recognition problem : the particular form of spikes helps to detect them in the signals. That is the *spike-sorting* problem ; it is difficult in general for several reasons. First, spikes generated by different cells can overlap in time. Second, there is a certain amount of noise in the recordings. Finally, the algorithm should be able to identify which cell emitted each spike.

The step of spike sorting is very important in spike train analysis. Indeed, statistical exploration of spike trains deals exclusively with outputs of spike sorting algorithms. Therefore, it is critical to have an efficient spike sorting method that doesn't introduce any bias.

4.3 Spike train data analysis with point processes

Here we mention a specific line of work applying the theory of point processes to spike train data analysis. The second part of this thesis describes my work on an other method of application.

Point processes

Point processes are the mathematical formalization of spike trains. Multidimensional point process models can account for experimentally-observed correlations between spike trains. Such models, if shown to be accurate, can help us to explore the role of correlations in the retinal code.

The more basic point process model for spike trains consists of independent Poisson processes. It is a simple but inefficient model in several cases, and in particular for the retina. Indeed, as mentioned later in this section, it can't capture experimentally-observed ISI distributions neither correlations between cells. Therefore, a few work, described below, tried to design more accurate models.

The conditional intensity function

Previous work using the theory of point processes to study spike trains use the *conditional intensity function* (CIF). The latter can be described as a "conditional firing rate" of a point process : it is the probability that a spike is emitted between t and $t + \mathrm{d}t$ given the past of the process until time t. An analytical expression for the likelihood function can be usefully obtained from the CIF, making statistical inference easy.

Goodness-of-fit tests

Goodness-of-fit tests can be performed using the time-rescaling theorem and a Kolomogorov-Smirnov plot [4]. A suitable time reparametrization transforms any point process model given in terms of its CIF into an homogeneous Poisson process. Then, the Kolmogorov-Smirnov plot displays the rescaled empirical versus Poisson cumulative distribution functions. The closer the plot is from a $y = x$ line, the better is the model. Also, the *Aikake Information Criterion* (AIC) can be used. It is a score giving the overall goodness-of-fit of a model with respect to the data. It measures the trade-off between how well the model fits the data and the number of model parameters needed to achieve this fit.

Results

Those tools were applied to several data sets [3], and in particular to retinal ganglion cell spike trains. Several models were tested : the Poisson process, and point processes with gamma and inverse Gaussian ISI distributions. Goodness-of-fit tests described above showed that the inverse Gaussian distribution is better than the two other models.

A new approach

The limitation with the previous approach is that models defined with their ISI distributions are renewal processes. Non-renewal processes may be interesting in order to capture time correlations. Also, defining a model with its CIF is difficult except for a linear autoregressive model. An approach enabling to design a greater variety of models was investigated during this internship. It makes use of the *probability generating functional*, a simple but powerful tool. To our knowledge, this tool hasn't been used yet in spike train data analysis. It is the main topic of the second part of this thesis.

4.4 Role of correlations

Here we give important results obtained recently about the statistical structure of retinal ganglion cell spike trains. They emphasize the role of correlations between cells in the neural code of the retina, and legitimize the approach used in this internship.

Modeling correlations

It was recently shown that correlations between cells capture a very large part of the whole statistical structure of the retina output [12]. More specifically, let $\sigma_i = \pm 1$ be the activity of cell i : +1 if it spikes, -1 otherwise. At any time, the state of the system (comprising N cells) can be described by a N-long binary word, where each digit is the activity of each cell. The object under study is the probability distribution over the set of all N-long binary words.

The Ising model

A model for that probability distribution, assuming observed pairwise correlations but no higher-order interaction, was used. The more natural model satisfying those conditions is a maximum-entropy model. It is called the *Ising model* and it comes from statistical physics. Its expression is :

$$P(\sigma_1, \ldots, \sigma_N) = \frac{1}{Z} \exp\left(\sum_i h_i \sigma_i + \sum_{i \neq j} J_{ij} \sigma_i \sigma_j \right),$$

where Z is the partition function (normalization factor). The h_i and J_{ij} coefficients must be chosen in order to fit *exactly* the observed firing rates and correlations (in other words, so that order 1 and 2 moments fit the data). Computing these parameters is a particular case of a *Boltzmann machine learning* problem [13]. One method for this computationally hard problem is Monte Carlo simulation.

Results emphasize the importance of correlations

Results show that the Ising model is much better than an independent model which assumes no interaction between cells. Moreover, it was shown with tools from information theory that this model captures 90% of the statistical structure of the output. It suggests that order-2 interactions are sufficient to explain the statistical structure of the retinal neural code. It offers a very interesting simplification since higher-order interactions imply a big complexity cost in any model. As a consequence, a model successfully capturing correlations is probably also able to capture the whole statistical structure of the neural code. This result legitimizes the fact that in the second part of this thesis, we try to design a model for spike trains that captures well correlations between cells.

Point process models of spike trains

5 Theory of point processes

Point processes are the mathematical formalization of single or multiple spike trains. The theory is relatively recent, since it has been extensively studied especially in the second half of the twentieth century. There are applications in a lot of very different scientific domains that involve random instantaneous time events (forestry, epidemiology, zoology, geography, seismology, astronomy...).

Applications of the theory of point processes to spike train data analysis consist in defining point process models for spike trains, assess their validity, and possibly infer their parameters to fit the data. One would like such models to capture observed statistical properties of spike trains. Those models would then give a framework for investigating the role of correlations between cells in the neural code.

During my internship, I've studied a way for designing point process models for multiple spike trains, and a statistical tool to infer their parameters. Then, I've applied those tools to find an efficient model with respect to experimental data obtained in the lab. The results of these work are presented in the remaining of this thesis.

5.1 Definitions of point processes

There are several ways of defining a point process. Here, we give first an intuitive definition, and then we give the formal definition of a point process, following the reference textbooks [5, 6]. More precise mathematical statements about the foundations of point processes can be found in Appendix C. It contains every mathematical definition and theorem required in order to design particular models of point processes.

An intuitive definition

A *point process* on the real line is a random process of discrete events $\ldots < t_{-1} < t_0 < t_1 < \ldots \subset \mathbb{R}$. In neuroscience applications, each t_i represents the time of a spike emission. Actually, one could give a formal definition of a point process this way, but a very more general definition is available.

A measure-theory definition

A general point process can be defined over any well-behaved topological space. In this document, we are only interested in point processes on the real line, even if some generalizations are needed to consider, for example, multidimensional point processes. Such details are treated only in the appendix.

Recall that a simple counting measure over $\mathbb{R}$ is a measure μ such that $\mu(B) \in \mathbb{N}$ for any bounded Borel set B in $\mathbb{R}$, and, $\forall x \in \mathbb{R}$, $\mu(\{x\}) \in \{0, 1\}$. Such a measure assigns the (finite) number of events in any bounded subset $B \subset \mathbb{R}$. The points $x \in \mathbb{R}$ such that $\mu(\{x\}) = 1$ are the *atoms* of the measure. The space $\mathcal{M}(\mathbb{R})$ is the set of all simple counting measures.

A *point process* is a *random* simple counting measure N, that is, a measurable map $N : (\Omega, \mathcal{A}, P) \to \mathcal{M}(\mathbb{R})$, where $(\Omega, \mathcal{A}, P)$ is a probability space. Each realization of the point process is a simple counting measure. Its atoms are the time events (spikes) of the realization.

The *counting process* associated to the point process, also noted $N(t)$, is defined as the number of spikes between 0 and t (in the case where the support of the point process is in $\mathbb{R}_+$). It is an increasing, right-continuous, integer-valued step stochastic process, and contains all the information about the point process.

5.2 Common objects associated to point processes

In this section, we give the definitions of some objects commonly associated to point processes, which are useful in order to define and manipulate particular point process models.

Interarrival times

Instead of considering the arrival times t_i, one can consider the *interarrival times* $u_i = t_{i+1} - t_i$. They are positive random variables. Joint probability distributions of subsets of interarrival times can be used to define particular classes of point processes. For example, assuming that the interarrival times are independent and identically distributed leads to the notion of *renewal processes*. Such processes have the property that the occurrence of an event at time t depends only on the time elapsed since the last event. To define a particular renewal process, it is sufficient to give the probability distribution of the interarrival distribution.

Event counts

One can also consider the number of events in a fixed time window, or in a set of time windows, yielding a joint discrete probability distribution. Indeed, the number of events in a window is a random variable over $\mathbb{N}$. Specifying such distribution for any set of time windows completely characterizes a point process, under certain compatibility conditions (see Appendix C for details).

The probability generating functional

The *probability generating functional* (pgfl) of a point process is the function $G(h)$, where h is a well-behaved function $\mathbb{R} \to [\,0, 1\,]$ (such that $h(t) = 1$ if $|t|$ is large enough), defined by :

$$G(h) = \mathbb{E}\left(\exp\left(\int_{\mathbb{R}} \log h(t) N(\,\mathrm{d}t)\right)\right) = \mathbb{E}\left(\prod_i h(t_i)\right),$$

where the t_i's are the random time events of the point process. It completely characterizes a point process. From the pgfl, it is easy to obtain the *probability generating function* (pgf) of the joint probability distribution of spike counts in any set of disjoint time windows. Actually, the pgfl is simply the generalization of the pgf for point processes. That property will be used for statistical inference of point processes in Section 6.

The pgfl allows to define a particular point process model : one first expresses a formula for the pgfl, and checks a few natural conditions (see Theorem 24). This theorem then claims that a point process with that pgfl actually exists. This tool is powerful since it allows to express a large variety of different models.

The conditional intensity

The *conditional intensity* is the stochastic process $\lambda(t)$ defined (informally) by :

$$\lambda(t) \simeq \frac{1}{dt} P(N(t+dt) - N(t) = 1 \mid \mathcal{H}_t),$$

where $\mathcal{H}_t$ is the past of the process up to time t. Therefore, $\lambda(t)$ is the conditional expectation that a spike occurs between t and $t+$ dt, given the past up to time t. It is a bit difficult to define the conditional intensity in formal terms. One possibility is to use the Doob-Meyer decomposition of a submartingale.

One can define a point process by giving an expression for the conditional intensity. An important example is the class of self-exciting processes (or Hawkes processes), where the conditional intensity is basically a linear regression of the past.

5.3 Generalizations, operations and properties

Multidimensional point processes

A *multidimensional point process* is a collection of a finite number of point processes on $\mathbb{R}$. One can also see it as a point process over $\mathbb{R} \times \{1, \dots, K\}$ (it is a particular case of a *marked point process*). Each event occurrence is of the form (t, k), where t is the time of the event, and k the index of dimension. For example, multiple spike trains are represented by multidimensional point processes. The index is the cell which emitted the spike.

Independent superposition

Let N_1 and N_2 be two point processes. One can define the *independent sum* $N = N_1 + N_2$: it is the independent superposition of the point processes. The event count in a time window $B \subset \mathbb{R}$ is the independent sum of the event counts of N_1 and N_2 : $N(B) = N_1(B) + N_2(B)$, where the two terms in the sum are independent discrete random variables. Also, the pgfl of N is the product of the pgfls of N_1 and N_2 : $G(h) = G_1(h)G_2(h)$. The independent superposition of point processes is the simplest operation, but it is not very useful when defining a new point process. It can't capture any correlation across time or across dimensions (for multidimensional point processes).

Cluster processes

A *cluster process* is a more complex operation on point processes. One considers a first point process, called the *center process*, which events represent centers of clusters. A second point process, called the *component process*, defines the behavior of the events inside each cluster. The two point processes are assumed independent. A few assumptions on the center and component processes are needed so that a cluster process doesn't explode, i.e. the events are almost surely finite in any finite Borel set in $\mathbb{R}$.

If the pgfls of the center and component processes are respectively $G_c(h)$ and $G_m(h)$, then the pgfl $G(h)$ of the cluster process is the composition of the pgfls : $G(h) = G_c(G_m(h(\cdot - t)))$. This notation should be explained : the functional G_c takes the function $t \mapsto G_m(h(\cdot - t))$ in input, where $h(\cdot - t)$ is a translated version of h.

This operation is useful because it might be relevant for spike trains models. Clusters could represent firing events, and spikes inside each cluster may have different statistical properties. That is the main operation used in Section 7, where we try to find an efficient model for retinal ganglion cell spike trains.

Stationary processes

In this thesis, we are interested in finding an efficient point process model for stationary spike trains. It means that spike trains are considered as if they were defined on the whole real line, and were translation-invariant. The condition for a point process to be stationary can be easily expressed with its pgfl : $G(h) = G(h(\cdot - t))$, for all $t \in \mathbb{R}$, where $h(\cdot - t)$ is a translated version of h.

5.4 The Poisson process

The Poisson process is a simple yet very important point process. That's why it is presented in this section, whereas more specific models for spike trains are presented in Section 7.

The homogeneous Poisson process

An (homogeneous) *Poisson process* with rate $\lambda > 0$ can be seen as a totally random point process. It satisfies two important properties :

1. *The interarrival times are independent and identically distributed* : they follow an exponential distribution with parameter λ. The probability of an event occurrence at time t depends only on the elapsed time since the last event.

2. The *independent increments* property : events occurring in a time window are totally independent of ones occurring in a disjoint time window.

The spike count distribution of an homogeneous Poisson process in an interval $[\,a, b\,]$ is a Poisson distribution of parameter $\lambda(b - a)$. Its pgfl is :

$$G(h) = \exp\left(\lambda \int_{\mathbb{R}} (h(t) - 1)\, \mathrm{d}t\right)$$

The inhomogeneous Poisson process

The Poisson process can be generalized first by removing the first property, leading to the definition of an *inhomogeneous Poisson process.* Its rate is a continuous positive function $\lambda(t)$. The interarrival times are not independent anymore, but the independent increments property still holds. An homogeneous Poisson process with parameter λ is an inhomogeneous Poisson process with constant rate $\lambda(t) = \lambda$.

The spike count distribution is a Poisson distribution of parameter $\int_a^b \lambda$. Its pgfl is :

$$G(h) = \exp\left(\int_{\mathbb{R}} \lambda(t)\, (h(t) - 1)\; \mathrm{d}t\right)$$

Renewal processes

Also, a point process which satisfies only the first property is a *renewal process* : the interarrival times are independent and identically distributed, but can follow any distribution over $\mathbb{R}_+$.

The special feature of the Poisson process among renewal processes lies in the *memorylessness property* of the exponential distribution of the interarrival times. This property states that $P(T > t + s \mid T > t) = P(T > s)$, and characterizes the exponential distribution among continuous probability distributions over $\mathbb{R}_+$. In a sense, the Poisson process is thus the most random process among renewal processes.

Interest of the Poisson process in computational neuroscience

The inhomogeneous Poisson process has been widely used in computational neuroscience. First, it is the simplest point process, and it has been extensively studied. Also, it is the correct mathematical formalization of spike trains when one assumes that a *rate code* is used in the brain instead of a *spike code*. The relevant quantity is then the firing rate (the rate of the inhomogeneous Poisson process), and spikes are emitted randomly from this rate. According to this point of view, spikes trains carry information only with their rate, and single spikes are irrelevant.

That point of view is now discussed : both correlations across time and cells and synchronization were shown to have important computational roles in the brain. Poisson processes can't account for that. Scientists now try to investigate the implications of the non-Poisson nature of spikes trains in the neural code.

In this internship, it was first checked that Poisson processes are a very bad model for stationary retinal ganglion cell spike trains. The goal was then to find more complex point process models that fit better the data. A statistical tool for checking the validity of point process models with respect to the data is presented in the next section.

6 Modeling spike count statistics of groups of cells

In this section, we describe a particular statistical method for estimating parameters of a multidimensional point process model given in terms of a probability generating functional with respect to the data.

6.1 Description of the problem

Original problem statement

The original problem statement is the following. We have a theoretical model of a multidimensional point process depending on a parameter $\Theta \in \mathbb{R}^M$, and we have experimental data of N_c simultaneous spike trains (N_c is the number of recorded cells). Under the hypothesis that these experimental spike trains follow this stochastic model, the problem is to estimate Θ from the data. In general, this is a difficult problem.

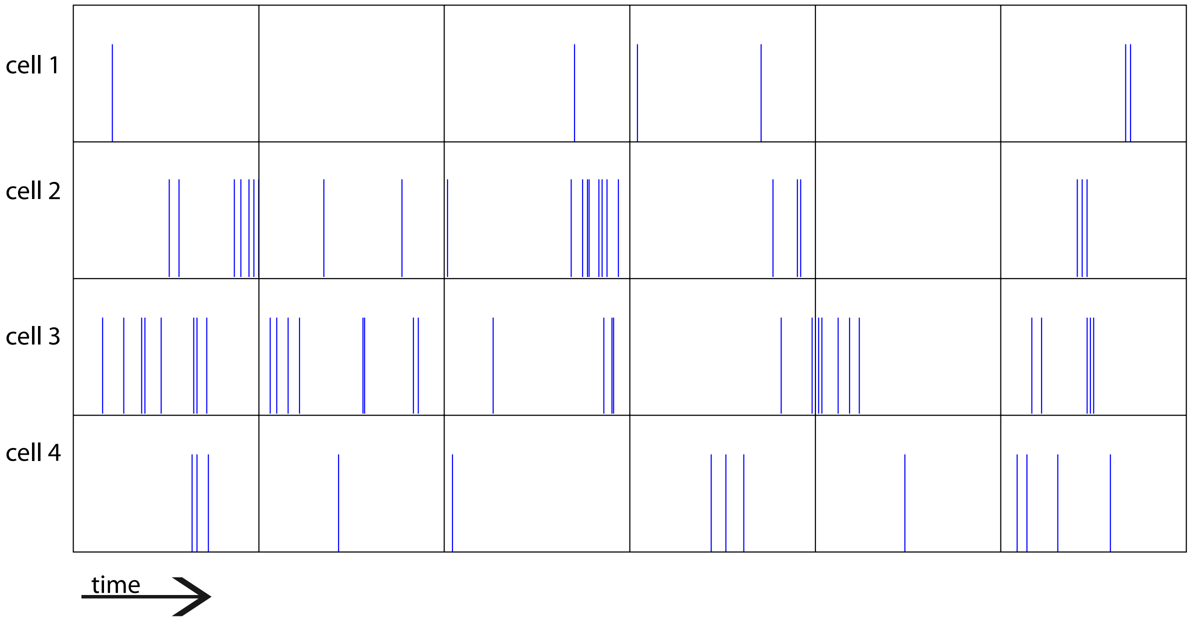

Figure 5: Spikes trains grid.

Binning process

The idea developed in this section is to transform this continuous estimation problem into a discrete one, via the *binning* operation. It means that we subdivide time in bins of equal length τ. In each bin, we consider the number of spikes.

We then restrict our attention to a grid consisting of N_b (Number of Bins) successive time bins and N_c cells. The total time course of this grid is $T = \tau N_b$. The object under study is then the $N_b N_c$-dimensional discrete spike count distribution in the grid.

As an example, Figure 5 shows experimental spike trains of four cells in a grid where $T = 3$ s. Here, the number of cells $N_c = 4$, the number of bins $N_b = 6$, the bin length $\tau = 500$ ms and the window length $T = 3$ s.

Discretization of the point process model

Recall that we have a multidimensional point process model for the spike trains, depending on a parameters vector Θ. This model is given in terms of a probability generating functional $G_\Theta(h_1, \ldots, h_{N_c})$. The goal of statistical estimation is, in this context, to find the Θ which fits best the data. As this problem can be difficult in general, we use the binning transformation to simplify the estimation process.

The pgf formulation allows to obtain an expression for the pgf of the joint probability distribution of spike counts in the grid from the pgfl of the point process :

$$\widetilde{G}_\Theta(z_{ij}) = G_\Theta\left(\left(\sum_{i=0}^{N_b-1} z_{ij} 1_{[\tau i, \tau(i+1)]}\right)_{j=1,\ldots,N_c}\right).$$

The N_c functions given as inputs for G_Θ are piecewise-constant functions. They are equal to 1 outside $[\,0, N_b\tau\,]$. Their values inside each bin are the parameters of the pgf of the multidimensional spike count distribution.

New problem statement

Finally, the new problem statement is the following. We observe successive independent [1] realizations of a $N_b N_c$-dimensional discrete probability distribution (the spike counts in each realization of a grid). The realizations are given in each time window $[\,Ti, T(i+1)\,]$. We have a model for this probability distribution in terms of a pgf depending on a multidimensional parameter Θ. The goal is to estimate Θ from the data.

The dimension of the probability distribution can be high, since the number of cells can be as large as the experimental conditions allow it (100+ cells). Moreover, we would like to have as much bins as possible in order to capture temporal correlations of spike trains.

In the remaining of this section, we begin by presenting common strategies for parametric estimations. Then, we study the particular statistical method of *maximum pseudo-likelihood estimation* to respond to this problem.

6.2 Parametric estimation of a probability distribution

Statistical estimation of a probability distribution

The *statistical estimation* problem is the following. We observe, from any type of experiment, a data set, consisting in a list of L multidimensional points $x_1, \dots, x_L \in \mathbb{R}^D$. In the following, we only consider the case where the data points always lie in $\mathbb{N}^D$. The starting hypothesis is that these data points are successive independent realizations of a probabilistic model. That is, we assume that there exists a probability distribution P over $\mathbb{N}^D$, such that $(x_1, \dots, x_L)$ is a realization of $(X_1, \dots, X_L)$ where the X_i's are independent identically distributed random variables over $\mathbb{N}^D$ with probability distribution P.

Maximum-likelihood estimation

In the *parametric* framework, one assumes that P lies in a given family of probability distributions $(P_\Theta)_\Theta$, where Θ is a multidimensional parameter in $\mathbb{R}^M$. The goal of the *parametric estimation* is then to find the Θ that fits best the data. The *maximum-likelihood* method consists in maximizing the *likelihood function* $L(\Theta) = P(x_1, \dots, x_L \mid \Theta)$ over the set of all possible Θ. The likelihood function is the probability of observing the data given the parameter Θ. The Θ which maximizes L is thus the parameter which explains best the data in this context.

Given that we made the hypothesis that the x_i are *independent*, the likelihood function takes the simpler form : $L(\Theta) = \prod_i P(x_i \mid \Theta)$. It is often convenient to take the logarithm of the likelihood :

$$\log L(\Theta) = \sum_{i=1}^{L} \log P(x_i \mid \Theta).$$

This function can also be expressed in terms of the *empirical probability distribution* $\hat{P}(k)$, which is the frequency of occurrences of $k \in \mathbb{N}^D$ in the data set :

[1] Actually, the independence condition is not necessarily verified. Temporal correlations can be observed over long periods (> 100 ms) in a single spike train. Therefore, a spike count realization in a time window is not necessarily independent from the realization in the following time window. However, if the window length T is high enough, one can assume that these realizations are independent.

$$\log L(\Theta) = \sum_{k \in \mathbb{N}^D} \hat{P}(k) \log P(k \mid \Theta).$$

Difficulty of computing the likelihood function

For practical applications, one should be able to compute the likelihood function, either analytically or numerically, in an efficient way. Indeed, its maximization can be performed by a numerical optimization algorithm, which requires an important number of function evaluations.

A first problem is that, when the dimension D of the probability distribution P is high, the analytic expression of P can be very complex, and its computation can be difficult. Moreover, the size of the data can be very important when D is high, and a large number of terms appear in the sum in the first expression of $\log L$. In the second expression, another problem is undersampling when D is large.

In order to compute the likelihood function when the probability distribution is expressed as a probability generating function (pgf) G, one first numerically computes the probability mass function (pmf) (the $P(k \mid \Theta)$ values) from G. The transformation from the pgf into the pmf is performed by an algorithm involving a D-dimensional *Fast Fourier Transform.* This transformation would occur several times at each step of the optimization procedure, for every evaluation of the likelihood function. When D is high, this method is totally intractable.

6.3 Maximum pseudo-likelihood estimation of a probability distribution given in terms of a probability generating function

The pseudo-likelihood function

Another possibility is to consider a variant of the likelihood function, called the *pseudo-likelihood function.* It consists in first considering a set of one-dimensional marginals. These marginals (see Section A.2) are sums over any subsets of the X_i's. In our case of interest, they consist in spike counts in any union of bins in the grid. They are unidimensional probability distributions.

Then, we introduce the *pseudo-likelihood* function :

$$\widetilde{L}(\Theta) = \sum_{j} \sum_{k=1}^{N} \hat{P}_j(k) \log P_j(k \mid \Theta),$$

where $\hat{P}_j(k)$ and $P_j(k \mid \Theta)$ are respectively the empirical and theoretical unidimensional marginals. The number N is such that $\hat{P}_j(k) = 0$ when $k > N$. It always exists since there is only a finite amount of data.

To compute the values of a theoretical marginal ($P_j(k \mid \Theta)$), one first needs to compute the unidimensional marginal's pgf from the multidimensional pgf of the model. As this marginal is a union of bins, its pgf can be computed immediately (see Proposition 2 in Appendix A). This particular property of the pgfs is one of the main advantages of defining models as pgfs. Then, the $P_j(k \mid \Theta)$ values can be computed from the marginal's pgf with a unidimensional FFT.

It is possible to use also marginals of several dimensions : joint distributions of unidimensional marginals. The contribution term in the likelihood expression is simply :

$$L_j(\Theta) = \sum_{k_1,\ldots,k_d=1}^{N} \hat{P}_j(k_1,\ldots,k_d) \log P_j(k_1,\ldots,k_d \mid \Theta).$$

Advantages and disadvantages of the algorithm

This procedure is more efficient than the full maximum-likelihood algorithm, but the result is theoretically less interesting, since the maximization function doesn't have a clear interpretation (especially because the marginals are not independent in general). Nevertheless, interesting results obtained with this algorithm may suggest that the model is interesting.

Implementation in Matlab

The maximum pseudo-likelihood algorithm was coded in Matlab. The main function, called *maxPL*, has three inputs : the model specifications (especially the probability generating function G), the list of marginals specifications (for each marginal, the set I with the notations of Section A.2), and the data (the list of the N first values of the probability mass functions of the empirical marginals). The output is the parameters vector maximizing the pseudo-likelihood function.

This function simply calls a Matlab optimization procedure (*fmincon* or *fminunc* for the constrained and unconstrained case, respectively) for the function *PLfun*. This function takes a parameter as input, and returns the pseudo-likelihood value and its gradient at the input parameter point. This computation involves a call to the function *laplace2proba*, which takes a probability generating function G as input, and returns the N first values of the corresponding probability mass function. It performs essentially a FFT of $(G(e^{ik\pi/N}))_k$.

7 A path toward an efficient model for spike trains

7.1 Introduction

Overview

In this section, we use the probabilistic and statistical tools presented in the previous sections to find an efficient stationary point process model of spike trains. The starting point is the Poisson process (Section 7.2) : several statistical tests show that it is a bad model. The investigation of experimentally-observed spike count distributions of multiple cells over different time bins lead us to introduce the *negative multinomial process* (Section 7.4). It is basically a generalization of the *negative multinomial distribution* for point processes. The existence of the negative multinomial (NM) process hasn't been found in the literature, but it was proved using classical point process theorems (see Theorem 26 in Appendix C). The model is shown to give better results than the Poisson process, but fails to capture the spike count distribution of single cells over large bins. Therefore, in order to improve the model, we use the *cluster process* technique to *compose*, in a precise mathematical sense, the Poisson and NM processes, leading to the PNMP model (Section 7.5).

Methodology

In order to find an efficient point process model for spike trains, we used the following methodology. First, we fitted different classical unidimensional discrete distributions to empirical spike count distributions of a single or several cells over different time bins. We used the method of moments or maximum-likelihood inference to fit those distributions to the data. Once a probability distribution was found, we tried to generalize the distribution to the point process case. That is, we tried to define a multidimensional point process model such that spike count distributions follow that distribution. We used probability generating functionals for this step (see Section 5). Then, we fitted the point process model to the data using the maximum pseudo-likelihood algorithm (described in Section 6). When the results were not accurate enough, we improved the model thanks to the cluster process technique.

To assess the validity of the models, we examined empirical and fitted unidimensional spike count distributions over several subsets of cells and time bins. We also compared empirical and theoretical spike count covariances of different cells.

Experimental data

We used two different data sets to fit our models. The *natural movie* data was obtained when a 26 seconds-long natural movie of bushes in a continuous loop (the total length experiment being one hour) was projected to the retina. The second data set was obtained with the *checker* experiment, where pixels appeared randomly white or black every few tens of milliseconds. All pixels acted independently across time and pixels. This stimulus was thus stationary.

7.2 The Poisson process

We investigate in this section whether spike trains from retinal ganglion cells can be well modeled as Poisson processes.

The Fano factor

For any given $T > 0$, the *Fano factor* $F(T)$ of a point process is defined as the quotient between the variance and the mean of the spike count distribution $N([\,a, a+T\,])$ in a time window of length T. If N is a Poisson process, the spike count distribution is a Poisson distribution, and $F(T) = 1$. We computed, from experimental data, an estimate of the Fano Factor for several time window lengths T. Indeed, when the stimulus is stationary, the Fano factor $F(T)$ can be estimated as the quotient of the variance and the mean of the experimental $N([\,nT, (n+1)T\,])$ values. In Figure 6, both experimental and Poisson Fano factors of a cell are shown.

Spike count distributions

In Figure 7, experimental and fitted Poisson spike count distributions are shown. The probability distribution is the spike count distribution of 10 cells in a 100 ms time bin. The parameter of the theoretical probability distribution was inferred from the data with the method of moments, which is equivalent to the maximum-likelihood estimation in this case. It is clear from the observation of the plots (especially the right one, with a logarithmic y-axis) that the experimental spike count distribution is not Poisson.

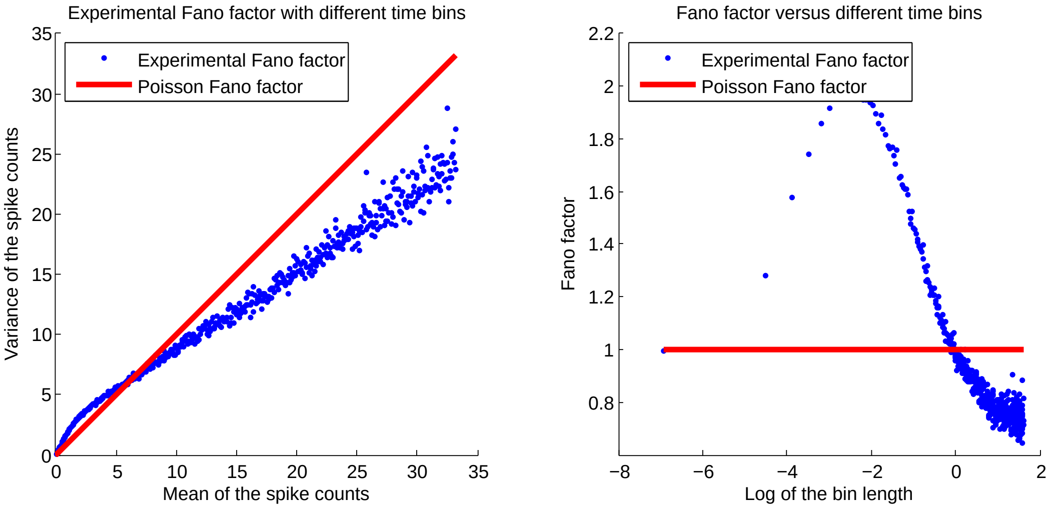

Figure 6: Fano factor of a ganglion cell spike train.

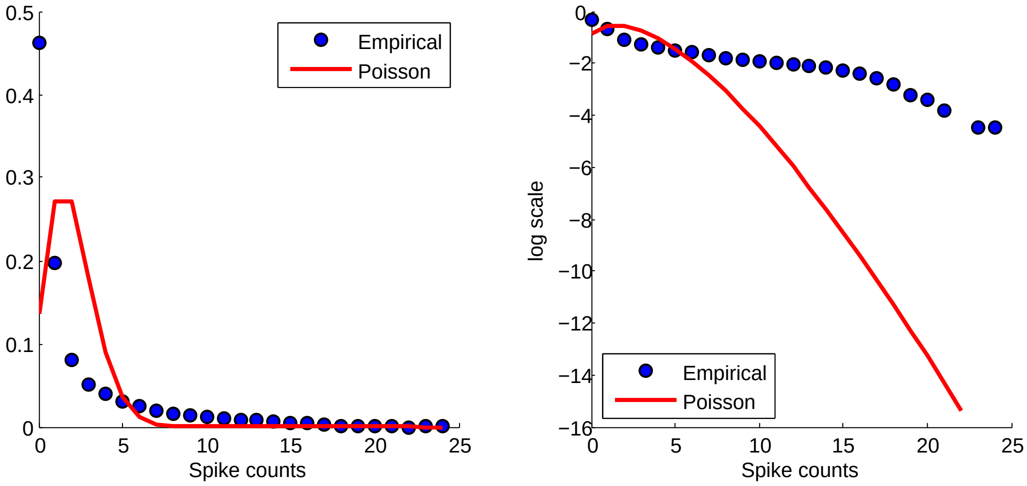

Figure 7: Experimental and Poisson spike count distributions.

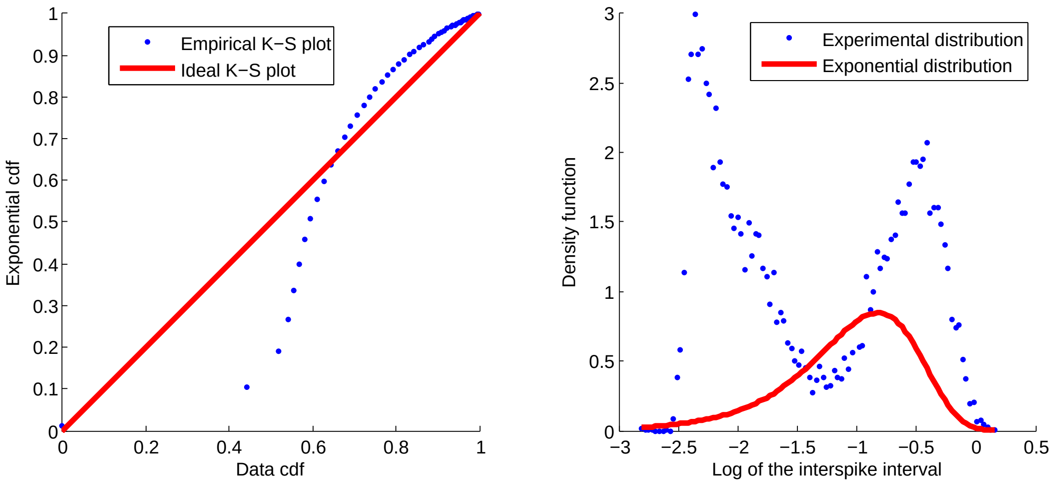

Figure 8: Left : Kolmogorov-Smirnov plot. Right : ISI distribution of a ganglion cell spike train.

The Kolmogorov-Smirnov test

The *Kolmogorov-Smirnov test* (KS-test) is a statistical test to check whether data follows a specified theoretical probability distribution. This test yields a plot of the empirical cumulative distribution function (cdf) of the data versus the cdf of the theoretical probability distribution. If the data is actually resulting from the specified probability distribution, the points in the plot would lay on a 45 ° line.

In Figure 8 (left), the KS plot of the experimental ISI data of a particular cell is shown. The theoretical cdf of the exponential distribution is $F(x) = 1 - e^{-\lambda x}$, where λ is estimated with the method of moments.

The interspike interval distribution

The *interspike interval (ISI) distribution* is defined as the probability distribution of the delay between two successive spikes. In the case of a homogeneous Poisson process with parameter λ, the ISI distribution is an exponential distribution with parameter λ. We computed the experimental ISI distribution of cells with the experimental data. We also computed an estimate of the rate parameter of the exponential distribution with the method of moments. In this case, it is equivalent to the maximum-likelihood estimation. The experimental and fitted exponential probability densities functions are shown in Figure 8 (right). The x-axis is a logarithmic time-scale. Here again, the discrepancy between the empirical and theoretical distribution is very important.

Time correlations

For a Poisson process, the number of spikes in two disjoint time intervals are independent (it is the *independent increments property*). In particular, the correlation between the spike count distributions in two successive time bins of same length is 0. We computed this empirical correlation with the same experimental data, and for the same cell. More specifically, we computed the correlation between $N([\,(n-1)T, nT\,])$ and $N([\,nT, (n+1)T\,])$ with respect to

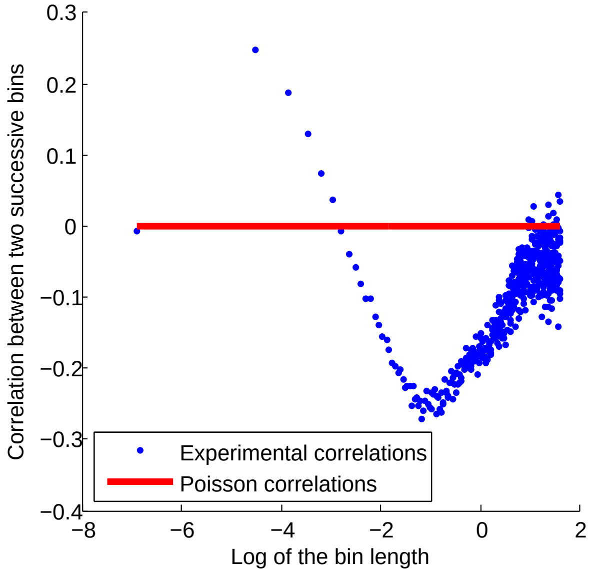

Figure 9: Correlations between two successive time bins.

n, with different values of T. The results are shown in Figure 9. We can observe important time correlations, which a good point process model should be able to capture.

Conclusion

In conclusion, these statistical tests show that the Poisson process is not a satisfactory model for stationary spike trains of retinal ganglion cells. It emphasizes the necessity of finding a better model.

7.3 The geometric distribution

In order to find a better point process model than the Poisson process, we consider spike count distributions of cells over different time bins. We would like to obtain a theoretical discrete probability distribution which fits well the data. We tried to test, instead of the Poisson distribution, the *geometric distribution*. It is defined by :

$$P(X=n)=\frac{1}{1+K}\left(\frac{K}{1+K}\right)^n.$$

where $K \in [\,0,1\,]$ is a parameter. Its probability generating function is :

$$G(x)=\frac{1}{1+K(1-x)}.$$

We used a standard maximum-likelihood procedure to infer the parameter of the geometric distribution from the data. Here, it happens to be equivalent to the method of moments : the estimated K is the mean of the empirical spike count distribution. One-dimensional fits of this model are shown in Figure 10. Here, spike count distributions are considered over 10 cells in a 100 ms time bin.

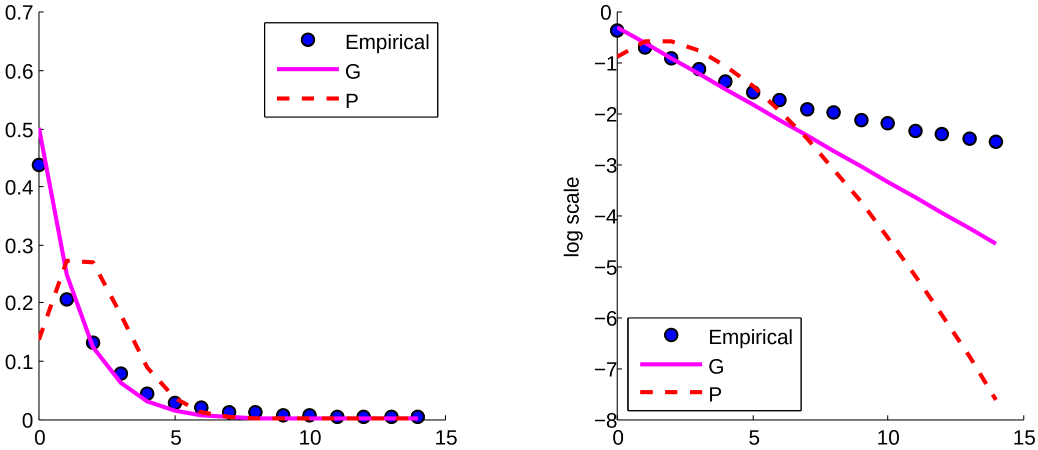

Figure 10: Experimental, Poisson and geometric spike count distributions.

The geometric distribution seems to be a better model than the Poisson distribution. However, an improvement is possible by considering a more general distribution : the negative multinomial distribution.

7.4 The negative multinomial process

The negative multinomial distribution

The geometric distribution is a particular case of the *negative multinomial* distribution. The latter's pgf is :

$$G(x) = \left(\frac{1}{1 + K(1-x)} \right)^{\kappa},$$

where $\kappa > 0$ is a parameter of the distribution. The pmf can be computed, but its analytical expression is a bit complicated. It is much more convenient to manipulate this distribution with its pgf. Fits of this model given by a maximum-likelihood estimation are shown in Figure 11. This distribution seems to capture very well correlations between cells.

The negative multinomial process

In order to define a point process model extending the negative multinomial distribution, we tried the following expression for the pgfl :

$$G(h) = \left(\frac{1}{1 + \int K(t)(1-h(t))\,\mathrm{d}t} \right)^{\kappa}.$$

Here, h is a measurable function $\mathbb{R} \to [\,-1, 1\,]$, which is equal to 1 if $|t|$ is large enough, and $K(t)$ is a measurable function $\mathbb{R} \to [\,0, 1\,]$. When deriving the pgf of the spike count distribution in a time window (using a piecewise constant h), it is easy to check that we obtain the pgf of the negative multinomial distribution. It can be generalized in several dimensions (as a multidimensional point process) :

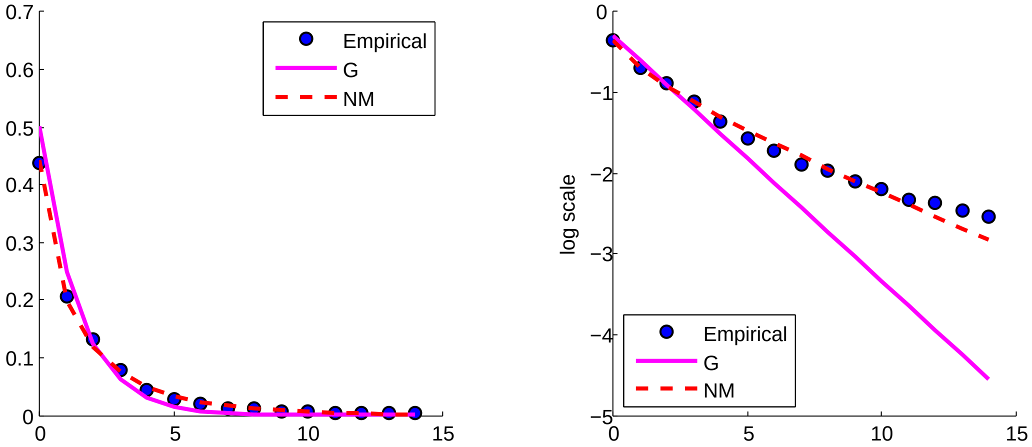

Figure 11: Experimental, geometric and NM spike count distributions.

$$G(h_1,\dots,h_n) = \left(\frac{1}{1+\sum_i \int K_i(t)(1-h_i(t))\,\mathrm{d}t}\right)^{\kappa}.$$

Theorem 25 in Appendix C proves that this process is well defined : we call it the **negative multinomial (NM) process**.

The PNM process

The NM process is not stationary, but one can slightly alter it in order to obtain a stationary process. If G is the pgfl of a point process satisfying certain conditions, then the cluster process obtained by the composition of an homogeneous Poisson process and that process yields a stationary point process, the pgfl of which is :

$$\widetilde{G}(h) = \exp\left(\lambda \int (G(h(\cdot - t)) - 1)\ \mathrm{d}t\right).$$

It is immediate to check that this process is stationary. In this particular case, the resulting point process is called the *Poisson-negative multinomial (PNM) process*, and its pgfl is :

$$G(h_1,\dots,h_n) = \exp\left(\lambda\left(\int\left(\left(\frac{1}{1+\sum_i \int K_i(t)(1-h_i(t-u))\,\mathrm{d}t}\right)^{\kappa} - 1\right)\ \mathrm{d}u\right)\right).$$

7.5 The PNMP process

Goals

The NM process is much better than the Poisson process. But, for certain cells, it fails to model efficiently spike count distributions in large time windows, as shown in Figure 12 (a single cell in a 100 ms time bin). The goal is then to improve the NM process in order to enhance the results for single cells.

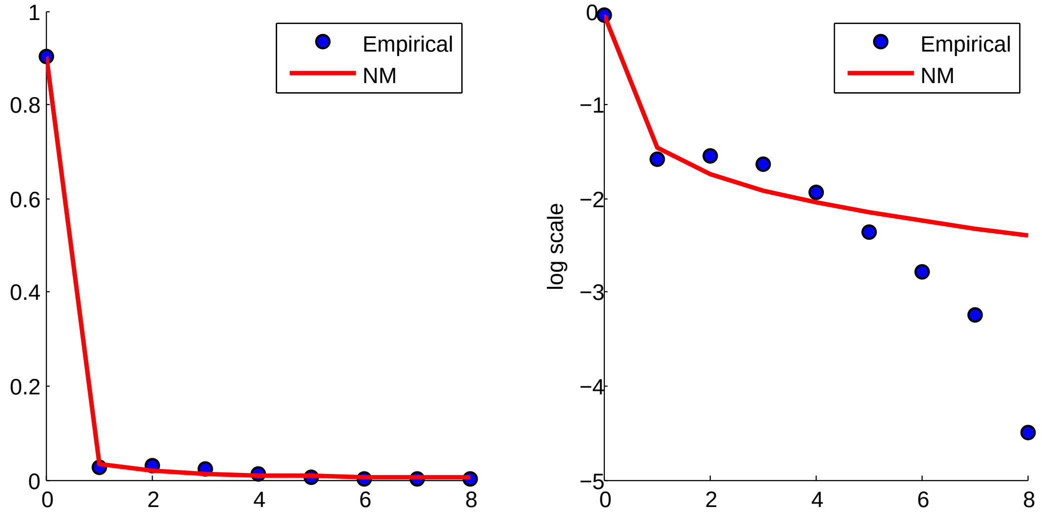

Figure 12: Experimental and NM spike count distributions for a single cell.

The PNMP process

Our idea to improve the PNM process is to right-compose it with an inhomogeneous Poisson process (using the same cluster process technique as for the PNM process). The resulting process is called the PNMP (Poisson-negative multinomial-Poisson) process, and its pgfl is :

$$G(h_1,\ldots,h_n) = \exp\left(\lambda\int\left(\left(\frac{1}{1+\sum_i\int K_i(t)\left(1-\exp\left(\int\mu_i(u)(h_i(u-t-v)-1)\,\mathrm{d}u\right)\right)\,\mathrm{d}t}\right)^{\kappa}-1\right)\mathrm{d}v\right).$$

Results obtained with this model are presented in the next section.

8 Results of the PNMP model

8.1 Implementation details and parameters

Implementation of the model

The PNMP model was coded in *Matlab* in a discretized version. More precisely, the probability generating function of the multidimensional spike count distribution of the grid (see Figure 5) associated to the PNMP model was implemented. This discretization step is very easy thanks to probability generating functionals : continuous variables of the pgfl (the h_i's) are replaced by piecewise-constant functions : their values on the intervals are the parameters of the associated discrete pgf.

Parameters of the discretized model

Also, continuous parameters (the K_i's and μ_i's) are replaced by piecewise-constant functions, which values are the parameters of the discretized model. Continuous integrals are thus replaced by discrete convolutions. The step of the discretization is fixed to 5 ms : it is also the size of the bin in the grid. The number of intervals for both parameter sets (K_i's and μ_i's) are fixed to 3. The total number of parameters to fit in our implementation of the model is then $2 + 6N_c$ (for each cell, 3 values for both K_i and μ_i, plus λ and κ).

Parameters of the grid

Our *Matlab* implementation of the maximum pseudo-likelihood algorithm was then used with this model on the available experimental data. The number of bins and cells in the grid, and the marginals to fit, were yet to be chosen. We tried several sets of parameters ; in general, we chose between 4 and 20 bins, and between 2 and 5 cells.

Choice of marginals to fit

About the marginals to fit in the inference algorithm, we chose them so that the model could capture both spike count distributions of single cells over a long time (the time course of the grid), and correlations between cells. Therefore, we found that a reasonable choice of marginals consisted in the combination of two types of marginals. First, one marginal for each cell consisting of the union of all bins of the cell (total number of spikes for the cell in the grid). Second, one *bidimensional* marginal for each pair of cells : the joint spike count distribution of both cells during the full time course of the grid. If N is the number of cells in the grid, then the number of marginals is $N(N + 1)/2$.

Complexity issues

After these parameters were fixed, we were able to run the optimization algorithm on several sets of cells. It took about one minute on a recent multiprocessor Apple computer for the algorithm to terminate in the case of two cells, and several minutes for up to five cells. As we wanted to run the algorithm for every possible choice of pairs, triplets, quintuples, etc. of cells, it took several days to obtain all results. For greater numbers of cells, both combinations number and the complexity of single runs were far too high to obtain results in a reasonable amount of time. We also couldn't find an efficient method to parallelize the implementation in *Matlab*. The different contributions of the pseudo-likelihood function (one per marginal) may have been parallelized, but parallelization toolboxes in *Matlab* seemed not efficient enough to yield significant reductions of time runs of the algorithm.

8.2 Pairs and triplets of cells

Overview

In order to test the PNMP model with respect to the data, we ran the maximum pseudo-likelihood algorithm on every pair and triplet of cells. As there were 40 cells in our recordings, there were 780 pairs and 9880 triplets. For each pair or triplet, we checked whether one-dimensional fitted marginals (spike count for each of the two cells) were visually close to empirical marginals. We also checked that covariances between cells were well captured.

We sorted the pairs and triplets according to the score of the model. This score was the maximum of the relative errors (between empirical and theoretical values) of marginals' variances and covariances. Figure 13 displays the distribution of scores among pairs and triplets (only errors less than 100% are shown). A few pairs or triplets give very bad results

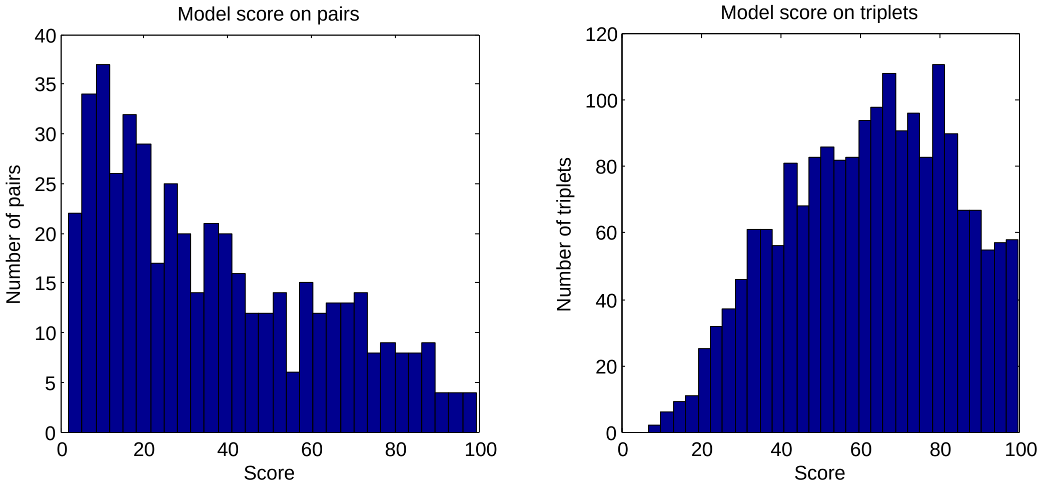

Figure 13: Distribution of model scores among all pairs and triplets of cells.

with the PNMP model : it is very often due to negative covariances which the model currently fails to capture. It is a known weakness of the model which is discussed later in this section.

Results : pairs and triplets of cells

We now give some unidimensional marginals of pairs and triplets of cells giving good results (see Figures 14 and 16), along with relative errors of their variances and covariances (see Figure 15 and 17). The bin length is 100 ms here (20 bins of 5 ms each).

Discussion

For those pairs and triplets, we can observe that the PNMP model gives good results. Marginals are visually well captured, and relative errors of variances and covariances are very small. For large values of spike counts (5 for cell 32 in the first pair, for example), we can observe systematic overpredictions of the model. It may be an artifact due to undersampling. Indeed, our experimental recordings of spike trains last about one hour, so there are about 30k instances of the grid (100 ms each). When experimental spike count distributions values are less than -4 in a logarithmic scale, it means that a few samples only are taken into account. The actual value, which would be obtained with infinitely-long recordings, may be higher, and closer to the model prediction.

As we can observe in the distribution of relative errors, not all cells give such interesting results. A lot of pairs and triplets have large relative errors for their covariances, even if marginals are very often well captured.

8.3 Weaknesses of the model

Negative covariances

An important remark is that even if we observe that some pairs of cells have negative empirical covariances, the model can't capture negative covariances. Indeed, theoretical covariances between cells can be computed for the PNMP model. It can be shown that they are either

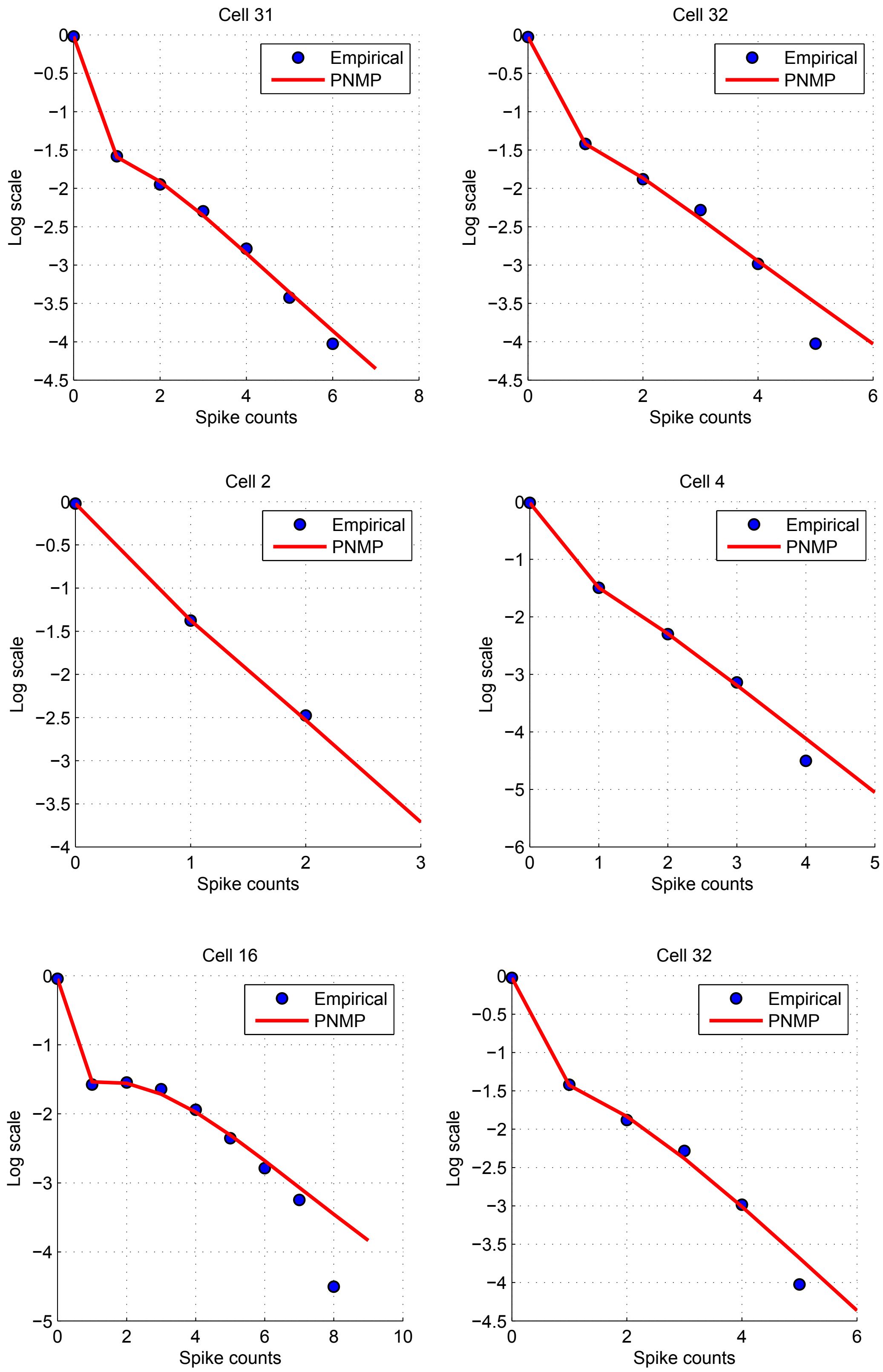

Figure 14: Results of three pairs : unidimensional marginals (spike count in a 100 ms bin).

Cells	Empirical	Model	Error
31	0.1502	0.1518	1.07%
32	0.1493	0.1532	2.56%
31-32	0.0183	0.0178	2.81%
2	0.0530	0.0536	1.07%
4	0.0572	0.0577	0.97%
2-4	0.0020	0.0020	2.86%
16	0.6733	0.7150	6.19%
32	0.1493	0.1492	0.12%
16-32	0.0033	0.0033	1.57%

Figure 15: Results of three pairs : variances and covariances.

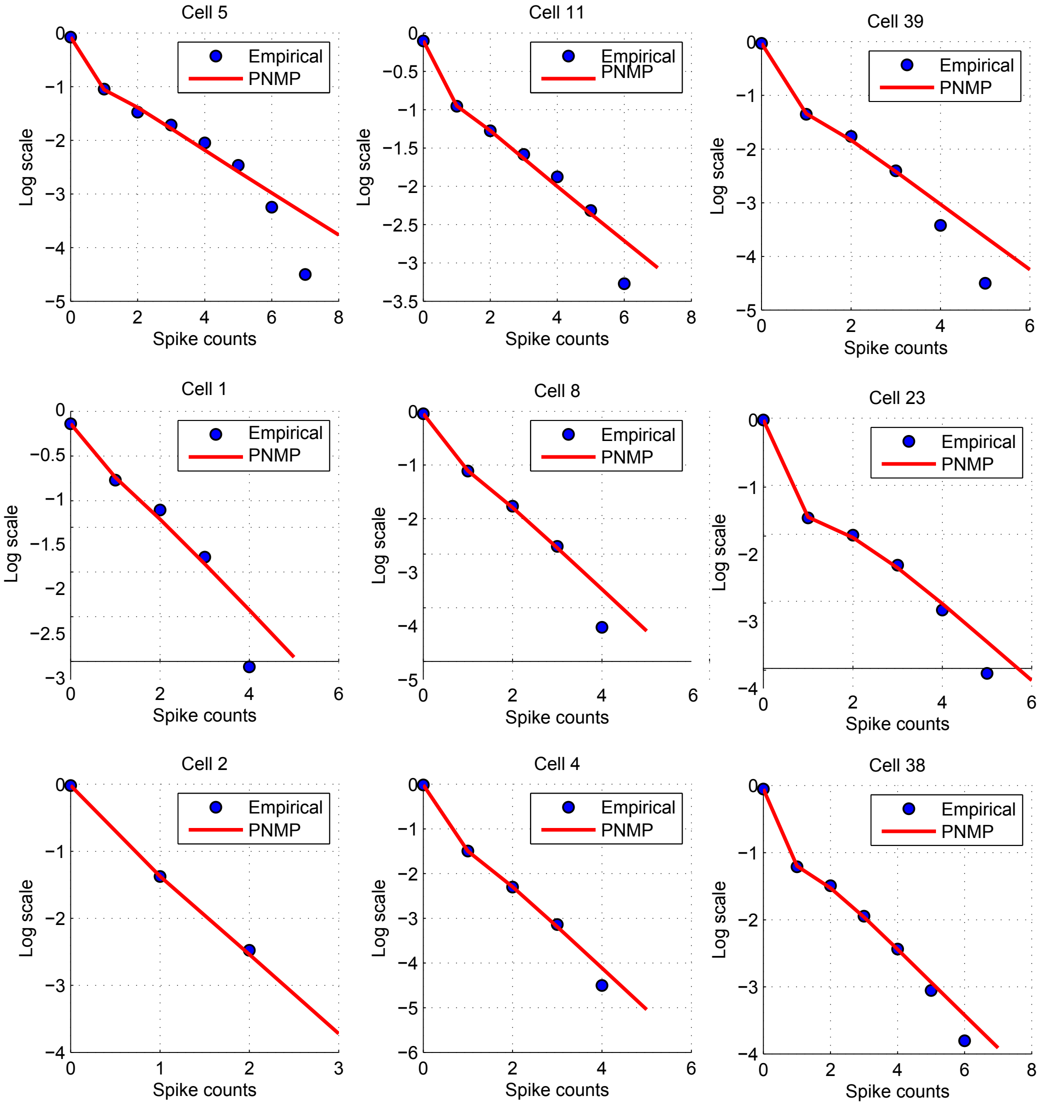

Figure 16: Results of three triplets : unidimensional marginals (spike count in a 100 ms bin).

Cells	Empirical	Model	Error
5	0.5761	0.5788	0.48%
11	0.7692	0.8307	7.99%
39	0.1471	0.1539	4.66%
5-11	0.1906	0.1754	7.96%
5-39	0.0392	0.0430	9.60%
11-39	0.0507	0.0593	16.93%
1	0.5553	0.6160	10.93%
8	0.1601	0.1638	2.35%
23	0.1801	0.1898	5.44%
1-8	0.0464	0.0381	17.88%
1-23	0.0291	0.0297	2.27%
8-23	0.0102	0.0092	10.14%
2	0.0530	0.0534	0.76%
4	0.0572	0.0575	0.50%
38	0.3458	0.3594	3.96%
2-4	0.0020	0.0021	4.02%
2-38	0.0082	0.0086	4.63%
4-38	0.0100	0.0079	21.31%

Figure 17: Results of three triplets : variances and covariances.

all positive, either all negative, depending on the model's parameters. The model should be improved in order to capture both positive and negative covariances between cells.

Not enough parameters to capture all covariances

There are $\alpha N_c + 2$ parameters in the PNMP model, where N_c is the number of cells. In order to capture correlations between all pairs of cells, more parameters may be required. This weakness has no effect when there are two or three cells in the model, as in the presented results above. But some bad results we obtained with more cells may indicate that this is an important problem of our model. Some leads to overcome these issues will be explored after the end of this internship.

Conclusion

During this internship, we designed a new theoretical approach for understanding the statistical properties of spike trains. It lies in the framework of point processes theory, and makes use of the probability generating functional. Comparison of models and data was conducted through discretization of the problem, leading to statistical inference of spike count distributions in several temporal bins. The latter was performed with a maximum pseudo-likelihood algorithm, implemented in *Matlab* during this internship.

We applied these tools to experimental data coming from the retina. The goal was to find an efficient point process model for multiple retinal ganglion cell spike trains. Noting first that these spike trains were not Poisson at all, we progressively designed a better model (the PNMP model) which was thought to give interesting results. For a reasonable proportion of cells, the model captures correlations between cells well. According to previous work emphasizing the role of correlations in the statistical structure of retinal ganglion cell spike trains, it may indicate that the PNMP process is an efficient model for these spike trains.

A few weaknesses of the model were highlighted, and some improvements will be still explored after the end of this internship. Nevertheless, early results obtained with the PNMP model show that the approach developed during those few months using probability generating functionals is quite powerful. It seems to be an efficient complement of other approaches already explored, and may help us to understand the neural code in the retina and in the brain.

Mathematical appendices

A Probability generating functions

A.1 Definition

Let $(X_1, \ldots, X_D)$ be a non-negative random variable over $\mathbb{N}^D$, and $P = (a_{n_1,\ldots,n_D})$ its probability mass function (pmf). By definition, $a_{n_1,\ldots,n_D} = P(X_1 = n_1, \ldots, X_D = n_D)$.

Definition 1 (Probability generating function).
The probability generating function *(pgf) of the distribution law P is the function defined as :*

$$G(z_1, \ldots, z_D) = \sum_{n_1,\ldots,n_D \geqslant 0} a_{n_1,\ldots,n_D} z_1^{n_1} \ldots z_D^{n_D},$$

where $(z_1, \ldots, z_D)$ is in the D-dimensional closed complex unit ball $\overline{\mathcal{D}}(0,1)^D$. G is $\mathcal{C}^\infty$ on $\mathcal{D}(0,1)^D$.

A.2 Marginals

The random variable $X = (X_1, \ldots, X_D)$ is a measurable function $(\Omega, \mathcal{A}, P) \to \mathbb{N}^D$, where $(\Omega, \mathcal{A}, P)$ is a probability space.

Let $\emptyset \subsetneq I \subset \{1, \ldots, D\}$ be a set of indices. A *marginal* is the distribution of probability of $X_I = \sigma_{|I|} \circ \pi_I \circ X$, where π_I is the projection from $\mathbb{N}^D$ onto the subspace of $\mathbb{N}^D$ corresponding to the indices I, and $\sigma_k : \mathbb{N}^k \to \mathbb{N}$ is the sum function. There are $2^D - 1$ different marginals (one for each I).

The following proposition shows that marginalizations with pgfs are extremely easy. It is one of the main advantages of using probability generating functions.

Proposition 2 (Marginalizations with pgfs).
The pgf of $(X_i)_{i \in I}$, where $I \subset \{1, \ldots, D\}$, is equal to the function $(z_{i \in I}) \mapsto G(z_1, \ldots, z_D)$ with $z_i = 1$ for all $i \notin I$.

B Inverse transform of a probability generating function using a Discrete Fourier Transform

Probability generating functions are very useful when manipulating discrete multidimensional distributions of probability, in particular when one wants to compute marginals. In this appendix, we give an algorithm for computing an approximation of the first N^D values of a probability mass function over $\mathbb{N}^D$ from its probability generating function. It is based on a Fast Fourier Transform and thus requires $O(DN^D \log N)$ operations.

B.1 Discrete Fourier Transform

Definition 3 (Discrete Fourier Transform).
Let $\mathbf{x} = (x_0, \ldots, x_{N-1}) \in \mathbb{C}^N$*, with* $N \geqslant 1$*. The* Discrete Fourier Transform *(DFT)* $\hat{\mathbf{x}} = (\hat{x}_0, \ldots, \hat{x}_{N-1}) \in \mathbb{C}^N$ *of* $\mathbf{x}$ *is defined as :*

$$\hat{x}_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2ikn\pi/N}.$$

The map $\mathbf{x} \in \mathbb{C}^N \mapsto \hat{\mathbf{x}} \in \mathbb{C}^N$ is a linear transformation, so we can write $\hat{\mathbf{x}} = U_N \mathbf{x}$ where U_N is a $N \times N$ complex matrix. We have :

$$U_N = \frac{1}{\sqrt{N}} F_N, \qquad \text{with } F_N = (e^{-2ikl\pi/N})_{(k,l)\in\{0,\ldots,N-1\}^2}.$$

F_N is called the *Vandermonde matrix* associated with the values $(1, e^{-2i\pi/N}, \ldots, e^{-2i(N-1)\pi/N})$.

The matrix U_N is symmetrical, and $U^{-1} = \overline{U}$, so U_N is a unitary matrix, and the DFT is a unitary transformation. Therefore, we have the following properties :

- $|\det U_N| = 1$,
- U_N is isometric : $\|\hat{\mathbf{x}}\|_2 = \|U_N \mathbf{x}\|_2 = \|\mathbf{x}\|_2$.
- Developing the previous equality leads to the *Parseval equality* :

$$\sum_{n=0}^{N-1} |x_n|^2 = \sum_{n=0}^{N-1} |\hat{x}_n|^2 .$$

- We also have $\|U_N\|_2 = 1$.
- If $\mathbf{y}$ is the DFT of $\mathbf{x}$, that is :

$$y_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2ikn\pi/N},$$

then we have the inverse relation (called *Inverse Discrete Fourier Transform*, or IDFT), coming from $U_N^{-1} = \overline{U}_N$:

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{2ikn\pi/N}.$$

Discrete Fourier Transform can be generalized to the multidimensional case. We use vector notation to obtain compact formulas. If $\mathbf{n} = (n_1, \ldots, n_D)$, with $D \geqslant 2$, and $\mathbf{m} = (m_1, \ldots, m_D)$ are two elements of $\mathbb{N}^D$, then products, divisions, and order relations are to be performed component-wise. We also use the notation $|\mathbf{n}| = \sum_{i=1}^{D} n_i$, $\mathbf{0} = (0, \ldots, 0)$, $\mathbf{1} = (1, \ldots, 1)$.

Definition 4 (Multidimensional Discrete Fourier Transform).
Let $\mathbf{N} = (N_1, \ldots, N_d)$ *and* $\mathbf{x} = \{\, x_{\mathbf{n}} \mid \mathbf{0} \leqslant \mathbf{n} \leqslant \mathbf{N}\} \in \mathbb{C}^{|\mathbf{N}|}$*. The* Multidimensional Discrete Fourier Transform $\hat{\mathbf{x}} = \{\, \hat{x}_{\mathbf{n}} \mid \mathbf{0} \leqslant \mathbf{n} \leqslant \mathbf{N}\} \in \mathbb{C}^{|\mathbf{N}|}$ *of* $\mathbf{x}$ *is defined as :*

$$\hat{x}_{\mathbf{n}} = \frac{1}{\sqrt{\prod_{i=1}^{d} N_i}} \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}-\mathbf{1}} x_{\mathbf{k}} e^{-2i\pi \mathbf{n}\cdot(\mathbf{k}/\mathbf{N})}.$$

B.2 Computing a probability mass function from a probability generating function using a DFT

Let X be a random variable over $\mathbb{N}$, $P = (a_n)_{n\in\mathbb{N}}$ its pmf and G its pgf. In this section, we give an algorithm for computing an approximation of the first N values $a_0, \ldots, a_{N-1}$ of P from G.

Theorem 5.
Let $(a_n)_{n\in\mathbb{N}}$ be a non-negative sequence such that the series $\sum a_n$ is convergent, and $G(z) = \sum a_n z^n$ defined in $\overline{\mathcal{D}}(0,1)$. Let $\mathbf{x} = (x_n)_{n\in\{0,\ldots,N-1\}}$ with :

$$x_n = G(e^{2in\pi/N})/\sqrt{N},$$

and $\hat{\mathbf{x}} = (\hat{x}_n)$ the DFT of $\mathbf{x}$. Then we have, for $N \geqslant 1$, with $R_N = \sum_N^\infty a_n$:

$$\max_{n\in\{0,\ldots,N-1\}} |\hat{x}_n - a_n| \leqslant R_N, \quad \textit{and} \quad \lim_{N\to\infty} \max_{n\in\{0,\ldots,N-1\}} |\hat{x}_n - a_n| = 0.$$

Proof. Let $N \geqslant 1$, $G_N(z) = \sum_{k=0}^{N-1} a_k z^k$, and $y_n = G_N(e^{2in\pi/N})/\sqrt{N}$.

We see that $(y_0, \ldots, y_{N-1})$ is the IDFT of $(a_0, \ldots, a_{N-1})$, so $a_n = \hat{y}_n$. From the Parseval equality, we have :

$$\sum_{n=0}^{N-1} |\hat{x}_n - a_n|^2 = \sum_{n=0}^{N-1} |\hat{x}_n - \hat{y}_n|^2 = \sum_{n=0}^{N-1} |x_n - y_n|^2 .$$

Besides, we have :

$$|x_n - y_n| = \left| \frac{1}{\sqrt{N}} \sum_{k=N}^{\infty} a_k e^{2ikn\pi/N} \right| \leqslant \frac{R_N}{\sqrt{N}},$$

where $R_N = \sum_N^\infty a_n$. Thus :

$$\begin{aligned} \max_{n\in\{0,\ldots,N-1\}} |\hat{x}_n - a_n| &\leqslant \sqrt{\sum_{n=0}^{N-1} |\hat{x}_n - a_n|^2} \\ &\leqslant \sqrt{\sum_{n=0}^{N-1} |x_n - y_n|^2} \\ &\leqslant \sqrt{\sum_{n=0}^{N-1} \left(\frac{R_N}{\sqrt{N}}\right)^2} \\ &\leqslant R_N. \end{aligned}$$

Since the remainder of a convergent series always tends to 0, we conclude that :

$$\lim_{N\to\infty} \max_{n\in\{0,\ldots,N-1\}} |\hat{x}_n - a_n| = 0.$$

□

We have a similar result for D-dimensional probability distributions.

Theorem 6.
Let $(a_{\mathbf{n}})_{\mathbf{n}\in\mathbb{N}^D}$ be a family in $\mathbb{R}_+$ such that the summable family $\sum a_{\mathbf{n}}$ is convergent, and :

$$G(z_1,\dots,z_D)=\sum a_{\mathbf{n}}z^{\mathbf{n}},$$

defined in $\overline{\mathcal{D}}(0,1)^D$. Let $\mathbf{x}=(x_{\mathbf{n}})_{\mathbf{n}\in\{0,\dots,N-1\}^D}$ with :

$$x_{\mathbf{n}}=G(e^{2in_1\pi/N},\dots,e^{2in_D\pi/N})/N^{D/2},$$

and $\hat{\mathbf{x}}=(\hat{x}_{\mathbf{n}})$ the multidimensional DFT of $\mathbf{x}$. Finally, let $R_N=\sum_{n_i\geqslant N}a_{\mathbf{n}}$. Then we have, for all $N\geqslant 1$:

$$\max_{\mathbf{n}\in\{0,\dots,N-1\}^D}|\hat{x}_{\mathbf{n}}-a_{\mathbf{n}}|\leqslant R_N,\quad \textit{and}\quad \lim_{N\to\infty}\max_{\mathbf{n}\in\{0,\dots,N-1\}^D}|\hat{x}_{\mathbf{n}}-a_{\mathbf{n}}|=0.$$

An algorithm to get the N^D first values of a distribution of probability from its pgf simply consists in computing the Discrete Fourier Transform of the D-dimensional matrix :

$$(M_{n_1,\dots,n_D})=G(e^{2in_1\pi/N},\dots,e^{2in_D\pi/N})/N^{D/2},$$

where, $\forall i$, $0\leqslant n_i\leqslant N-1$. The Discrete Fourier Transform can be computed using the Fast Fourier Transform algorithm in $O(DN^D\log(N^D))$ operations.

C Introduction to the theory of point processes on the real line

Point processes on the real line represent stochastic processes of random events in time. They are of interest in various disciplines such as telecommunication systems, queuing theory, computational neuroscience, and others. Here we give an introduction to the theory of point processes on the real line, presenting basic definitions and tools used to define and manipulate point process models. Then, some examples of models are given : the Poisson process, the NM process and the PNMP process.

C.1 Introduction

What are point processes ?

Point processes are the mathematical formalization of random processes which occurrences are discrete (or “atomic”) time or spatial events. For example, clients arriving in a queue, buses arriving at a bus stop, spikes occurring in a neuron, radioactive decays detected by a Geiger counter are examples of processes of random time events. They are intrinsically random, and each event occurrence can be seen as a single point. Epicenters of earthquakes, human settlements, locations of stars observed in the sky, positions of trees in a forest, are examples of processes of random spatial events.

All these observed phenomena can be modeled as point processes on $\mathbb{R}$ (time events) or $\mathbb{R}^d$ (spatial events). Actually, the theory can be generalized considering point processes on any metric space satisfying some technical conditions. The mathematical formalization of point processes relies on measure theory, Lebesgue integration, probability theory and stochastic processes.

Overview of the appendix

In this appendix, we give a brief introduction to the theory of point processes on the real line. Most proofs are not given, but references for the general theory are proposed.

We first present general definitions and practical characterizations of point processes on an abstract space (Section C.2). This is the only section where results are valid for point processes on any general space. A key result of this section is Theorem 13, which gives conditions to define properly a point process from its *probability generating functional.* It is used several times later in the document to define particular point process models. In Section C.3, we define the Poisson process. Then, in Section C.4, we present some extensions and particular classes of point processes, especially *cluster processes* and *multidimensional point processes.* We also give particular cases of these classes. Finally, we define the NM and the PNMP processes in Section C.5.

C.2 General definitions

A point process as a random simple counting measure

Point processes can be defined on any complete separable metric space X with a countable base, which form a very general class of spaces. Here, we are especially interested in point processes on the *real line* $\mathbb{R}$, which represent processes of random events in time. However, we will need to consider point processes on $\mathbb{R} \times \mathcal{K}$ where $\mathcal{K}$ is a finite set, and on any closed subset of $\mathbb{R}$.

Let (X, d) be a complete separable metric space (one can assume $X = \mathbb{R}$ to simplify matters), with $\mathcal{B}(X)$ its Borel σ-algebra. We will use the notation $|B| \equiv \mathrm{diam}(B)$ for any $B \in \mathcal{B}(X)$. Let $(\Omega, \mathcal{A}, P)$ be a probability space, where $\mathcal{A}$ is a σ-algebra over the space Ω, and P is a probability distribution over Ω.

Definition 7 (Counting measure).
A measure μ over X is a counting measure *if, for all bounded Borel set $B \in \mathcal{B}(X)$ (i.e. such that $|B| < \infty$), we have $\mu(B) \in \mathbb{N}$. It is a* simple counting measure *if $\forall t \in X, \mu(\{t\}) \in \{0, 1\}$.*

We denote by $\mathcal{M}(X)$ the set of all counting measures over X, and $\mathcal{M}^*(X)$ the subset of all simple counting measures. The space $\mathcal{M}(X)$ is equipped with the smallest σ-algebra such that, for any bounded Borel set $B \in \mathcal{B}(X)$, the application $\mu \in \mathcal{M} \mapsto \mu(B) \in \mathbb{N}$ is measurable.

Actually, one can shows that $\mathcal{M}(X)$ can be given a topology (the topology of weak convergence), which Borel σ-algebra is the σ-algebra of the previous definition. We denote by $B(\mathcal{M}(X))$ that σ-algebra.

Definition 8 (Point process).
A point process *is a measurable function $N : (\Omega, \mathcal{A}, P) \to (\mathcal{M}(X), B(\mathcal{M}(X)))$. It is a* simple point process *if $P(N \in \mathcal{M}^*(X)) = 1$. When $X = \mathbb{R}$, we talk about a* point process on the real line*.*

Note that a point process is finite on each bounded Borel set, that is, $N(\omega, B) < \infty$ if B is a bounded Borel set in X.

We will use the following abuse of notation : $N(\omega, B) \equiv N(\omega)(B)$ for any bounded Borel set B. Besides, we will use the notation $N(B) \equiv N(\cdot, B)$. According to the following proposition, it is a random variable over $\mathbb{N}$.

Proposition 9.
Let $N : (\Omega, \mathcal{A}, P) \to (\mathcal{M}(X), \mathcal{B}(\mathcal{M}(X)))$ be a function. It is a point process if and only if, for all $0 < s \leqslant t$, the application $N([\,s, t\,[) : \Omega \to \mathbb{N}$ is a random variable.

Proof. See [6], Theorem 9.1.VIII. □

From now on, we will consider only *simple* point processes.

Finite-dimensional distributions

The actual definition of a point process is not very convenient to define a particular point process model. In this section, we show that a point process can be defined via its finite-dimensional distribution, which is the joint probability distribution of events count in subsets of X.

Definition 10.
Let N be a point process, and $B_1, \dots, B_k$ be bounded Borel sets in $\mathcal{B}(X)$. The probability distribution of the multidimensional discrete random variable $(N(B_1), \dots, N(B_k))$ is called a finite dimensional distribution *of N.*

The following theorem gives conditions to define a point process with its finite-dimensional distribution.

Theorem 11 (Kolmogorov Existence Theorem for Point Processes).
For every $k \geqslant 1$, and every $B_1, \dots, B_k$ bounded Borel sets in $\mathcal{B}(X)$, let $p_k(B_1, \dots, B_k; \cdot)$ be a distribution of probability over $\mathbb{N}^k$. Then, they are the finite dimensional distributions of a point process, i.e. there exists a point process such that :

$$\forall n_1, \dots, n_k \in \mathbb{N},\ P(N(B_1) = n_1, \dots, N(B_k) = n_k) = p_k(B_1, \dots, B_k; n_1, \dots, n_k),$$

if and only if the following conditions are satisfied, where $B_1, \dots, B_{k+1}$ are any disjoint *bounded Borel sets in $\mathcal{B}(X)$:*

1. *for any permutation $(i_1, \dots, i_k)$ of the indices, we have :*

$$p_k(B_{i_1}, \dots, B_{i_k}; n_{i_1}, \dots, n_{i_k}) = p_k(B_1, \dots, B_k; n_1, \dots, n_k),$$

2. *we have :*

$$\sum_{r=0}^{\infty} p_{k+1}(B_1, \dots, B_k, B_{k+1}; n_1, \dots, n_k, r) = p_k(B_1, \dots, B_k; n_1, \dots, n_k),$$

3. *we have :*

$$\sum_{r=0}^{n} p_k(B_1, \dots, B_k; r, n-r, n_3, \dots, n_k) = p_{k-1}(B_1 \cup B_2, \dots, B_k; n, n_3, \dots, n_k),$$

4. *for any sequence (B_n) of bounded Borel sets such that $B_n \downarrow \emptyset$, we have $p_1(B_n; 0) \to 1$.*

Proof. The proof of this theorem can be found in [6], Theorem 9.2.X. □

Probability generating functionals

Probability generating functionals are the generalization for point processes of probability generating functions for discrete multidimensional probability distributions. They are very useful for defining and manipulating point processes.

Let $\mathcal{V}(X)$ be the set of all measurable functions $h : X \to [0,1]$ such that $1-h$ has a bounded support.

Definition 12.
Let N be a point process. The probability generating functional *(pgfl) of N is the function $G : \mathcal{V}(X) \to X$ defined by :*

$$G(h) = \mathbb{E}\left(\exp\left(\int_X \log h(t) N(\,\mathrm{d}t)\right)\right).$$

Here, the integral is with respect to the measure $N(\omega, \cdot)$ on X, so it is a random integral depending on $\omega \in \Omega$. The expectation is with respect to ω.

When N is a simple point process with time events $t_0 < t_1 < \ldots$, we have the following alternative formula for the pgfl of N :

$$G(h) = \mathbb{E}\left(\prod_i h(t_i)\right).$$

We can define a point process with a pgfl thanks to the following theorem, analogous to theorem 11, but in a certain way more elegant and natural.

Theorem 13 (Characterization of a point process from its pgfl).
Let $G : \mathcal{V}(X) \to X$ be a function. Then it is the pgfl of a point process if and only if :

1. *G(1) = 1,*

2. *for every piecewise-constant function h of the form :*

   $$1 - h = \sum_{i=1}^{k} (1 - z_i) 1_{B_i},$$

   with $|z_i| \leqslant 1$ and B_i bounded disjoint *Borel sets in X, the pgfl $G(h)$ reduces to the pgf of a discrete k-dimensional distribution of probability,*

3. *if $h_n \downarrow h \in \mathcal{V}(X)$ pointwise, then $G(h_n) \to G(h)$.*

Proof. The proof of this theorem can be found in [6], Theorem 9.4.V. □

A point process on the real line as a random time events sequence

In this paragraph, we assume that $X = \mathbb{R}$. Let N be a point process on the real line. For each $\omega \in \Omega$, $N(\omega)$ is a realization of the point process. It is a simple counting measure : if $t \in \mathbb{R}$ is such that $N(\omega, \{t\}) = 1$, it means that an event occurred at time t. Thus, $N(\omega)$ can be seen as a time events list $\ldots < t_{-1} < t_0 < t_1 < \ldots$. We formalize this in the following definitions.

Definition 14 (Counting process).
We define the counting process $N(t)$ *associated to the point process* N *as follows :*

$$N(t) = \begin{cases} N(]\,0,t\,]) & t > 0, \\ 0 & t = 0, \\ -N(]\,t,0\,]) & t < 0. \end{cases}$$

This stochastic process is an integer-valued, non-decreasing and right-continuous process. For $t > 0$, the integer $N(t)$ represents the number of events that occurred between times 0 and t.

We can now define the time events process and interarrival time process of a point process on the real line.

Definition 15.

1. *For each* $i \in \mathbb{N}$*, the* i-th time event *is the random variable over* $[\,0,+\infty\,]$ *defined by* $t_i = \inf\{\ t \in \mathbb{R} \mid N(t) \geqslant i\}$*.*
2. *For* $i \geqslant 1$*, the* i-th interarrival time *is* $\tau_i = t_i - t_{i-1}$*.*

From now on, we will consider only point processes on the real line.

C.3 The Poisson process

In this section, we give the definition of the Poisson process, which is the simplest non-trivial point process on the real line.

The homogeneous Poisson process

Definition 16 (Homogeneous Poisson process).
Let $\lambda > 0$*. The* homogeneous Poisson process N *with rate* λ *is defined by the following finite dimensional distributions :*

$$P\Big(N(]\,a_1,b_1\,]) = n_1,\dots,N(]\,a_k,b_k\,]) = n_k\Big) = \prod_{i=1}^{k} \frac{\big(\lambda(b_i-a_i)\big)^{n_i}}{n_i!} e^{-\lambda(b_i-a_i)}.$$

Its pgfl is :

$$G(h) = \exp\left(\lambda \int_{\mathbb{R}} (h(t)-1)\,\mathrm{d}t\right).$$

The verifications of the conditions of theorems 11 and 13 for the finite dimensional distributions and the pgfl of the Poisson process, and the proof that the two definitions are equivalent, can be found in [5] and [6]. We give now some important properties of the Poisson process.

Proposition 17.
Let N *be a Poisson process with constant rate* λ*.*

1. *The number of events $N\big(\,]\,a,b\,]\,\big)$ in an interval $]\,a,b\,]$ follows a Poisson distribution with parameter $\lambda(b-a)$.*

2. *This distribution is stationary, i.e. it depends only on the length $b-a$ of the interval.*

3. *The number of events in disjoint intervals are independent random variables (*independent increments property*).*

4. *The interarrival times are independent and identically distributed exponential random variables with parameter λ :*

$$P(\tau_i \leqslant T) = 1 - e^{-\lambda T}.$$

5. *The integer-valued random variable $N\big(\,]\,a,b\,]\,\big)$ has equal mean and variance :*

$$\mathbb{E}N\big(\,]\,a,b\,]\,\big) = \mathbb{V}N\big(\,]\,a,b\,]\,\big) = \lambda(b-a).$$

Intuitively, the homogeneous Poisson process is a totally random process : the probability of occurrence of a time event depends only on the occurrence of the previous time event, and not on the whole past of the process.

The inhomogeneous Poisson process

Definition 18 (Inhomogeneous Poisson process).
Let $\lambda(t) : \mathbb{R} \to \mathbb{R}_+$ be a positive integrable function. The inhomogeneous Poisson process *N with rate $\lambda(t)$ is defined by the following finite dimensional distributions :*

$$P\Big(N(\,]\,a_1,b_1\,]\,) = n_1, \ldots, N(\,]\,a_k,b_k\,]\,) = n_k\Big) = \prod_{i=1}^{k} \frac{\big(\Lambda(a_i,b_i)\big)^{n_i}}{n_i!} e^{-\Lambda(a_i,b_i)},$$

with :

$$\Lambda(a,b) = \int_a^b \lambda(t)\,\mathrm{d}t.$$

Its pgfl is :

$$G(h) = \exp\left(\int_{\mathbb{R}} \lambda(t)(h(t)-1)\,\mathrm{d}t\right).$$

The independent increments property is still valid for the inhomogeneous Poisson process, but it is not stationary anymore. The interarrival times are still independent but not identically distributed.

C.4 Extensions of point processes

Cluster process

A cluster process is a point process consisting of the superposition of two point processes : the centre process, which represent the centers of random clusters, and the independent component processes, which define the time events sequence inside each cluster. The formal definition can be found in [5], Definition 6.3.I. Here we only give a convenient way to define cluster processes as a composition of pgfl's.

Definition 19 (Cluster process).
Let N_c be a point process with pgfl G_c, and $N(\cdot \mid t)$ a family of point processes indexed by $t \in \mathbb{R}$, with pgfl's $G_m(\cdot \mid t)$ such that, for each $h \in \mathcal{V}(\mathbb{R})$, the function $t \mapsto G_m(h \mid t)$ is measurable on $\mathbb{R}$. If the formula $G(h) = G_c(G_m(h \mid \cdot))$ is well defined and finite for each h, then it is the pgfl of the cluster process *with centre process N_c and component processes $N(\cdot \mid t)$.*

The existence condition of the cluster process can also be expressed as :

$$\int_{\mathbb{R}} P\big(N(B \mid t) > 0\big) N_c(\,\mathrm{d}t) < \infty \quad P\text{-a.s., for every bounded Borel set } B \in \mathcal{B}(\mathbb{R}).$$

It is necessary so that the cluster process is a well-defined point process, i.e. the random counting measure is finite on each bounded Borel set. The following proposition gives a particular case for which a cluster process is well defined : the component processes are translated independent instances of a point process with bounded support.

Proposition 20.
A cluster process with centre process N_c (with pgfl G_c) and component processes $N_m(\cdot \mid t)$ (with pgfl $G_m(h \mid t)$) is well defined if the following sufficient conditions are satisfied :

1. *the component processes are translated independent instances of a single process with pgfl $G_m^0(h)$, i.e. we have :*

$$\forall t \in \mathbb{R}, G_m(h \mid t) = G_m^0(h(\cdot - t)),$$

2. *the pgfl $G_m^0(h)$ satisfies :*

$$\exists T > 0, \forall h \in \mathcal{V}(\mathbb{R}) \textit{ such that } \forall\, |t| \leqslant T, h(t) = 1, \textit{ we have } G_m^0(h) = 1.$$

Multidimensional point process

In this section, we give the definition of a multidimensional point process, which can be seen as a set of point processes on the real line. In the following, $\mathcal{K}$ is a finite set : $\mathcal{K} = \{1, \ldots, K\}$.

Definition 21 (Multidimensional point process).
A multidimensional point process *is a point process on $\mathbb{R} \times \mathcal{K}$.*

One can see a multidimensional point process as a list of random time events with a mark : (t_i, k_i) where $k_i \in \mathcal{K}$. Hence, each time event carries an extra information. Actually, this definition is a particular case of a *marked point process*, for which the mark space can be very general.

The following proposition shows that one can also see a multidimensional point process as a set of K point processes on the real line, $N_1, \ldots, N_K$, defined on the same probability space $(\Omega, \mathcal{A}, P)$.

Proposition 22.

1. *Let $N : (\Omega, \mathcal{A}, P) \to (\mathcal{M}(\mathbb{R} \times \mathcal{K}), \mathcal{B}(\mathcal{M}(\mathbb{R} \times \mathcal{K})))$ be a multidimensional point process on the real line with mark space $\mathcal{K}$. The* component processes *$N_1, \ldots, N_K : (\Omega, \mathcal{A}, P) \to (\mathcal{M}(\mathbb{R}), \mathcal{B}(\mathcal{M}(\mathbb{R})))$ are the point processes on the real line defined by :*

$$\forall \omega \in \Omega, \forall B \in \mathcal{B}(\mathbb{R}),\ N_i(\omega)(B) = N(\omega)(B \times \{i\}).$$

2. *Let $N_1, \dots, N_K : (\Omega, \mathcal{A}, P) \to (\mathcal{M}(\mathbb{R}), \mathcal{B}(\mathcal{M}(\mathbb{R})))$ be K point processes on the real line defined on the same probability space. Then we can define a multidimensional point process N with the following formula :*

$$\forall \omega \in \Omega, \forall B \in \mathcal{B}(\mathbb{R}),\ N(\omega)(B \times \{i\}) = N_i(\omega)(B).$$

With this construction, the N_i are the component processes of N.

Proof. The proof is easy and left to the reader. It relies on the fact that a measure over $\mathbb{R} \times \mathcal{K}$ can be seen as a set of K measures over $\mathbb{R}$, thanks to the formula $\mu(B) = \sum \mu(B_i \times \{i\}) = \sum \mu_i(B_i)$ (with evident notations). □

That equivalence can help us to define the pgfl of a multidimensional point process. We define $\mathcal{V}_K(\mathbb{R})$ as the set of all the measurable bounded multivalued functions $h : \mathbb{R} \to [\,0, 1\,]^K$, such that $1 - h$ has a bounded support.

Definition 23 (Pgfl of a multidimensional point process).
Let N be a multidimensional point process on the real line with mark space $\mathcal{K}$. Let $N_1, \dots, N_K$ be its component processes. The probability generating functional *of N is the function $\mathcal{V}_K(\mathbb{R}) \to \mathbb{R}$ defined by :*

$$G(h) = \mathbb{E}\left(\prod_{i=1}^{K} \exp\left(\int_{\mathbb{R}} \log h_i(t) N_i(\,\mathrm{d}t)\right)\right), \qquad \textit{where } h(t) = (h_1(t), \dots, h_K(t)).$$

In particular, we can observe that if the component processes are *independent*, the pgfl of N is simply the product of the pgfl's of the component processes : $G(h) = G_1(h_1) \cdots G_K(h_K)$.

We have an analogous of the theorem 13 allowing to define a multidimensional point process from its pgfl.

Theorem 24 (Characterization of a multidimensional point process from its pgfl).
Let $G : \mathcal{V}_K(\mathbb{R}) \to \mathbb{R}$ be a function. Then it is the pgfl of a multidimensional point process if and only if :

1. *G(1) = 1,*

2. *for every $h = (h_1, \dots, h_K)$ of the form :*

$$1 - h_j = \sum_{i=1}^{k} (1 - z_{ij}) 1_{B_{ij}},$$

   with $|z_{ij}| \leqslant 1$ and B_{ij} bounded disjoint *Borel sets in $\mathbb{R}$, the pgfl $G(h)$ reduces to the pgf of a discrete kK-dimensional distribution of probability,*

3. *if, for all j, the functions sequence $h_j^{(n)} \downarrow h_j \in \mathcal{V}_K(\mathbb{R})$ pointwise, then $G(h^{(n)}) \to G(h)$.*

Proof. The proof is similar to the proof of theorem 13. □

C.5 The PNMP process

In this section, we define the negative multinomial process, and the PNMP process. They are used as point process models for spike trains in Section 7.

The NM process

Definition 25.
Let $k_1, \dots, k_N : \mathbb{R} \to [\,0,1\,]$ *be integrable functions, and* $\kappa > 0$. *The* negative multinomial (NM) process *is defined by the following pgfl :*

$$G(h_1, \dots, h_n) = \left(\frac{1}{1 + \sum_i \int k_i(t)(1 - h_i(t))\,\mathrm{d}t} \right)^{\kappa}.$$

Proof. We use theorem 24 to prove that this pgfl defines a point process. There are three conditions to check.

1. If $\forall i, h_i(t) = 1$, then $G(h_1, \dots, h_N) = 1$.
2. If $1 - h_j = \sum_i (1 - z_{ij}) 1_{B_{ij}}$ where B_{ij} are bounded Borel sets in $\mathbb{R}$ and $z_{ij} \in \mathbb{C}$ such that $|z_i| \leqslant 1$, then :

$$\begin{aligned} G(h_1, \dots, h_N) &= \left(1 + \int_{\mathbb{R}} \left(\sum_{i,j} (1 - z_{ij}) 1_{B_{ij}} \right) k_j(t)\,\mathrm{d}t \right)^{-\kappa} \\ &= \left(1 + \sum_{i,j} (1 - z_{ij}) \int_{B_{ij}} k_j(t)\,\mathrm{d}t \right)^{-\kappa} \\ &= \left(1 + \sum_{i,j} p_{ij}(1 - z_{ij}) \right)^{-\kappa}, \end{aligned}$$

   which is the pgf of a negative multinomial distribution.
3. If $h_i^{(n)} \downarrow h_i \in \mathcal{V}(\mathbb{R})$ pointwise, then $(1 - h_i^{(n)}(t))k_i(t) \to (1 - h_i(t))k_i(t)$ for almost every $t \in \mathbb{R}$. Moreover, $0 \leqslant (1 - h_i^{(n)}(t))k_i(t) \leqslant k_i(t)$ since $\forall n \in \mathbb{N}, \forall t \in \mathbb{R}, h_i^{(n)}(t) \in [\,0,1\,]$. But k_i is integrable on $\mathbb{R}$, and the dominated convergence theorem states that $\int(1 - h_i^{(n)}(t))k_i(t)\,\mathrm{d}t \to \int(1 - h_i(t))k_i(t)\,\mathrm{d}t$, and $G(h_1^{(n)}, \dots, h_N^{(n)}) \to G(h_1, \dots, h_N)$.

Therefore, theorem 13 applies, and the existence of the NM process is proved. □

PNMP

We now define the PNMP model as a cluster process, composed of Poisson and NM processes.

Definition 26 (PNMP).
Let $k_1, \dots, k_N$ *and* $\mu_1, \dots, \mu_N : \mathbb{R} \to [\,0,1\,]$ *be integrable functions with bounded supports, and* $\kappa, \lambda > 0$. *The* PNMP *process is the multidimensional point process defined by the following pgfl :*

$$G(h_1, \dots, h_n) = \\ \exp\left(\lambda \int \left(\left(\frac{1}{1 + \sum_i \int k_i(t)\left(1 - \exp\left(\int \mu_i(u)(h_i(u - t - v) - 1)\,\mathrm{d}u\right)\right)\,\mathrm{d}t} \right)^{\kappa} - 1 \right) \mathrm{d}v \right).$$

It is formed by successive compositions (cluster processes) of an homogeneous Poisson process, the NM process, and an inhomogeneous Poisson process, hence the name PNMP process*.*

Proof. To prove that the PNMP is well defined, it is sufficient to check conditions of Theorem 20 for the two following compositions of cluster processes : the NM process composed with an inhomogeneous Poisson process with bounded support, and an homogeneous Poisson process with that process. But the conditions hold since the supports of the k_i's and μ_i's are bounded. □

References

[1] Abbott, L. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin 50*, 5-6 (1999), 303–304.

[2] Brette, B. *Modèles Impulsionnels de Réseaux de Neurones Biologiques.* PhD thesis, Ecole Doctorale Cerveau - Cognition - Comportement, December 2003.

[3] Brown, E., Barbieri, R., Eden, U., and Frank, L. Likelihood methods for neural spike train data analysis. *Computational Neuroscience: A Comprehensive Approach* (2003).

[4] Brown, E., Barbieri, R., Ventura, V., Kass, R., and Frank, L. The Time-Rescaling Theorem and Its Application to Neural Spike Train Data Analysis, 2002.

[5] Daley, D., and Vere-Jones, D. *An Introduction to the Theory of Point Processes.* Springer, 2003.

[6] Daley, D., and Vere-Jones, D. *An Introduction to the Theory of Point Processes, Volume II: General Theory and Structure.* Springer, 2007.

[7] Hodgkin, A., and Huxley, A. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol 117*, 4 (1952), 500–544.

[8] Hopfield, J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences of the United States of America 79*, 8 (1982), 2554–2558.

[9] Matlin, M., and Foley, H. *Sensation and perception.* Allyn and Bacon Boston, 1992.

[10] McCulloch, W., and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology 5*, 4 (1943), 115–133.

[11] Meister, M., Pine, J., and Baylor, D. Multi-neuronal signals from the retina: acquisition and analysis. *J Neurosci Methods 51*, 1 (1994), 95–106.

[12] Schneidman, E., Berry, M., II, R., and Bialek, W. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature 440*, 7087 (2006), 1007.

[13] Tkacik, G., Schneidman, E., Berry, M., and Bialek, W. Ising models for networks of real neurons. q–bio. *NC/0611072* (2006).

[14] Wikipédia. Action potential, May 2008. `en.wikipedia.org/wiki/Action_potential`.

[15] Wikipédia. Neuron, May 2008. `en.wikipedia.org/wiki/Neuron`.

[16] Wohrer, A. *Model and large-scale simulator of a biological retina, with contrast gain control.* PhD thesis, INRIA Sophia Antipolis, March 2008.