INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAG011

Thèse de doctorat

ENSAE

IP PARIS

Offline Contextual Bandit : Theory and Large Scale Applications

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École nationale de la statistique et de l'administration économique

École doctorale n°574 École doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 18/12/2023, par

OTMANE SAKHI

Composition du Jury :

Olivier Catoni Directeur de Recherche, (CREST)	Président
Benjamin Guedj Chargé de recherche (UCL & Inria)	Rapporteur
Emilie Kauffman Chargée de recherche, Univ. Lille, Inria (CRIStAL)	Rapporteuse
Julyan Arbel Chargé de recherche, Inria Grenoble (Statify)	Examinateur
Anna Korba Maître de conférences, Institut Polytechnique de Paris (ENSAE)	Examinatrice
Nicolas Chopin Professeur, Institut Polytechnique de Paris (ENSAE)	Directeur de thèse
David Rohde Chercheur, Criteo AI Lab	Invité

FMJH
FONDATION MATHÉMATIQUE
JACQUES HADAMARD

CREST
CENTER FOR RESEARCH
IN ECONOMICS AND STATISTICS

CRITEO
AI Lab

Acknowledgements

First and foremost, I want to express my sincere and deep gratitude to my two thesis supervisors, David and Nicolas. Your guidance and enthusiasm for research have been immensely helpful throughout these intense years, both in the development of this thesis and in building my research abilities.

David, thank you for taking me under your wing in the middle of my internship and your trust to prolong the adventure into a doctorate. Thank you for the human qualities that you have and the patience and optimism that characterize you. I am grateful for all the laughter, the research discussions we engaged, your attention to detail, the freedom you gave me to explore different research topics and the unending support you showed at the beginning of each new project.

Nicolas, a big thanks for accepting to be part of the adventure and for your gentle supervision. Your experience, broad knowledge and rigour ignited my research curiosity and inspired me to strive for excellence. Thank you for all the scientific discussions, your open-mindedness and the infinitely-many research ideas that a single thesis cannot cover. Thank you for always finding great collaborators to push our work and for your never ending trust in my capabilities. I am confident that this thesis only represents the beginning of a beautiful friendship and many future research collaborations.

I also want to thank Criteo for providing a stimulating environment, rich with interesting problems that inspire impactful research. Thanks to the entire CAIL and to the Reco research team in particular for all the moments shared. Thanks to all my colleagues and friends with whom I collaborated, and with whoever I could share a discussion, a coffee or a beer. Numerous special moments within Criteo made this whole experience much more enjoyable.

Thanks to Julyan Arbel, Olivier Catoni, Benjamin Guedj, Émilie Kaufmann and Anna Korba for accepting to be part of my committee. Your enthusiasm and interest in my work have been truly heartwarming. A special thanks to Benjamin and Émilie for accepting to review my dissertation; it is a great honor for me.

Finally, I want to thank my family and loved ones. Special thanks to my parents and my brother, who have always supported and encouraged me unwaveringly. A huge thanks to Ayman, Oussama and El Ghanjaoui for turning this experience into a pleasant journey and to Iness for adding beauty to my days.

Abstract

Abstract. Modern interactive systems shape our internet experience. From search to recommendation engines, these systems organise vast amounts of content, allowing users to efficiently find answers to their needs. The quality of user experiences within these systems can vary significantly, and the ability to provide users with relevant options at the right moment can greatly enhance both user satisfaction and the profitability of the businesses operating these systems. In recent years, there has been a concerted effort to leverage machine learning techniques to improve interactive systems by combining various signals. This thesis focuses on harnessing a specific type of signal: *user interaction logs*. These logs are uniquely valuable as they directly capture successes and failures in previous interactions. Nonetheless, the interactive nature of the logs makes their analysis more challenging compared to classical supervised learning problems. The Offline Contextual Bandit formalises an idealized version of this learning problem. It reduces the interaction logs to triplets of an observed context, an action made by the system and a reward received. These triplets are core to analyse the problem and to learn improved interactive systems. Notwithstanding recent advances, there remains significant challenges to learn decision systems with performance certificates and scale current approaches to real world problems.

Our first concern is being able to measure how well an interactive system will perform before it engages with the environment. Statistical learning theory focuses on studying the generalization ability of algorithms, and presents itself as the perfect candidate to answer this question. Historically, its tools were used to improve our understanding of the supervised learning paradigm, resulting in Empirical and Structural Risk minimization principles. More recently, statistical learning theory was adapted to learning from interaction logs, and resulted in the Counterfactual Risk minimization principle. This new objective captures the difficulties of learning from contextual bandit logs, but its application is limited to simple scenarios. In particular, the learning objective is non-convex, it cannot be accelerated with stochastic gradient methods, it introduces new hyperparameters that are difficult to tune and fails to provide performance certificates on the newly trained interaction systems. The first part of the thesis focuses on developing new statistical learning ideas to address these challenges. We reframe the Counterfactual Risk minimization using Distributionally Robust Optimization. This change of perspective allows us to improve the optimization procedure, to automatically calibrate hyperparameters while enjoying the same guarantees. Furthermore, we explore PAC-Bayesian learning, a statistical learning framework that provides a finer analysis of the generalization ability of algorithms. Using this paradigm, we build new strategies that require no hyperparameter tuning, that enable fast optimization and can provide strong guarantees on the performance of our interactive systems.

Another concern is to efficiently learn decision systems operating on massive action spaces. The second part of the thesis addresses this challenge, focusing primarily on large scale recommendation. Efficient learning in this case can be achieved by exploiting different signals and speeding up the optimization routine. Existing methods rely solely on the bandit signal: the log of the past successes and failures. However, non-bandit signal, such as collaborative filtering, can be extremely valuable. Building on this observation, we dedicate a chapter to develop a Bayesian approach to recommendation that combines both signals. We give proper computational tools to scale the learning to large datasets and prove empirically that the resulting systems enjoy improved recommendation quality.

Large scale recommender systems are updated frequently to match the ever-shifting interests of the users. The ability to perform these updates regularly relies on the efficiency of the optimization routine. When confronted with exceedingly large action spaces, these systems are constrained to the maximum inner product search (MIPS) structure for rapid query responses. Despite their prevalence in the industry, optimizing these systems with common learning objectives tend to be slow. Indeed, every gradient iteration scales at least linearly with the catalog size. This complexity can be detrimental to learning recommender systems operating on billions of items. The last two chapters address this issue by proposing optimization routines with sublinear complexities; a first solution is based on a new importance sampling variant of the reinforce algorithm, and a second one introduces a novel architecture and method for optimizing MIPS-based interactive systems. The proposed solutions accelerate optimization without losing on the recommendation quality.

Résumé. Les systèmes interactifs modernes façonnent notre expérience de l'internet. Des moteurs de recherche aux moteurs de recommandation, ces systèmes organisent de vastes quantités de contenu, permettant aux utilisateurs de trouver efficacement des réponses à leurs besoins. La qualité de l'expérience utilisateur au sein de ces systèmes peut varier de manière significative, et la capacité à fournir aux utilisateurs des options pertinentes au bon moment peut, non seulement améliorer leur satisfaction, mais aussi la rentabilité des entreprises qui exploitent ces systèmes. Ces dernières années, des efforts concertés ont été déployés pour exploiter les techniques d'apprentissage automatique afin d'améliorer les systèmes interactifs en combinant différents signaux. Cette thèse se concentre sur l'exploitation d'un type spécifique de signaux : *les données d'interaction.* Ces données ont une valeur unique car ils enregistrent directement les succès et les échecs des interactions précédentes. Néanmoins, la nature interactive de ces données rend leur analyse plus difficile par rapport aux problèmes classiques d'apprentissage. Le bandit contextuel hors-ligne formalise une version idéalisée de ce problème d'apprentissage. Il réduit les données d'interaction à des triplets; un contexte observé, une action effectuée par le système et une récompense reçue. Ces triplets sont essentiels à l'analyse du problème et à l'apprentissage de systèmes interactifs améliorés. Malgré les progrès récents, il reste des défis importants à relever pour apprendre des systèmes de décision avec des certificats de performance et pour adapter les approches actuelles aux problèmes de grande échelle.

Notre première préoccupation est de pouvoir mesurer les performances de notre système avant qu'il intéragisse avec l'environnement. La théorie de l'apprentissage statistique se concentre sur l'étude de la capacité de généralisation des algorithmes et se présente comme le candidat idéal pour répondre à cette question. Historiquement, ses outils ont été utilisés pour améliorer notre compréhension du paradigme de l'apprentissage supervisé, donnant naissance aux principes de minimisation du risque empirique et structurel. Plus récemment, la théorie de l'apprentissage statistique a été adaptée à l'apprentissage à partir de données d'interactions, ce qui a donné nais-

sance au principe de minimisation du risque contrefactuel. Ce nouvel objectif tient compte des difficultés liées à l'apprentissage à partir de données de bandits contextuels, mais son application est limitée à des scénarios simples. En particulier, l'objectif d'apprentissage n'est pas convexe, il ne peut pas être accéléré avec des méthodes de gradient stochastique, il introduit de nouveaux hyperparamètres qui sont difficiles à régler et ne parvient pas à fournir des certificats de performance sur les systèmes d'interaction nouvellement formés. La première partie de la thèse se concentre sur le développement de nouvelles idées d'apprentissage statistique pour relever ces défis. Nous recadrons la minimisation du risque contrefactuel **(CRM)** en utilisant l'optimisation distributionnellement robuste **(DRO)**. Ce changement de perspective nous permet d'améliorer la procédure d'optimisation, de calibrer automatiquement les hyperparamètres tout en bénéficiant des mêmes garanties. En outre, nous nous intéressons à l'apprentissage PAC-Bayésien, un cadre d'apprentissage statistique capable de mieux analyser la capacité de généralisation des algorithmes. En utilisant ce paradigme, nous construisons de nouvelles stratégies qui ne nécessitent aucun réglage des hyperparamètres, qui permettent une optimisation rapide et qui peuvent fournir des garanties solides sur la performance de nos systèmes interactifs.

Une autre préoccupation est d'apprendre efficacement les systèmes de décision fonctionnant sur des espaces d'action massifs. La deuxième partie de la thèse aborde ce défi, en se concentrant principalement sur la recommandation à grande échelle. L'apprentissage efficace dans ce cas peut être réalisé en exploitant différents signaux et en accélérant la procédure d'optimisation. Les méthodes existantes s'appuient uniquement sur le signal de bandit : les données d'intéraction du système avec les utilisateurs. Cependant, les signaux autres que le signal bandit, tels que le comportement organique, peuvent s'avérer extrêmement précieux. Sur la base de cette observation, nous consacrons un chapitre au développement d'une approche bayésienne de la recommandation qui combine les deux signaux. Nous fournissons les outils d'optimisation appropriés pour étendre l'apprentissage à de grands ensembles de données et prouvons empiriquement que les systèmes résultants bénéficient d'une meilleure qualité de recommandation.

Les systèmes de recommandation à grande échelle sont fréquemment mis à jour pour s'adapter aux intérêts en constante évolution des utilisateurs. La capacité à effectuer ces mises à jour régulièrement dépend de l'efficacité de la procédure d'optimisation. Lorsqu'ils sont confrontés à des espaces d'action extrêmement vastes, ces systèmes sont contraints à la structure **(MIPS)**: recherche du produit scalaire maximal pour répondre rapidement aux requêtes. Malgré leur prévalence dans l'industrie, l'optimisation de ces systèmes avec des objectifs d'apprentissage communs tend à être lente. En effet, le calcul de chaque gradient a une complexité au moins linéaire par rapport à la taille du catalogue. Cette complexité peut être préjudiciable à l'apprentissage de systèmes de recommandation fonctionnant sur des milliards d'éléments. Les deux derniers chapitres abordent ce problème en proposant des procédures d'optimisation avec des complexités sous-linéaires ; une première solution est basée sur une nouvelle variante d'échantillonnage préférentiel, et une seconde introduit une nouvelle architecture et une méthode pour optimiser les systèmes interactifs de la structure **(MIPS)**. Les solutions proposées accélèrent l'optimisation sans nuire à la qualité de la recommandation.

Contents

Introduction en français

1 Présentation générale

Ce manuscrit présente des contributions récentes, allant de la théorie aux applications à grande échelle, à un formalisme hors ligne du problème de la prise de décision séquentielle. Il s'agit d'un problème important avec de nombreuses applications dans le monde réel où un décideur, chargé d'optimiser un objectif spécifique, intéragit avec un environnement inconnu, enregistre ces intéractions et les exploite afin de mieux résoudre la tâche. Dans ce contexte, nous souhaitons répondre à la question suivante :

Comment tirer parti des interactions antérieures du décideur pour améliorer ses performances?

La réponse à cette question peut avoir un impact important sur les problèmes pratiques du monde réel. Par exemple, elle peut aider une campagne de marketing en ligne à obtenir plus de dons pour une campagne caritative, elle peut rendre plus précise la prescription des médicaments, ou elle peut simplement améliorer la qualité de la recommandation de votre plateforme de streaming préférée. Dans cette introduction, nous présentons le problème de l'apprentissage des décideurs à l'aide de l'exemple de la recommandation, qui sera au centre d'une grande partie de cette thèse. Les systèmes de recommandation se présentent comme le plus grand pilier de l'expérience Internet moderne. Dans chaque interaction, ces systèmes naviguent silencieusement une quantité écrasante d'informations et la traitent pour répondre aux besoins spécifiques de l'utilisateur. Une seule interaction d'un moteur de recommandation peut être résumée comme suit : le système rencontre un utilisateur, il choisit un article (ou plusieurs articles) à recommander dans un catalogue potentiellement vaste, délivre la recommandation et observe un retour de l'utilisateur.

Le retour obtenu est précieux car il représente les succès et les échecs des interactions passées. Ces interactions sont enregistrées et sont ensuite utilisées pour améliorer la qualité des recommandations du système. La nature interactive de l'ensemble des données collectées fait que les paradigmes d'apprentissage courants, tels que l'apprentissage supervisé, ne sont pas adaptés à l'étude de ce problème. Récemment, on s'est intéressé à l'adaptation des formalismes de prise de décision séquentielle pour améliorer la recommandation à partir des interactions enregistrées. L'apprentissage par renforcement (RL) (Sutton and Barto, 2018) et les bandits contextuels (CB) (Lattimore and Szepesvári, 2020) commencent à s'imposer comme de bons candidats pour modéliser ce problème d'apprentissage. Le cadre RL repose sur l'idée que les actions effectuées peuvent avoir un impact sur l'environnement. Ce paradigme peut modéliser des problèmes de décision séquentielle complexes et permet la planification. Ses outils peuvent optimiser les systèmes de recommandation pour des objectifs long terme ; par exemple, augmenter l'engagement et la rétention des utilisateurs (Afsar et al., 2022). L'adoption de ce formalisme a toutefois

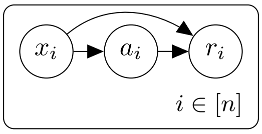


Figure 1: L'ensemble des données enregistrées $\mathcal{D}_n$ représentant n interactions du système de recommandation. Tous les triplets (contexte, action, récompense) sont indépendants.

un coût. La prise en compte des effets à long terme de la recommandation sur les utilisateurs rend l'analyse de cette approche plus difficile, ce qui nous incite à envisager un formalisme plus simple. Le bandit contextuel offre un compromis utile entre l'analyse formelle et l'impact pratique. Son hypothèse sous-jacente est que les actions effectuées par le système n'influencent pas les résultats futurs. Si cette formulation est moins convaincante lorsqu'il s'agit de récompenses différées (Afsar et al., 2022), son utilisation est raisonnable si nous voulons nous concentrer sur l'apprentissage de systèmes de recommandation qui optimisent des objectifs à court terme, limitées à l'action, telles que le taux de clics (Sakhi et al., 2020a) ou la durée de visionnage (Chen et al., 2019a). Dans cette thèse, nous adoptons la boîte à outils des bandits contextuels hors ligne (Bottou et al., 2013; Nguyen-Tang et al., 2022) pour formaliser l'apprentissage à partir des données d'interaction. Nous donnons de nouvelles approches fondées théoriquement pour apprendre des politiques avec de fortes garanties de performance et proposons de nouveaux algorithmes pour élargir l'impact de ce cadre à des applications à grande échelle du monde réel.

L'interaction d'un utilisateur avec un article recommandé peut être réduite à l'exemple suivant. Un utilisateur navigue sur un site web, le système de recommandation choisit un article dans un catalogue et le montre à l'utilisateur, l'utilisateur interagit avec l'article (clique ou non) et le résultat de cette interaction est encodé dans un retour d'information (présence/absence de clic) que le système enregistre. Dans le cadre du bandit contextuel, un utilisateur est représenté par un contexte x, généralement un vecteur réel vivant dans un espace à d dimensions $\mathcal{X} \subseteq \mathbb{R}^d$. Ces contextes, et donc les utilisateurs, sont échantillonnés *indépendamment* à partir de la même distribution inconnue $\nu(\mathcal{X})$. Après avoir vu un utilisateur, le moteur de recommandation lui fournit un article a issu d'un catalogue $\mathcal{A}$ de taille $|\mathcal{A}|\ dans \mathbb{N}$. Le système de recommandation est modélisé comme une politique $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$, qui est une fonction qui prend un contexte x et produit une distribution $\pi(\cdot|x)$ sur l'espace des actions possibles $\mathcal{A}$. Recommander un article a pour le contexte x revient à échantillonner l'article à partir de la distribution produite $a \sim \pi(\cdot|x)$. Après avoir livré l'article a à l'utilisateur du contexte x, notre système reçoit un retour de l'utilisateur; une récompense stochastique $r \in \mathbb{R}^+$ provenant d'une distribution inconnue $p(\cdot|x,a)$. Cette récompense encode la performance de l'élément recommandé par rapport à la mesure souhaitée ; plus la récompense est élevée, plus la performance l'est aussi. Notre objectif est de trouver des politiques très performantes, en minimisant le risque, défini comme la récompense négative attendue en tirant des actions de notre politique. Le risque d'une politique donnée π peut être exprimé comme suit :

$$R(\pi) = -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\mathbb{E}_{r\sim p(\cdot|x,a)}[r]\right].$$

Ce risque est définie comme une espérance sous la distribution générée par la politique évaluée. Comme nous n'avons pas accès aux interactions de la nouvelle politique π avec l'environnement, un moyen simple d'estimer cette quantité est de laisser π interagir avec les utilisateurs en ligne. Dans la plupart des scénarios, cela n'est pas possible, car nous n'avons pas le luxe de déployer de mauvaises politiques. Dans les applications réelles, nous disposons déjà de la version actuelle de notre système de recommandation, représentée par la politique π_0, qui interagit avec l'environnement et enregistre ces intéractions. Notre objectif principal est d'évaluer dans quelle mesure une nouvelle itération du système améliorera la version actuellement déployée. Un moyen courant d'y parvenir est de réaliser des A/B-tests en ligne (Kohavi et al., 2012). Cette approche est considérée comme l'"étalon-or" pour estimer l'effet du remplacement de la politique actuelle π_0 par une politique potentiellement meilleure (Gupta et al., 2019). Les A/B-tests nécessitent toutefois un effort d'ingénierie important et un monitoring constant s'étalant sur plusieurs jours pour être correctement analysés. Idéalement, nous avons besoin d'outils d'évaluation et d'apprentissage hors ligne qui puissent nous trouver des politiques prometteuses afin de réduire le nombre d'A/B-tests inutiles. Lorsque les hypothèses du bandit contextuel sont satisfaites, nous pouvons utiliser la boîte à outils du cadre pour y parvenir. L'idée est d'exploiter les interactions existantes de π_0 pour trouver des politiques plus performantes. L'ensemble de données d'interaction est appelé dans la littérature "logged bandit feedback dataset" (Swaminathan and Joachims, 2015a) :

$$\mathcal{D}_n = \{x_i \sim \nu, a_i \sim \pi_0(\cdot|x_i), r_i \sim p(\cdot|x_i, a_i), \pi_0(a_i|x_i)\}_{i\in[n]}.$$

La figure 1 présente une représentation graphique des données. La principale difficulté rencontrée lors de l'apprentissage à partir de ces données est le biais potentiel créé par la procédure de collecte ; nous n'avons accès qu'aux résultats des actions échantillonnées à partir de π_0. Le cadre d'apprentissage hors ligne du bandit contextuel propose deux approches distinctes pour résoudre ce problème : l'approche de modélisation du coût et l'approche d'échantillonnage préférentiel.

l'approche de modélisation du coût ou *la méthode directe* exploite les données d'interaction $\mathcal{D}_n$ pour construire un modèle de la récompense (Sakhi et al., 2020a; Jeunen and Goethals, 2021). Une politique optimale est alors naturellement dérivée en jouant pour chaque contexte x, l'action avec la récompense la plus élevée selon le modèle. La méthode directe est simple à mettre en œuvre, car elle réduit l'apprentissage à un problème de régression (Brandfonbrener et al., 2021). Cette approche est théoriquement bien étudiée et bénéficie de solides garanties (Nguyen-Tang et al., 2022). Cependant, elle souffre d'un biais important et incontrôlé lorsque la récompense est complexe, ce qui rend son efficacité entièrement dépendante de notre capacité à modéliser la structure du problème. La méthode directe est efficace lorsque nous avons confiance en notre capacité à comprendre le problème. Lorsque le signal de récompense est complexe, nous pouvons préférer une autre approche qui ne dépend pas entièrement de notre effort de modélisation.

L'approche d'échantillonnage préférentiel (Horvitz and Thompson, 1952; Bottou et al., 2013; Dudík et al., 2014), souvent appelée *apprentissage hors politique*, ne nécessite pas de modèlisation. Elle apprend une nouvelle politique π directement à partir des interactions $\mathcal{D}_n$ en utilisant des estimateurs corrigés par échantillonnage préférentiel (Chopin and Papaspiliopoulos, 2020). Sous des hypothèses modérées (Horvitz and Thompson, 1952), cette méthode peut produire des estimateurs non biaisés, qui se présentent plus faciles à analyser et à optimiser (Ajalloeian and Stich, 2020). Ces estimateurs souffrent cependant d'une variance potentiellement importante dès que la politique apprise s'éloigne de la politique d'enregistrement π_0, ce qui les rend peu fiables pour l'apprentissage. Il est prouvé empiriquement que l'apprentissage avec ces estimateurs peut aboutir à des politiques peu performantes (Swaminathan and Joachims, 2015a,b), parfois même pires que π_0 (Chen et al., 2019b; London and Sandler, 2019). Cette observation motive l'utilisation d'outils de la théorie de l'apprentissage (Zhou, 2002; McAllester, 1998) pour proposer des objectifs avec un meilleur comportement, sans connaissance de la fonction de récompenses. L'objectif de cet effort de recherche est de produire de nouvelles politiques qui sont **théoriquement meilleures** que la politique d'enregistrement sans interactions additionelles avec l'environnement. Cela est utile dans les environnements de production où nous aimerions proposer un nouveau système qui améliorera le système de production actuel avec certitude.

Le premier effort dans ce sens a été mené par Swaminathan and Joachims (2015a) et a abouti au principe **CRM : Counterfactual Risk Minimisation** ou Minimisation du risque contrefactuel. Le principe **CRM** s'appuie sur les outils de la théorie de l'apprentissage statistique (Vapnik, 1998), un cadre qui permet d'étudier la capacité de généralisation des algorithmes d'apprentissage. Motivé par la construction d'une borne empirique de type Bernstein (Maurer and Pontil, 2009) sur le risque réel des politiques, et utilisant des arguments de nombre de couverture (Zhou, 2002), ce principe préconise de pénaliser les estimateurs de poids d'importance avec la *racine carrée de la variance empirique du risque*. Cette pénalité est contrôlée par un hyperparamètre λ, défini à l'aide d'une validation croisée sur une partie de validation. L'intuition sous-jacente est que pour améliorer la politique π_0, nous devrions rechercher des politiques qui ont un *petit risque empirique* tout en restant proches de π_0. Ce principe permet d'obtenir des politiques plus performantes que l'optimisation directe d'estimateurs d'échantillonnage préférentiel (Swaminathan and Joachims, 2015a,b). Toutefois, son paradigme d'apprentissage souffre de différentes limitations, ce qui réduit son application à des scénarios simples. En particulier, l'ajout de la pénalisation rend l'objectif d'apprentissage non convexe et non décomposable, ce qui interdit l'utilisation de méthodes de gradient stochastique. Cette pénalité est également contrôlée par un nouvel hyperparamètre λ qui est difficile à régler et qui ajoute à la complexité de l'approche. Enfin, le principe **CRM** ne fournit pas de certificats de performance sur la politique nouvellement formée. Ces limites seront examinées en détail plus loin dans l'introduction. Plus récemment, un nouveau principe a été introduit pour atténuer certaines de ces limitations. En analysant ce problème d'apprentissage sous l'angle PAC-Bayesien (McAllester, 1998; Alquier, 2021), London and Sandler (2019) développent une approche améliorée. Les auteurs fondent leur analyse sur la borne PAC-Bayesienne de McAllester (2003). Pour les politiques paramétriques, cela motive une régularisation L_2 du paramètre de la nouvelle politique vers le paramètre de la politique d'enregistrement π_0. La régularisation est également contrôlée par un hyperparamètre λ qui doit être réglé. Ce principe est basé sur la même intuition de rester proche de π_0, mais cette fois, il est effectué sur l'espace des paramètres. L'adoption d'une régularisation L_2 au lieu d'une pénalisation de la variance d'échantillon facilite le problème d'optimisation et permet l'utilisation de la descente de gradient stochastique. Cependant, le paramètre λ de la régularisation L_2 souffre des mêmes limitations et le principe ne peut pas produire de meilleures politiques. Les résultats empiriques démontrent que ces principes échouent parfois à améliorer la politique π_0 (Chen et al., 2019b). Ces limites seront développées dans la section suivante, avant que nous ne présentions les contributions de la première partie de la thèse. Le chapitre 3 recadre **CRM** en utilisant les outils de **Distributionnally Robust Optimisation** (Duchi et al., 2021), un cadre statistique conçu pour la prise de décision face à l'incertain. En outre, les chapitres 4 et 5 s'appuient sur les travaux de London and Sandler (2019) et poursuivent le développement des outils **PAC-Bayésien** (McAllester, 1998) pour le bandit contextuel hors ligne. L'analyse produit des principes qui sont plus faciles à optimiser, ne nécessitent pas d'hyperparamètres supplémentaires à régler et bénéficient, pour certains, de meilleures garanties de performance, ce qui nous rapproche de l'apprentissage de politiques améliorant π_0 hors ligne.

Dans le monde réel, les systèmes interactifs sont souvent confrontés à des scénarios à grande échelle, dans lesquels ils doivent apprendre à partir d'un nombre considérable d'interactions ($n \gg 1$) et opérer sur des catalogues massifs ($|\mathcal{A}| \gg 1$). Pour que ces systèmes puissent fournir des recommandations en quelques millisecondes, ils sont limités à une certaine structure (Shrivastava and Li, 2014; Aouali et al., 2022) afin de permettre une réponse rapide aux requêtes. Pendant longtemps, les systèmes de recommandation à grande échelle ont été formés à la prédiction des préférences (Harper and Konstan, 2015; Gomez-Uribe and Hunt, 2016) ou à la prédiction de l'élément suivant (Hidasi et al., 2015; Wu et al., 2019). Ces approches de modélisation sont généralement considérées comme de piètres substituts à la récompense que nous souhaitons optimiser (Jannach and Jugovac, 2019). L'adaptation de la boîte à outils de bandits contextuels hors ligne à l'apprentissage de systèmes de recommandation à grande échelle aura un impact considérable sur le secteur. Ces outils peuvent permettre d'aligner les recommandations sur des signaux de récompense complexes, améliorant ainsi la satisfaction des utilisateurs et la rentabilité des entreprises qui développent ces systèmes. Comme nous l'avons vu précédemment, nous pouvons soit adopter la **méthode directe** si nous savons comment modéliser la récompense, soit utiliser des **principes d'apprentissage avec des estimateurs d'échantillonnage préférentiel** pour apprendre une politique directement. Ces deux méthodes permettent d'apprendre de manière fiable un système de recommandation performant. Malheureusement, ces méthodes, dans leur forme simple, présentent des inconvénients lorsqu'elles traitent des problèmes à grande échelle. La deuxième partie de la thèse aborde ces limitations et permet un apprentissage efficace et rapide des systèmes de recommandation à grande échelle.

La méthode directe repose entièrement sur notre capacité à apprendre un modèle qui reflète les propriétés de la récompense. La compréhension parfaite du problème réduit le biais lié à la modélisation, mais il existe un autre problème, lié à l'apprentissage à partir de $\mathcal{D}_n$, qui devient plus prononcé dans les scénarios à grand catalogue. En effet, l'apprentissage naïf du modèle de récompense à partir de $\mathcal{D}_n$ souffre du déséquilibre présent dans les données collectées. Le modèle de récompense sera bien estimé pour les actions qui sont susceptibles d'être échantillonnées sous π_0, et mal estimé pour le reste. Cette différence dans la qualité de l'estimation peut rendre les décisions prises par la politique dérivée peu fiables (Smith and Winkler, 2006). Ce phénomène est accentué lorsqu'on a affaire à des catalogues de grande taille, car π_0 ne peut jamais collecter suffisamment d'échantillons pour couvrir l'ensemble de l'espace d'action. Nous consacrons le chapitre 6 à l'examen d'une solution bayésienne à ce problème. Nous introduisons une structure au modèle et utilisons une autre source de données pour apprendre efficacement le modèle de récompense. Plus de détails sur cette approche peuvent être trouvés dans la section contribution.

Les objectifs d'échantillonnage préférentiel deviennent intéressants lorsque le signal de récompense est complexe. Toutefois, dans les scénarios à grande échelle, ces objectifs d'apprentissage souffrent de deux problèmes majeurs. Le premier problème est lié à la variance de ces estimateurs, qui augmente avec la taille de l'espace d'action. En effet, la variance des estimateurs courants (Horvitz and Thompson, 1952; Ionides, 2008; Dudík et al., 2014) devient incontrôlable lorsque les politiques opèrent sur des catalogues massifs (Saito and Joachims, 2022b). Comme cette variance peut être très importante, l'ajout d'une pénalisation de la variance, par exemple, obligera la politique nouvellement apprise π à imiter le comportement de π_0. Ce phénomène rend nos principes d'apprentissage trop conservateurs, en renvoyant des politiques très proches de π_0. Cette observation a motivé la construction d'une nouvelle famille d'estimateurs (Saito and Joachims, 2022a; Saito et al., 2023) pour atténuer ce problème de variance. Ces contributions récentes traitent des limites statistiques des objectifs d'échantillonnage préférentiel dans les scénarios à grand catalogue, mais les problèmes de temps de calcul liés à l'optimisation de ces objectifs restent non résolus. Les systèmes à grande échelle sont fréquemment mis à jour, et des routines d'optimisation rapides sont hautement souhaitables dans ce contexte. Les méthodes existantes proposent des itérations de gradient dont l'échelle est au moins linéaire sur la taille du catalogue. Cette complexité peut être préjudiciable à l'apprentissage des systèmes de recommandation fonctionnant sur des milliards d'éléments. Les deux derniers chapitres (chapitres 7 et 8) se concentrent sur l'aspect computationnel et proposent des routines d'optimisation avec des complexités sous-linéaires. Ces solutions seront développées plus en détail dans la section contribution.

Dans cette thèse, nous couvrons différentes disciplines connectées, tout en équilibrant les outils théoriques et les algorithmes pratiques. Pour faciliter la présentation, nous souhaitons donner aux lecteurs un aperçu de l'avancement de chaque domaine de recherche. À cette fin, nous consacrons un chapitre à l'examen de la littérature existante, que nous jugeons utile pour tout chercheur.

Chapter 2. Literature Review. Ce chapitre présente une revue de la littérature couvrant ainsi les différents outils utilisés tout au long de cette thèse. Nous donnons un bref aperçu de la littérature sur le Bandit Contextuel, un formalisme pratique pour étudier la recommandation basée sur la récompense, en présentant à la fois ses formulations en ligne et hors ligne. En nous concentrant sur le cadre hors ligne, nous consacrons une section à la présentation des outils d'apprentissage statistique, nécessaires à l'étude des systèmes de décision d'apprentissage avec des garanties de performance. Nous présentons ensuite le développement des systèmes de recommandation et la manière dont la modélisation de la recommandation est passée de la prédiction des préférences à la maximisation de la récompense, et nous concluons par les considérations algorithmiques qui se posent dans le contexte de la prise de décision à grande échelle.

Part I - Offline Learning with Performance Guarantees. La première partie de la thèse se concentre sur les limites des principes d'apprentissage actuels. Ces principes ont été proposés pour améliorer la politique d'enregistrement π_0, en atténuant les problèmes liés à l'échantillonnage préférentiel. Sans perte de généralité, nous présentons le problème à l'aide du `IPS` : *Inverse Propensity Scoring* (Horvitz and Thompson, 1952), sans doute l'estimateur le plus simple et le plus étudié. Pour une politique π, nous rappelons son expression :

$$\hat{R}_n^{\texttt{IPS}}(\pi) = -\frac{1}{n}\sum_{i=1}^{n}\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}r_i.$$

Lorsqu'elle est évaluée sur π_0, `IPS` donne la moyenne empirique des coûts collectés en tant qu'estimation du risque, ce qui est considéré comme un estimateur sans biais de $R(\pi_0)$. Toutefois, une simple analyse de la variance de cet estimateur montre que le fait de s'éloigner de π_0 entraîne une baisse de la qualité de l'estimation. Si la récompense observée est bornée (par exemple, $r \in [0,1]$), nous avons :

$$\begin{aligned}\mathbb{V}\left[\hat{R}_n^{\texttt{IPS}}(\pi)\right] &= \frac{1}{n}\left(\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x), r\sim p(\cdot|x,a)}\left[\left(\frac{\pi(a|x)}{\pi_0(a|x)}\right)^2 r^2\right] - \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right]^2\right)\\ &\leq \frac{1}{n}\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x), r\sim p(\cdot|x,a)}\left[\left(\frac{\pi(a|x)}{\pi_0(a|x)}\right)^2\right] = \frac{1}{n}\left(\chi^2(\pi,\pi_0)+1\right),\end{aligned}$$

avec $\chi^2(\pi,\pi_0)$ la divergence χ-deux entre π et π_0. La variance a à peu près le même comportement que la divergence χ-deux, augmentant lorsque π s'éloigne de π_0. En particulier, la variance augmente avec les poids d'importance. Les poids d'importance sont très élevés lorsque la nouvelle politique π attribue une forte probabilité à des actions qui étaient très peu susceptibles d'être jouées sous π_0. Cela signifie que la qualité de l'estimation dépend fortement de la politique évaluée π, ce qui rend `IPS`[1] indigne de confiance pour les politiques éloignées

[1] Tous les estimateurs basés sur les poids d'importance souffrent de la même limitation.

du voisinage de π_0. Cette observation est confirmée dans la pratique, en particulier lorsque l'on utilise ces estimateurs comme objectif d'apprentissage. Par exemple, la minimisation de l'estimateur IPS par rapport à une classe de politiques peut conduire à des politiques ayant de mauvaises performances en ligne. Lorsque la politique apprise π est éloignée de π_0, l'estimation IPS du risque de π ne reflète pas son risque réel, car π se trouve dans une partie de l'espace qui induit un estimateur avec une grande variance. Pour contourner ces limitations, il faudrait restreindre l'optimisation aux politiques autour de la politique π_0. Le **CRM : Minimisation du risque contrefactuel** (Swaminathan and Joachims, 2015a) formalise cette idée en utilisant des arguments d'apprentissage statistique. Motivé par la construction d'une borne empirique de type Bernstein (Maurer and Pontil, 2009), le principe préconise la minimisation de l'estimateur IPS pénalisé par *sa variance*. Ce principe est ensuite utilisé pour produire un algorithme d'apprentissage de politiques "softmax" (Mei et al., 2020b) de la forme :

$$\begin{aligned}\forall(x,a) \quad \pi_\theta(a|x) &= \texttt{softmax}_{\mathcal{A}}\left(f_\theta(x,a)\right)\\ &= \frac{\exp(f_\theta(x,a)}{\sum_{a'\in\mathcal{A}}\exp(f_\theta(x,a'))}.\end{aligned} \tag{1}$$

avec θ un paramètre provenant d'un espace paramétrique $\theta \in \Theta$ et $f_\theta : \mathcal{X} \times \mathcal{A}$ une fonction qui encode la pertinence de l'action a par rapport au contexte x. L'algorithme proposé est appelé **POEM** : Policy Optimizer for Exponential Models (Swaminathan and Joachims, 2015a) et résout l'objectif suivant pour les politiques softmax :

$$\arg\min_{\theta\in\Theta}\left\{\hat{R}_n^{\texttt{IPS}}(\pi_\theta) + \lambda\sqrt{\hat{V}^{\texttt{IPS}}(\pi_\theta)}\right\},$$

avec λ un hyperparamètre généralement défini à l'aide de données de validation, et $\hat{V}^{\texttt{IPS}}(\pi_\theta)$ la variance empirique induite par l'évaluation de π avec IPS :

$$\hat{V}^{\texttt{IPS}}(\pi_\theta) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}r_i + \hat{R}_n^{\texttt{IPS}}(\pi_\theta)\right)^2.$$

Swaminathan and Joachims (2015a) a démontré empiriquement la supériorité de ce principe ; les politiques renvoyées par **POEM** présentent un risque beaucoup plus faible que celles obtenues en minimisant naïvement l'objectif IPS. L'ajout de la régularisation rend l'approche plus fondée, mais souffre encore de limitations qui réduisent son applicabilité dans les scénarios de la vie réelle:

(1) Mise à l'échelle. La plus grande limitation du principe CRM est sa capacité à s'adapter à de grands ensembles de données $\mathcal{D}_n$. La présence du terme de variance fait que le calcul du gradient de l'objectif CRM se fait en $\mathcal{O}(n)$, en termes de coûts de calcul et de mémoire, car il nécessite de parcourir l'entièreté des données. Dans un scénario de système de recommandation, des millions d'interactions sont enregistrées chaque jour. Ces applications traitent un très grand nombre d'échantillons n et ne peuvent se permettre ce coût de calcul. Ce problème est généralement résolu en recourant à un algorithme d'optimisation stochastique, qui ne nécessite qu'un accès aux gradients stochastiques non biaisés de l'objectif. Ceux-ci sont particulièrement faciles à obtenir lorsque l'objectif se décompose en une somme sur les entrées de l'ensemble de données, car il suffit de calculer la somme sur des lots pour obtenir des gradients non biaisés. Malheureusement, l'objectif CRM n'est pas adapté à l'optimisation stochastique, car le terme de pénalisation ne s'écrit pas comme une somme. Swaminathan and Joachims (2015a) a proposé une relaxation de l'objectif CRM, basée sur une stratégie de minimisation/majoration, qui peut bénéficier *partiellement* des gradients stochastiques. Leur approche nécessite toujours de passer par l'ensemble des données enregistrées de temps à autre, ce qui permet d'obtenir une procédure d'une complexité informatique identique.

London and Sandler (2019) propose un principe amélioré qui traite de la première limitation du CRM. Au lieu de s'appuyer sur la borne empirique de type Bernstein (Maurer and Pontil, 2009), London and Sandler (2019) adapte la borne PAC-Bayesienne de McAllester (2003) pour dériver des objectifs d'apprentissage pour ce problème. Le principe obtenu motive l'utilisation d'une régularisation L_2 vers le paramètre θ_0 de la politique π_0. Cette régularisation est contrôlée par un hyperparamètre λ, ce qui donne le problème d'optimisation suivant pour les politiques softmax paramétrées :

$$\arg\min_{\theta \in \Theta} \left\{ \hat{R}_n^{\texttt{IPS}}(\pi_\theta) + \lambda \|\theta - \theta_0\|^2 \right\},$$

L'objectif d'optimisation se prête à l'optimisation stochastique (décomposable en une somme), s'adapte à de grands ensembles de données et produit des politiques avec de meilleures performances empiriques. Cependant, ce principe, comme le CRM, souffre d'autres limitations, présentées ci-après :

(2) Pas de garanties de performance. Les deux principes dérivés sont motivés par la construction de bornes couvrant le risque réel des politiques. Ces bornes, dans leur forme brute, ne peuvent pas être utilisées directement comme objectif d'apprentissage. En effet, la borne dérivée dans Swaminathan and Joachims (2015a) contient des quantités théoriques et celle dérivée dans London and Sandler (2019) donne une couverture triviale. L'introduction de l'hyperparamètre λ permet d'obtenir des objectifs pratiques, qui perdent les garanties théoriques données par les bornes initiales. Ces objectifs ne couvrent pas nécessairement le risque réel, et leur optimisation peut conduire à des politiques pires que π_0. Des preuves empiriques peuvent être trouvées dans (Chen et al., 2019b) où le principe CRM ne parvient pas à améliorer π_0.

(3) Rajout d'hyperparamètre. Un autre problème majeur de ces principes est également causé par l'introduction de λ et sa sélection. Le paramètre libre λ nécessite un réglage minutieux, car son choix a un impact considérable sur les performances de la politique obtenue. Comme il n'existe pas de lignes directrices théoriques pour définir une bonne valeur de λ, la stratégie consiste à procéder à une validation croisée du paramètre sur une grille relativement fine en utilisant l'estimateur `IPS`. La validation croisée ajoute à la complexité de l'algorithme. Cette procédure nécessite également de disposer d'un ensemble de validation qui ne sera pas utilisé pour l'apprentissage, ce qui accentue le problème de la variance. En outre, comme l'estimateur `IPS` est toujours utilisé pour sélectionner la meilleure valeur de λ, la politique renvoyée à la fin est celle qui minimise le risque `IPS` sur l'ensemble de validation, ce qui rend l'ensemble du principe incohérent.

L'objectif de cette première partie est de fournir aux praticiens de meilleurs principes qui contournent complètement ces limitations, en bénéficiant de meilleures garanties statistiques et de performances empiriques.

Chapter 3. **Offline Learning with Distributionally Robust Optimization.** Dans ce chapitre, nous présentons une formulation alternative au principe CRM en recourant au cadre de l'optimisation distributionnellement robuste (DRO) (Duchi et al., 2021). Ces outils permettent de construire élégamment des intervalles de confiance sensibles à la variance sur le vrai risque en utilisant des ensembles d'ambiguïté basés sur la f-divergence. Nous appliquons ce principe au problème de l'évaluation et de l'optimisation des politiques hors ligne. L'objectif résultant traite des limites **(1)** et **(3)** ; il bénéficie des mêmes garanties statistiques que le CRM, peut être calibré automatiquement en utilisant des arguments de couverture asymptotique et se prête à l'optimisation stochastique. Nous présentons des expériences numériques solides montrant que l'approche proposée traite efficacement les lacunes de la CRM. Ce chapitre est adapté de la publication suivante :

- Otmane Sakhi, Louis Faury, and Flavian Vasile (2020b). Improving Offline Contextual Bandits with Distributional Robustness. *Proceedings of the ACM RecSys Workshop on Reinforcement Learning and Robust Estimators for Recommendation Systems, 2020.*

Chapter 4. Offline Learning with PAC-Bayesian Theory. Dans ce chapitre, nous remettons complètement en question le paradigme de l'apprentissage hors politique et préconisons une stratégie théoriquement fondée pour améliorer avec certitude la politique déployée π_0. La méthode proposée consiste à créer des bornes inférieures de la quantité d'améliorations uivante $\mathcal{I}(\pi) = R(\pi_0) - R(\pi)$, et à déployer de nouvelles politiques uniquement lorsque nous sommes sûrs que $\mathcal{I}(\pi) > 0$. Nous basons notre approche sur la théorie de l'apprentissage PAC-Bayesien (Alquier, 2021) et démontrons que ses outils conviennent parfaitement au problème de l'apprentissage hors ligne. En particulier, en interprétant les politiques comme des mélanges de règles de décision, nous dérivons une borne PAC-Bayesienne étroite, de type Bernstein, qui rend notre stratégie viable. La stratégie résultante traite les trois limitations ; nous montrons que l'algorithme résultant peut donner des certificats d'amélioration, se prête à l'optimisation stochastique et ne nécessite aucun réglage d'hyperparamètre, ce qui constitue un grand pas en avant vers la réalisation d'un apprentissage hors politique pratique avec de véritables garanties de performance. Ce chapitre est basé sur la publication suivante :

- Otmane Sakhi, Pierre Alquier, and Nicolas Chopin (2023a). PAC-Bayesian Offline Contextual Bandits with Guarantees. *Proceedings of the 40th International Conference on Machine Learning, 2023.*

Chapter 5. A Better PAC-Bayesian Analysis of Offline Learning. Dans ce chapitre, nous poursuivons le développement de l'analyse PAC-Bayesienne du problème de l'apprentissage hors ligne des politiques. En exploitant la nature négative du risque, nous dérivons de nouvelles bornes plus étroites qui s'appliquent à une classe plus large d'estimateurs de risque. L'idée est basée sur un traitement raffiné de la fonction génératrice de moments du risque et étend les limites empiriques de Bernstein à des ordres supérieurs. La particularité de ces résultats est qu'ils sont entièrement empiriques ; nous ne supposons pas l'accès à π_0 contrairement aux bornes dérivées précédemment. Nous observons que nos résultats peuvent donner de meilleures garanties et nous permettent d'obtenir de nouvelles informations sur les estimateurs utilisés. Ce chapitre se concentre sur la fourniture de résultats techniques et est basé sur un travail non publié.

Part II - Offline Learning of Large Scale Recommendation. L'apprentissage hors ligne offre des solutions pratiques pour aligner efficacement les systèmes de décision sur des signaux de récompense complexes. Si la communauté des chercheurs s'est concentrée sur l'amélioration des estimateurs et des paradigmes d'apprentissage existants, peu d'attention a été accordée à l'adaptation de ces approches au contexte des grands espaces d'action. Ceci est intéressant pour les moteurs de recherche d'apprentissage, les systèmes de recommandation et pratiquement toutes les applications où le nombre d'interactions n et la taille de l'espace d'action $|\mathcal{A}|$ sont massifs. Le principal défi dans ces applications est de concevoir des règles de décision qui satisfont aux contraintes d'ingénierie, tout en fournissant des algorithmes pratiques qui permettent leur alignement avec les signaux de récompense d'une manière rapide et fiable. Les moteurs de recherche doivent répondre aux requêtes en quelques millisecondes, et les systèmes de recommandation du monde réel (pensez à une plateforme de streaming vidéo) doivent remplir de manière rapide la page d'accueil avec du contenu. Ces contraintes de vitesse doivent être respectées même si le catalogue (espace d'action) contient des milliards d'éléments. Un autre aspect à prendre en considération est que ces systèmes sont fréquemment mis à jour, ce qui impose une contrainte considérable sur le temps d'apprentissage de ces systèmes. En effet, si nous devons mettre à jour notre système de décision *quotidiennement* sur la base de ses interactions, le temps d'apprentissage devrait être nettement inférieur à un*jour* car il faut collecter suffisamment d'interactions et mettre à jour le système dans le même laps de temps. Dans leurs implémentations naïves, la prise de décision et l'apprentissage de ces systèmes sont linéairement proportionnels à la taille de l'espace d'action $\mathcal{O}(|\mathcal{A}|)$, ce qui n'est pas possible dans les scénarios d'espace d'action massif. Dans ce qui suit, nous développons la discussion autour de ces deux aspects importants et présentons nos contributions dans ce domaine.

Prise de décision rapide. La politique déployée permet de répondre à une requête ou de fournir des recommandations précises. Quelle que soit la nature de la politique et de sa mise en œuvre, cette étape se résume généralement à l'identification *rapide* d'un sous-échantillon de taille $K \geq 1$ de bonnes actions (Chen et al., 2019a) à partir de l'espace d'action potentiellement massif. En règle générale, et pour un utilisateur x, la qualité des actions est encodée dans la fonction de score $f_\theta(\cdot, x) : \mathcal{A} \rightarrow \mathbb{R}$ tandis que les bonnes actions sont identifiées en trouvant les actions ayant le meilleur score, en résolvant le problème suivant :

$$[a_1, ..., a_K] = \underset{a' \in \mathcal{A}}{\overset{K}{\arg\operatorname{sort}}} \left\{ f_\theta(a', x) \right\}, \tag{2}$$

avec l'opérateur $\arg\operatorname{sort}^K_{a' \in \mathcal{A}}$ qui renvoie les K actions les mieux notées. Cette opération de tri a une complexité linéaire sur la taille de l'espace des actions $\mathcal{O}(|\mathcal{A}| \log K)$ et ne peut pas être adoptée dans un environnement de production à grande échelle. La solution courante pour réduire cette complexité consiste à imposer une structure à la fonction de score. En limitant l'espace de la fonction de score à ce qui suit :

$$\forall (x, a) \quad f_\theta(a, x) = h_\Xi(x)^\intercal \beta_a$$

avec $\theta = [\Xi, \beta]$, la fonction de score devient un produit scalaire entre une transformation du contexte $h_\Xi(x)$ et une transformation de l'action β_a, tous deux résidant dans un espace latent $\mathbb{R}^l$ de dimension $l \ll |\mathcal{A}|$. Avec cette structure, l'équation (2) peut être résolue en approximant MIPS : Maximum Inner Product Search (Shrivastava and Li, 2014) dans une complexité temporelle de $\mathcal{O}(\log |\mathcal{A}|)$ au lieu de $\mathcal{O}(|\mathcal{A}|)$, ce qui rend possible une prise de décision rapide sans considérations supplémentaires.

Apprentissage efficace pour la méthode directe. La méthode directe dérive une politique optimale, qui identifie pour chaque contexte les actions ayant le meilleur score selon le modèle de récompense $r_\mathcal{M}$. Cela signifie que si $r_\mathcal{M}$ est correctement paramétré, une prise de décision rapide est possible. L'apprentissage d'un bon modèle $r_\mathcal{M}$ nécessite une excellente compréhension du problème sous-jacent et est généralement obtenue par *maximum de vraisemblance* (Aouali et al., 2023b) ou des *heuristiques de classement* (Rendle et al., 2009), qui sont des méthodes dont l'apprentissage est indépendant de la taille de l'espace d'action. Ces algorithmes peuvent toutefois présenter d'autres lacunes si nous ne prenons pas garde aux particularités de ce cadre.

Dans les scénarios à grand espace d'action, il est impossible pour la politique déployée de collecter suffisamment d'interactions pour chaque action dans $\mathcal{A}$. Le signal de récompense est inégalement réparti, car la majorité des données collectées proviennent d'actions très probables sous π_0 et peu ou pas de données sont disponibles pour le reste des actions. Cela signifie que si nous utilisons le *principe du maximum de vraisemblance*, la qualité du modèle appris $r_{\mathcal{M}}$ dépendra de la paire contexte/action ; l'estimation est précise pour les paires action/contexte qui sont suffisamment présentes dans les données. Ce déséquilibre dans la qualité de l'estimation a un impact négatif sur la politique dérivée, car les décisions basées sur le *MLE* peuvent souffrir d'une déception post-décisionnelle (Smith and Winkler, 2006). L'un des moyens d'atténuer ce problème est de l'inscrire dans le cadre de la théorie de la décision *Bayesienne* (West et al., 2021). Par exemple, Jeunen and Goethals (2021) démontre que même une simple modélisation bayésienne de la récompense permet d'améliorer le comportement des politiques. Avec l'aide de distribution à priori bien choisis, cette formulation peut également intégrer des corrélations supplémentaires entre les contextes et les actions, ce qui rend l'apprentissage encore plus efficace (Aouali et al., 2023c). Cependant, le principal défi de la modélisation bayésienne est d'ordre computationnel ; l'approximation des distributions à posteriori sur des milliards d'interactions, à l'aide de modèles, est difficile et nécessite un soin particulier (Chopin and Papaspiliopoulos, 2020). Nous consacrons un chapitre à cette discussion et construisons un modèle de récompense bayésien pour la recommandation en utilisant des à priori bien construit, tout en fournissant des outils appropriés pour accélérer son apprentissage dans des applications à grande échelle.

Chapter 6. Scalable Bayesian Reward Modelling. Dans ce chapitre, nous empruntons la voie de la méthode directe et développons un modèle bayésien de la récompense dans le cas de la recommandation d'un seul article. Nous reconnaissons la présence de deux types de signaux dans les problèmes de recommandation : les signaux organiques et les signaux de bandits. Alors que nous conditionnons notre modèle au retour bandit, les interactions organiques entre les contextes et les actions nous aident à construire une distribution a priori qui incorpore trois similarités : la similarité contexte-action, la similarité action-action et la similarité contexte-contexte. Ces similarités nous permettent d'obtenir de bonnes estimations de la récompense dans toutes les régions de l'espace, même pour les actions et les contextes les moins explorés. Le modèle proposé est flexible, utilise efficacement les données existantes mais produit une distribution à posteriori intraitable. Nous fournissons des outils computationnels faciles à mettre en œuvre pour approximer sa solution en nous basant sur des approches variationnelles (Blei et al., 2017). L'algorithme résultant s'adapte à de grands ensembles de données, peut apprendre efficacement dans différents scénarios et bénéficie de la paramétrisation du produit scalaire, ce qui permet une prise de décision rapide. Ce chapitre est basé sur la publication suivante :

- Otmane Sakhi, Stephen Bonner, David Rohde and Flavian Vasile (2020a). BLOB: A Probabilistic Model for Recommendation that Combines Organic and Bandit Signals. *KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Apprentissage rapide avec les estimateurs d'échantillonnage préférentiel Dans notre quête d'algorithmes évolutifs d'apprentissage de politiques hors ligne, nous exprimons également notre intérêt pour les paradigmes d'apprentissage basés sur l'échantillonnage préférentiel. Cette approche apprend une politique directement, et même les opérations simples impliquent le calcul de sommes sur l'ensemble de l'espace d'action. En particulier, nous devons être très prudents lorsque nous calculons/approximons les gradients de nos objectifs, car cette opération se calcule linéairement dans $|\mathcal{A}|$, ce qui peut ralentir considérablement la routine d'optimisation. La question de la mise à l'échelle des objectifs généraux d'apprentissage hors politique a attiré peu d'attention ; Chen et al. (2019a) a appris une politique prête pour la production avec un objectif basé sur IPS sans se préoccuper de l'aspect computationnel. Nous nous intéressons à cette question et souhaitons fournir des méthodes d'accélération générales. Nous étudions la famille spécifique d'objectifs qui peuvent être écrits comme des espérances sous la politique évaluée. Cette famille comprend les estimateurs couramment adoptés (Horvitz and Thompson, 1952; Dudík et al., 2014; Wang et al., 2017; Saito and Joachims, 2022a; Saito et al., 2023; Aouali et al., 2023a), et les nouveaux objectifs d'apprentissage (London and Sandler, 2019; Sakhi et al., 2023a). Nous commençons par étudier l'accélération de cette famille spécifique d'objectifs d'apprentissage et fournissons des procédures d'optimisation en temps logarithmique pour les politiques à article unique, en nous concentrant particulièrement sur les politiques paramétrées avec la fonction de lien softmax.

Chapter 7. Fast Offline Learning for One-Item Recommendation. Dans ce chapitre, nous nous attachons à fournir une méthode pour accélérer l'apprentissage des politiques de softmax à produit scalaire pour un large panel d'objectifs. Nous identifions les problèmes posés par les gradients couramment adoptés et proposons une solution basée sur trois ingrédients : une nouvelle formule de gradient de covariance, l'exploitation de la structure MIPS : Maximum Inner Product Search dans la phase d'apprentissage et la conception d'outils Monte Carlo appropriés (Chopin and Papaspiliopoulos, 2020) pour obtenir des approximations accélérées. Il en résulte un algorithme d'apprentissage avec des mises à jour de gradient sous-linéaires (logarithmiques ou constantes). Nous menons des expériences approfondies sur des ensembles de données de recommandation à grande échelle et démontrons l'impact de notre approche ; la méthode proposée est jusqu'à 25 fois plus rapide que la méthode de base tout en produisant des politiques de qualité similaire. Ce chapitre est basé sur la publication suivante :

- Otmane Sakhi, David Rohde, and Alexandre Gilotte (2023c). Fast Offline Policy Optimization for Large Scale Recommendation. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023.*

Après avoir abordé le problème de l'apprentissage de systèmes de décision à grande échelle à un élément avec des objectifs linéaires, nous étendons notre analyse au cas plus difficile de l'apprentissage de systèmes de décision à ardoise. Au lieu de jouer une action, nos politiques doivent délivrer des ardoises, une liste ordonnée d'éléments de taille $K \geq 1$. Cela signifie que nos règles de décision et nos politiques sont construites pour agir sur l'espace combinatoire $\mathcal{S}_K$ de permutations tronquées à K. La taille de cet espace est $\mathcal{O}(|\mathcal{A}|^K)$ et rend les opérations de base, du calcul d'une moyenne à la recherche de la meilleure ardoise, infaisables. Nous nous concentrons sur une famille de systèmes de décision qui réduisent l'espace de recherche de l'ensemble combinatoirement grand des ardoises $\mathcal{S}_K$ à l'espace d'action original $\mathcal{A}$. Pour un contexte donné x, cette réduction consiste à attribuer un score $f_\theta(a, x)$ à chaque action a et à recommander une liste composée des K premiers éléments ayant les scores les plus élevés. Cela conduit à un temps de livraison de $\mathcal{O}(\log|\mathcal{A}|)$ lorsque nous adoptons la structure de produit scalaire pour $f_\theta(a, x)$. Aouali et al. (2023b) propose une méthode directe pour apprendre les systèmes de recommandation d'ardoises à grande échelle. Dans le chapitre suivant, nous présentons les défis posés par l'apprentissage des systèmes de décision en ardoise et proposons des solutions pour accélérer leur apprentissage.

Chapter 8. Fast Offline Learning for Slate Recommendation. Dans ce chapitre, nous nous concentrons sur l'accélération de l'apprentissage des politiques d'ardoise, un élément omniprésent des systèmes en ligne modernes. Nous commençons par présenter le problème et par analyser les algorithmes existants, leurs hypothèses communes et leurs limites. Nous proposons ensuite une nouvelle classe d'algorithmes, basée sur une nouvelle relaxation qui traite élégamment les contraintes à grande échelle. La méthode résultante fonctionne avec des récompenses arbitraires, possède de meilleures propriétés statistiques tout en réalisant des mises à jour d'apprentissage sous-linéaires. Nous menons des expériences à grande échelle et démontrons que l'approche proposée est plus rapide de plusieurs ordres de grandeur que les lignes de base, tout en produisant des politiques plus performantes. Ce chapitre est basé sur la publication suivante:

- Otmane Sakhi, David Rohde, and Nicolas Chopin (2023b). Fast Slate Policy Optimization: Going Beyond Plackett-Luce. *Transactions on Machine Learning Research.*

Chapter 1

Introduction

1.1 Overview

This manuscript presents recent contributions, ranging from theory to large scale applications, to an offline formalism of the problem of sequential decision-making under uncertainty. An important problem with numerous real-world applications where a decision maker, tasked with solving a specific goal, interacts with an unknown environment, log these interactions and leverage them in order to better solve the task. In this context, we want to answer the following:

How can we leverage previous interactions of the decision-maker to improve its performance?

Answering this question can have a big impact on real world practical problems. For example, it may help a charity online marketing campaign get more donations for a good cause, it may be of service to doctors improving the quality of drug prescription, or it may simply improve the recommendation quality of your favourite music streaming service making it easier to discover new artists. In this introduction, we showcase the problem of learning decision-makers using the example of recommendation, as it will be the focus of a big part of this thesis. Recommender systems are the backbone of the modern internet experience. In each interaction, these systems silently navigate an overwhelming amount of information and filter it to cater to the specific needs of the user. An interaction of a recommendation engine can be summarized in the following: the system encounters a user, the system chooses an item (or multiple items) to recommend from a potentially large catalogue and observes a feedback.

The feedback received is valuable as it represents successes and failures of past interactions. These interactions are logged and are later used to improve the recommendation quality of the system. The interactive nature of the collected dataset makes common learning paradigms, such as supervised learning, not adapted to study such problem. Recently, there has been an interest in adapting sequential decision-making framework to improve recommendation based on the log of interactions. Reinforcement learning (RL) (Sutton and Barto, 2018) and Contextual bandits (CB) (Lattimore and Szepesvári, 2020) start to take the spotlight as good candidates to model this learning problem. The RL framework builds on the idea that performed actions may impact the environment. This paradigm can model complex sequential decision problems, is versatile and allows for planning. Its tools can optimize recommender systems for long term metrics; for example, increase user engagement and retention (Afsar et al., 2022). This versatility however comes with a cost. Taking into account the long term effects of recommendation on users makes the analysis more difficult, prompting us to consider a simpler formalism. Contextual Bandit offers a useful compromise between principled analysis and practical impact. Its underlying assumption is that actions made by the system do not influence future outcomes. If this

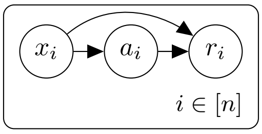


Figure 1.1: The logged dataset $\mathcal{D}_n$ representing n interactions of the recommender system. All (context, action, reward) triplets are independent.

formulation is less compelling when dealing with delayed rewards (Afsar et al., 2022), its use is reasonable if we want to focus on learning recommender systems that optimize short-term, action-bounded metrics, such as click-through rate (Sakhi et al., 2020a) or watch time (Chen et al., 2019a). In this thesis, we adopt the offline contextual bandits' (Bottou et al., 2013; Nguyen-Tang et al., 2022) toolbox to formalize learning from interaction logs. We give new principled approaches to learn policies with strong performance guarantees and propose new algorithms to widen the impact of this framework to large scale, real world applications.

An interaction of a user with a recommended item can be reduced to the following example. A user navigates a website, the recommender system chooses an item from a catalogue and shows it to the user, the user interacts with the item (either clicks or not) and the result of this interaction is encoded in a feedback (presence/absence of click) that the system logs. Within the Contextual Bandit framework, a user is represented by a context x, usually a real vector living in a d-dimensional space $\mathcal{X} \subseteq \mathbb{R}^d$. These contexts, and thus users, are sampled *independently* from the same, unknown distribution $\nu(\mathcal{X})$. After seeing a user, the recommendation engine delivers an item a from a catalogue $\mathcal{A}$ of size $|\mathcal{A}| \in \mathbb{N}$. The recommender system is modelled as a policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$, which is a function that takes a context x and produces a distribution $\pi(\cdot|x)$ over the space of possible actions $\mathcal{A}$. Recommending an item a for the context x boils down to sampling the item from the produced distribution $a \sim \pi(\cdot|x)$. After delivering the item a to the user of context x, our system receives feedback; a stochastic reward $r \in \mathbb{R}^+$ coming from an unknown distribution $p(\cdot|x, a)$. This reward encodes how well the recommended item has performed on our desired metric; the higher the reward, the higher the performance. Our goal is to find policies of great performance, achieved by minimizing the risk, defined as the expected negative reward under the actions of the policy. The risk of any given policy π can be expressed as:

$$R(\pi) = -\mathbb{E}_{x \sim \nu, a \sim \pi(\cdot|x)} \left[\mathbb{E}_{r \sim p(\cdot|x,a)}[r] \right].$$

This risk is an expectation under actions taken by the policy evaluated. As we do not have access to interactions of the new policy π with the environment, a simple way to estimate this quantity is to let π interact with users online. In most scenarios, this is not possible, as we do not have the luxury to deploy bad policies. In real world applications, we already have the current version of our recommender system, represented by the policy π_0, that interacts with the environment and logs the feedbacks. Our primary focus is to assess how well a new iteration of the system will improve upon the currently deployed version. A common way to achieve this is by conducting online A/B-tests (Kohavi et al., 2012). This is considered the "gold standard" approach to estimate the effect of replacing the current policy π_0 by a potentially better one (Gupta et al., 2019). A/B-tests however require substantial engineering effort, constant monitoring and need several days to be properly analysed. Ideally, we want offline evaluation and learning tools that can give us promising policies to reduce the number of unnecessary A/B-tests. When the contextual bandit assumptions are satisfied, we can use the framework's toolbox to achieve this (Bottou et al., 2013). The idea is to leverage the existing interactions of π_0 to find policies of greater performance. The interaction dataset is called in the literature the logged bandit feedback dataset (Swaminathan and Joachims, 2015a):

$$\mathcal{D}_n = \{x_i \sim \nu, a_i \sim \pi_0(\cdot|x_i), r_i \sim p(\cdot|x_i, a_i), \pi_0(a_i|x_i)\}_{i\in[n]}.$$

A graphical representation of the data is shown in Figure 1.1. The main challenge encountered when learning from this data is the potential bias created by the collection procedure; we only have access to the outcome of actions sampled from π_0. The offline learning framework of contextual bandit offers two distinct approaches to solve this issue; the model-based approach and the importance weighting approach.

The model-based approach or *the direct method* leverages the interaction data $\mathcal{D}_n$ to construct a reward model (Sakhi et al., 2020a; Jeunen and Goethals, 2021). An optimal policy is then naturally derived by playing for each context x, the action with the highest reward according to the model. The direct method is straightforward to implement, as it reduces the learning to a regression problem (Brandfonbrener et al., 2021). This approach is theoretically well-studied and benefits from strong guarantees (Nguyen-Tang et al., 2022). However, it will suffer from a substantial, uncontrolled bias whenever the reward is complex, making its efficiency entirely dependent on our ability to model the problem's structure. The direct method is efficient when we are confident in our ability to understand the problem. When the reward signal is complex, we may prefer another approach that does not completely rely on our modelling effort.

The Importance-weighting approach (Horvitz and Thompson, 1952; Bottou et al., 2013; Dudík et al., 2014), often called *off-policy learning*, is agnostic to the reward model. It learns a new policy π directly from the interactions $\mathcal{D}_n$ using estimators corrected with importance sampling (Chopin and Papaspiliopoulos, 2020). Under mild assumptions (Horvitz and Thompson, 1952), this method can produce unbiased estimators, which are arguably easier to analyse and optimize (Ajalloeian and Stich, 2020). These estimators however suffer from a potentially large variance once the learned policy drifts away from the logging policy π_0, making them unreliable for learning. It is empirically proven that learning with these estimators can result in bad performing policies (Swaminathan and Joachims, 2015a,b), sometimes even worse than π_0 (Chen et al., 2019b; London and Sandler, 2019). This observation motivates the use of learning theory tools (Zhou, 2002; McAllester, 1998) to come up with principled objectives that are agnostic to the reward structure. The objective of this research effort is to produce new policies that are **provably better** than the logging policy without engaging with the environment. This is beneficial in production settings where we would like to propose a new system, that will improve on the current production system with high probability.

The first effort in this direction was driven by Swaminathan and Joachims (2015a) and resulted in the **CRM: Counterfactual Risk Minimization** principle. The **CRM** principle builds on tools from Statistical Learning Theory (Vapnik, 1998), a framework that has great success studying the generalization ability of learning algorithms. Motivated by the construction of an Empirical Bernstein Upper Bound (Maurer and Pontil, 2009) on the true risk of policies, and using covering number arguments (Zhou, 2002), this principle advocates for penalizing importance weights estimators with their *square-root sample variance*. This penalty is controlled by a hyperparameter λ that needs to be cross-validated on a hold-out set. The underlying intuition is that to improve on the logging policy, we should look for policies that have a *small empirical risk* while staying close to the logging policy π_0. This principle results in better performing policies compared to optimizing crude importance weighting estimators (Swaminathan and Joachims, 2015a,b). However, its learning paradigm suffers from different limitations, hindering its applicability to simple scenarios. In particular, adding the sample variance penalization makes the learning objective non-convex and non-decomposable, which forbids the use of stochastic gradient methods. This penalty is also controlled with a new hyperparameter λ that is difficult to tune and adds to the complexity of the approach. Finally, the **CRM** principle fails to provide performance certificates on the newly trained policy. These limitations will be discussed in details later in the introduction. More recently, a new principle was introduced to mitigate some of these limitations. By analysing this learning problem from the PAC-Bayesian lens (McAllester, 1998; Alquier, 2021), London and Sandler (2019) develop an improved approach. The authors build their analysis around McAllester (2003)'s PAC-Bayesian bound. For parametric policies, this motivates an L_2 regularization of the parameter of the new policy towards the parameter of the logging policy π_0. The regularization is also controlled by a hyperparameter λ that requires tuning. This principle is based on the same intuition of staying close to π_0, but this time, it is carried out on the parameter space. The adoption of an L_2 regularization instead of a *sample variance penalization* makes the optimization smoother and allows the use of stochastic gradient descent. However, the L_2 regularization parameter λ suffers from the same limitations and the principle cannot produce provably better policies. Empirical findings demonstrate that these principles sometimes fail at improving the logging policy π_0 (Chen et al., 2019b). These limitations will be developed even further in the next section, before we present the contributions of the first part of the thesis. Chapter 3 reframes **CRM** using tools from **Distributionally Robust Optimization** (Duchi et al., 2021), a statistical framework designed for decision-making under uncertainty. Furthermore, Chapters 4 and 5 build on London and Sandler (2019)'s work and continue the development of **PAC-Bayesian** tools (McAllester, 1998) for offline contextual bandit. The analysis yields principles that are easier to optimize, do not require additional hyperparameters to tune and enjoy, for some, even better performance guarantees, taking us a step closer to learn **provably better** policies offline.

In real world problems, interactive systems often deal with large scale scenarios, where they need to learn from enormous number of interactions ($n \gg 1$) and operate on massive catalogues ($|\mathcal{A}| \gg 1$). For these systems to deliver recommendations in a matter of milliseconds, they are restricted to a certain structure (Shrivastava and Li, 2014; Aouali et al., 2022) to allow for rapid query response. For a long time, large scale recommender systems were trained for preference prediction (Harper and Konstan, 2015; Gomez-Uribe and Hunt, 2016) or next-item prediction (Hidasi et al., 2015; Wu et al., 2019). These modelling approaches are usually considered poor proxies to the reward we are interested to optimize (Jannach and Jugovac, 2019). Adapting the offline contextual bandit toolbox to learn large scale recommender systems will have a great impact on the industry. These tools can enable the alignment of recommendation with complex reward signals, enhancing both user satisfaction and the profitability of the businesses operating these systems. As presented earlier, we can either adopt the **direct method** if we know how to model the reward, or **importance weighting estimators with learning principles** to learn a policy directly. Both can reliably learn a performing recommender system. Unfortunately, these methods in their simple form present some caveats when dealing with large scale problems. The second part of the thesis addresses these limitations and allows for efficient and fast training of reward optimizing, large scale recommender systems.

The direct method relies completely on our ability to learn a model that reflects the properties of the reward. Understanding perfectly the problem lowers the bias linked to modelling, but there is another problem, linked to learning from $\mathcal{D}_n$, that becomes more pronounced in large catalogue scenarios. Indeed, naively learning the reward model using $\mathcal{D}_n$ suffers from the unbalance present in the collected data. The reward model will be well estimated for actions that are likely to be sampled under π_0, and poorly estimated for the rest. This difference in the estimation quality can make the decisions taken by the derived policy unreliable (Smith and Winkler, 2006). This phenomenon is accentuated when dealing with large catalogue sizes, as π_0 can never collect enough samples to cover the whole action space. We dedicate Chapter 6 to discuss a Bayesian solution to this issue. We introduce structure to the model and use another valuable source of data to efficiently learn the reward model. More details about this approach can be found in the contribution section.

Importance-weighting objectives become interesting when the reward signal is complex. However, in large scale scenarios, these learning objectives suffer from two major caveats. The first issue is linked to the variance of these importance-weighting estimators, which grows with the size of the action space. Indeed, the variance of common importance weighting estimators (Horvitz and Thompson, 1952; Ionides, 2008; Dudík et al., 2014) become uncontrollable when the policies operate on massive catalogues (Saito and Joachims, 2022b). As this variance can be very large, adding a variance penalization for example will force the newly learned policy π to mimic the behaviour of π_0. This phenomenon makes our learning principles too conservative, returning policies very close to π_0. This observation motivated the construction of a new family of importance weighting estimators (Saito and Joachims, 2022a; Saito et al., 2023) to mitigate this variance problem. These recent contributions deal with the statistical limitations of importance-weighting objectives in large catalogue scenarios, but computational issues linked to optimizing these objectives remain unsolved. The importance weighting approach learns policies directly, and use gradient-based methods to computationally optimize the learning objectives. Large scale systems are updated frequently, and fast optimization routines are highly desirable in this context. Existing methods offer gradient iterations that scale at least linearly on the catalogue size. This complexity can be detrimental to learning recommender systems operating on billions of items. The last two chapters (Chapters 7 and 8) focus on the computational aspect and propose optimization routines with sublinear complexities. These solutions will be developed more in the contribution section.

We cover different, connected disciplines in this thesis while balancing between theoretical tools and practical algorithms. To ease the presentation, we want to give readers an overview of the advancement of each research field. To this end, we dedicate a chapter to review existing literature, that we deem valuable to researchers, whether they come from a theoretical or practical background.

Chapter 2. Literature Review. This chapter conducts a literature review to cover the different tools used throughout this thesis. We give a brief overview of the literature of Contextual Bandit, a practical formalism to study reward-driven recommendation, presenting both its online and offline formulations. With a focus on the offline setting, we dedicate a section to present statistical learning tools, necessary to study learning decision systems with online performance guarantees. We then present the development of recommender systems and how modelling recommendation shifted from predicting preferences to reward maximization, and conclude with the algorithmic considerations that arise in the context of large scale decision-making.

Part I - Offline Learning with Performance Guarantees. The first part of the thesis focuses on addressing the limitations of current learning principles. These principles were proposed to allow learning policies that improve on the logging policy π_0, mitigating the problems of importance weighting approaches. Without any loss of generality, we present the problem with the help of the `IPS`: *Inverse Propensity Scoring* estimator (Horvitz and Thompson, 1952), arguably the simplest and most studied estimator. For a policy π, we recall its expression:

$$\hat{R}_n^{\texttt{IPS}}(\pi) = -\frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)} r_i.$$

When evaluated on π_0, `IPS` gives the empirical mean of the collected costs as an estimation of the risk, which is considered to be a well-behaved, unbiased estimator of $R(\pi_0)$. However, a simple analysis of the variance of this estimator demonstrates that drifting away from π_0 leads to poorer estimation quality. If the observed reward is bounded (i.e. $r \in [0,1]$), we have:

$$\begin{aligned}\mathbb{V}\left[\hat{R}_n^{\texttt{IPS}}(\pi)\right] &= \frac{1}{n}\left(\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x), r\sim p(\cdot|x,a)}\left[\left(\frac{\pi(a|x)}{\pi_0(a|x)}\right)^2 r^2\right] - \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right]^2\right)\\ &\leq \frac{1}{n}\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x), r\sim p(\cdot|x,a)}\left[\left(\frac{\pi(a|x)}{\pi_0(a|x)}\right)^2\right] = \frac{1}{n}\left(\chi^2(\pi,\pi_0)+1\right),\end{aligned}$$

with $\chi^2(\pi,\pi_0)$ the χ-Square divergence between π and π_0. The variance has roughly the same behaviour as the χ-Square divergence, growing when π is far from π_0. In particular, the variance grows with the importance weights. Importance weights are very large when the new policy π assigns high probability to actions that were very unlikely to be played under π_0. This means that the estimation quality is highly dependent on the policy evaluated π, making `IPS`[1] untrustworthy for policies far from the neighbourhood of π_0. This observation is confirmed in practice, especially when using importance weights-based estimators as a learning objective. For example, minimizing the `IPS` estimator with respect to a policy class can lead to policies with bad online performance. When the learned policy π is far from π_0, the `IPS` estimation of the risk of π will not reflect its true risk, as π will lie in a part of the space that induces an estimator with large variance. To circumvent these limitations, one would want to restrict the optimization to policies around the logging policy π_0. The **CRM: Counterfactual Risk Minimization** Principle (Swaminathan and Joachims, 2015a) formalizes this idea using statistical learning arguments. Motivated by the construction of an Empirical Bernstein Bound (Maurer and Pontil, 2009) covering the true risk of policies in a class of policies, the principle advocates for minimizing a *sample variance penalized* `IPS` estimator. This principle is then used to produce a tractable algorithm for learning, parametrized softmax policies (Mei et al., 2020b) of the form:

$$\begin{aligned}\forall(x,a) \quad \pi_\theta(a|x) &= \texttt{softmax}_{\mathcal{A}}\left(f_\theta(x,a)\right)\\ &= \frac{\exp(f_\theta(x,a))}{\sum_{a'\in\mathcal{A}}\exp(f_\theta(x,a'))}.\end{aligned} \tag{1.1}$$

with θ a parameter coming from a parametric space $\theta \in \Theta$ and $f_\theta : \mathcal{X}\times\mathcal{A}\to\mathbb{R}$ a function that encodes the relevance of action a to the context x. The algorithm proposed is named **POEM**: Policy Optimizer for Exponential Models (Swaminathan and Joachims, 2015a) and solves the following objective for softmax policies:

$$\arg\min_{\theta\in\Theta}\left\{\hat{R}_n^{\texttt{IPS}}(\pi_\theta) + \lambda\sqrt{\hat{V}^{\texttt{IPS}}(\pi_\theta)}\right\},$$

with λ a tuning parameter usually set with the help of a validation split, and $\hat{V}^{\texttt{IPS}}(\pi_\theta)$ the sample variance term induced by evaluating π with `IPS`:

$$\hat{V}^{\texttt{IPS}}(\pi_\theta) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}r_i + \hat{R}_n^{\texttt{IPS}}(\pi_\theta)\right)^2.$$

[1] All importance weights based estimators suffer from the same caveat.

Swaminathan and Joachims (2015a) reported empirically the superiority of this principle; the policies returned by **POEM** have much lower risk than those obtained by naively minimizing the `IPS` objective. Adding the sample variance regularizer makes the approach more principled, but still suffers from limitations that reduce its applicability in real-life scenarios:

(1) Scalability. The biggest limitation of the CRM principle is its scalability to large logged datasets $\mathcal{D}_n$. The presence of the sample variance term makes computing the gradient of the CRM objective scale in $\mathcal{O}(n)$ in both computational and memory cost, as it requires going through the entire dataset. In a recommender system scenario, millions of interactions are logged daily. Such applications deal with extremely large number of samples n and cannot afford the cost of these computations. This issue is usually solved by resorting to stochastic optimization algorithm, which requires only access to unbiased stochastic gradients of the objective. Those are particularly easy to obtain when the objective decomposes into a sum over the dataset's entries, as computing the sum on batches of the dataset is enough to obtain unbiased gradients. Unfortunately, the CRM objective is not suited for stochastic optimization, as the square-root empirical variance term does not write as a sum. Swaminathan and Joachims (2015a) proposed a relaxation of the CRM objective, based on a majorization-minimization strategy, that can benefit *partially* from stochastic gradients. Their approach still requires passing through the whole logged dataset once in a while, obtaining a procedure of the same computational complexity.

London and Sandler (2019) propose an improved principle that deals with the first limitation of CRM. Instead of relying on Maurer and Pontil (2009)'s Empirical Bernstein Bound, London and Sandler (2019) adapts McAllester (2003)'s PAC-Bayesian bounds to derive learning objectives for this problem. The derived principle motivates the use of an L_2 regularization towards the parameter θ_0 of the logging policy π_0. This regularization is controlled by a hyperparameter λ, giving the following optimization problem for parametrized softmax policies:

$$\arg\min_{\theta\in\Theta} \left\{ \hat{R}_n^{\texttt{IPS}}(\pi_\theta) + \lambda\|\theta - \theta_0\|^2 \right\},$$

The optimization objective is amenable to stochastic optimization (decomposable into a sum), scales to large datasets and returns policies with better empirical performance. However, this principle, like CRM, suffers from other limitations, presented in the following:

(2) No Performance Guarantees. Both principles derived are motivated by the construction of bounds covering the true risk of policies. These bounds in their raw form cannot be used directly as a learning objective. Indeed, the bound derived in Swaminathan and Joachims (2015a) contains an intractable quantity and the one derived in London and Sandler (2019) is vacuous. Introducing the hyperparameter λ helps us obtain practical objectives, that lose the theoretical guarantees given by the initial bounds. These objectives do not necessarily cover the true risk, and optimizing them can lead to policies worse than the logging π_0. Empirical evidence can be found in (Chen et al., 2019b) where the CRM principle fails to improve on π_0.

(3) Hyper-parameter Selection. Another major problem of these principles is also caused by the introduction of λ and it is selected. The free-parameter λ requires careful tuning, as its choice drastically impacts the performance of the obtained policy. As there are no theoretical guidelines to define a good value of λ, the strategy consists of cross-validating the parameter over a relatively fine grid using the `IPS` estimator. Cross validation adds to the complexity of the algorithm. This procedure also requires having a hold-out set that will not be used for training, accentuating the variance problem. In addition, as the `IPS` estimator is still used to select the best value of λ, the policy returned in the end is the one that minimizes the `IPS` risk on the validation set, which renders the whole principle incoherent.

The goal in this first part is to provide practitioners with better principles that circumvent such limitations altogether, enjoying better statistical guarantees and empirical performance.

Chapter 3. **Offline Learning with Distributionally Robust Optimization.** In this chapter, we present an alternative formulation to the CRM principle by resorting to the distributionally robust optimization (DRO) framework (Duchi et al., 2021). These tools enable elegant construction of variance-sensitive confidence upper-bounds on the true risk by using f-divergence based ambiguity sets. We apply this principle to the problem of offline policy evaluation and optimization. The resulting objective deals with limitations **(1)** and **(3)**; it enjoys the same statistical guarantees than CRM, can be automatically calibrated using asymptotic coverage arguments and is amenable to stochastic optimization. We display strong numerical experiments showing that the proposed approach effectively deals with the shortcomings of CRM. This chapter is adapted from the following publication:

- Otmane Sakhi, Louis Faury, and Flavian Vasile (2020b). Improving Offline Contextual Bandits with Distributional Robustness. *Proceedings of the ACM RecSys Workshop on Reinforcement Learning and Robust Estimators for Recommendation Systems, 2020.*

Chapter 4. **Offline Learning with PAC-Bayesian Theory.** In this chapter, we question the off-policy learning paradigm completely and advocate for a theoretically-grounded strategy to confidently improve on the deployed policy π_0. The proposed method revolves around creating tight, empirical lower bounds on the improvement $\mathcal{I}(\pi) = R(\pi_0) - R(\pi)$, and deploying new policies only when we are confident of $\mathcal{I}(\pi) > 0$. We base our approach on PAC-Bayesian learning theory (Alquier, 2021) and demonstrate that its tools suit perfectly the problem of off-policy learning. In particular, by interpreting policies as mixtures of decision rules, we derive a tight, Bernstein-type PAC-Bayes bound that makes our strategy viable. The resulting strategy deals with all three limitations; we show that the resulting algorithm can give improvement certificates, is amenable to stochastic optimization and does not require any hyperparameter tuning, making a big step towards achieving practical off-policy learning with true performance guarantees. This chapter is based on the following publication:

- Otmane Sakhi, Pierre Alquier, and Nicolas Chopin (2023a). PAC-Bayesian Offline Contextual Bandits with Guarantees. *Proceedings of the 40th International Conference on Machine Learning, 2023.*

Chapter 5. **A Better PAC-Bayesian Analysis of Offline Learning.** In this chapter, we continue the development of the PAC-Bayesian analysis of the problem of offline policy learning. By exploiting the negative nature of the risk, we derive new, tighter bounds that hold for a larger class of risk estimators. The idea is based on a refined treatment of the moment generating function of the risk and extend empirical Bernstein bounds to higher orders. The particularity of these results is that they are fully empirical; we do not assume access to π_0 contrary to previously derived bounds. We observe that our findings can give better guarantees and allow us to derive new insight about the estimators used. This chapter is focused on providing technical results and is based on new, unpublished work.

Part II - Offline Learning of Large Scale Recommendation. The offline learning setting provides practical solutions to efficiently align decision systems with complex reward signals. If the research community has focused on improving the existing estimators and learning paradigms, little attention was directed towards adapting these approaches to the large action space setting. This is of interest to learning search engines, recommender systems and practically any application where the number of interactions n and the size of the action space $|\mathcal{A}|$ are massive. The main challenge in these applications is to design decision rules that satisfy engineering constraints, while providing tractable algorithms that enable their alignment with reward signals in a fast and reliable manner. Search engines must answer queries in a matter of milliseconds, and real-world recommender systems (think of a video streaming platform) must seamlessly fill the landing page with content the user may like. These delivery speed constraints should be respected even if the catalogue (action space) contains billions of items. Another aspect to take into consideration is that these systems are updated frequently, putting a considerable constraint on the training time of such systems. Indeed, if we need to update our decision system *daily* based on its interactions, the training time should be substantially smaller than a *day* as you need to collect enough interactions and update the system in the same time frame. In their naive implementations, both the decision-making and training of these systems scale linearly in the size of the action space $\mathcal{O}(|\mathcal{A}|)$, which cannot be allowed in massive action space scenarios. In the following, we develop the discussion around these two important aspects and present our contributions in this field.

Fast Decision Making. Answering a query or delivering accurate recommendations is performed by the policy deployed. No matter the nature of the policy and its delivery, this step generally boils down to the *fast* identification of a sub-sample of size $K \geq 1$ of good actions (Chen et al., 2019a) from the potentially massive action space. As a general rule, and for a user x, the quality of the actions is encoded in the score function $f_\theta(\cdot, x) : \mathcal{A} \to \mathbb{R}$ while the good actions are identified by finding the best scoring actions, solving the following:

$$[a_1, ..., a_K] = \underset{a' \in \mathcal{A}}{\overset{K}{\arg\mathrm{sort}}} \left\{ f_\theta(a', x) \right\}, \tag{1.2}$$

with the operator $\arg\mathrm{sort}^K_{a' \in \mathcal{A}}$ returning the K highest scoring actions. This sorting operation has a linear complexity on the size of the action space $\mathcal{O}(|\mathcal{A}| \log K)$ and cannot be adopted in a large scale production environment. The common solution to reduce this complexity is to impose a structure for the score function. By restricting the score function space to the following:

$$\forall (x, a) \quad f_\theta(a, x) = h_\Xi(x)^\intercal \beta_a$$

with $\theta = [\Xi, \beta]$, the score function becomes an inner product between a context embedding $h_\Xi(x)$ and an action embedding β_a, both residing in a latent space $\mathbb{R}^l$ of dimension $l \ll |\mathcal{A}|$. With this structure, Equation (1.2) can be solved with *approximate* MIPS: Maximum Inner Product Search algorithms (Shrivastava and Li, 2014) in a time complexity of $\mathcal{O}(\log |\mathcal{A}|)$ instead of $\mathcal{O}(|\mathcal{A}|)$, rendering fast decision-making possible without additional considerations.

Efficient training with the direct method. The direct method derives an optimal policy that depends on identifying the best scoring actions according to the reward model $r_\mathcal{M}$. This means that if $r_\mathcal{M}$ has the proper parameterization, fast decision-making is possible. Training a good model $r_\mathcal{M}$ requires an excellent understanding of the underlying problem and is usually achieved through *maximum likelihood estimation* (Aouali et al., 2023b) or *ranking-based heuristics* (Rendle et al., 2009), which are methods that scale independently of the action space size.

These training algorithms however can have other shortfalls if we are not careful about the particularities of this setting. In large action space scenarios, it is impossible for the deployed policy to collect enough interactions for each action in $\mathcal{A}$. The reward signal is unevenly distributed, as the majority of data collected comes from actions that are highly likely under π_0 and little to no data is available for the rest of the actions. This means that if we use the *maximum likelihood principle*, the quality of the learned model $r_{\mathcal{M}}$ will depend on the context/action pair; the estimate is precise for action/context pairs that are present enough in the data. This unbalance in the estimate quality negatively impacts the policy derived, as acting based on the *MLE* might suffer from post-decision disappointment (Smith and Winkler, 2006). One principled way to mitigate this issue is to frame the whole problem within the lens of *Bayesian* decision theory (West et al., 2021). For example, Jeunen and Goethals (2021) demonstrate that even simple Bayesian modelling of the reward result in better behaved policies. With the help of well-chosen priors, this formulation can also incorporate additional correlations we have between contexts and actions, making learning even more efficient (Aouali et al., 2023c). However, the main challenge of Bayesian modelling is computational; approximating posteriors over billions of interactions, using complex models and priors is difficult and needs particular care (Chopin and Papaspiliopoulos, 2020). We dedicate a chapter to develop this discussion, and construct a Bayesian reward model for recommendation with strong, data-driven priors while giving proper tools to accelerate its training in large scale applications.

Chapter 6. Scalable Bayesian Reward Modelling. In this chapter, we take the path of the direct method and develop a bayesian model of the reward for the case of one-item recommendation. We acknowledge the presence of two types of signals in recommendation problems; the organic and bandit signals. While we condition our model on the the bandit feedback, the organic interactions between the contexts and actions help us construct a novel prior that incorporates three similarities: the context-action similarity, the action-action similarity and the context-context similarity. These similarities allow us to obtain good estimates of the reward on all regions of the space, even for less explored actions and contexts. The proposed model is flexible, efficiently uses the existing data but produces an intractable posterior. We provide easy-to-implement computational tools to approximate its solution based on ideas from Variational Bayes (Blei et al., 2017). The resulting algorithm scales to large datasets, can learn efficiently in different scenarios and benefits from the inner-product parametrization, allowing fast decision-making. This chapter is based on the following publication:

- Otmane Sakhi, Stephen Bonner, David Rohde and Flavian Vasile (2020a). BLOB: A Probabilistic Model for Recommendation that Combines Organic and Bandit Signals. *KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Fast training with importance-weighting methods. In our pursuit for scalable offline policy learning algorithm, we also express interest for importance weighting based learning paradigms. This approach learns a policy directly, and even simple operations involve computing sums over the whole action space. In particular, we need to be extra-careful when computing/approximating gradients of our objectives because this operation scales linearly in $|\mathcal{A}|$ which can drastically slow down the optimization routine. The question of scaling general off-policy learning objectives attracted little attention; Chen et al. (2019a) learned a production-ready policy with an `IPS`-based objective without any focus on the computational aspect. We are interested in this question and want to provide general acceleration methods. We study the specific family of objectives that can be written as expectations under the policy evaluated. This family include commonly adopted estimators (Horvitz and Thompson, 1952; Dudík et al., 2014; Wang et al., 2017; Saito and Joachims, 2022a; Saito et al., 2023; Aouali et al., 2023a), and principled learning objectives (London and Sandler, 2019; Sakhi et al., 2023a). We first study the acceleration of this specific family of learning objectives and provide logarithmic time optimization procedures for single-item policies, focusing particularly on policies parameterised with the softmax link function.

Chapter 7. **Fast Offline Learning for One-Item Recommendation.** In this chapter, we focus on providing a principled way to accelerate the learning of inner-product softmax policies for a large panel of off-policy objectives. We identify the problems of commonly adopted gradients and propose a solution based on three ingredients; a new covariance gradient formula, exploiting the `MIPS`: Maximum Inner Product Search structure in the training phase and designing proper Monte Carlo tools (Chopin and Papaspiliopoulos, 2020) to achieve accelerated approximations. This results in a training algorithm with sub-linear (logarithmic or constant) gradient updates. We conduct extensive experiments on large scale recommendation datasets and demonstrate the impact of our approach; the proposed method is up to 25 times faster than the baseline while producing trained policies of similar quality. This chapter is based on the following publication:

- Otmane Sakhi, David Rohde, and Alexandre Gilotte (2023c). Fast Offline Policy Optimization for Large Scale Recommendation. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023.*

After attacking the problem of training large scale one-item decision systems with linear objectives, we extend our analysis to the more challenging case of training slate decision systems. Instead of playing one action, our policies need to deliver slates; an ordered list of items of size $K \geq 1$. This means that our decision rules and policies are constructed to act on the combinatorial space $\mathcal{S}_K$ of K-truncated permutation. The size of this space is $\mathcal{O}(|\mathcal{A}|^K)$ and makes basic operations, from computing an average to searching for the best slate infeasible. We focus on a family of decision systems that reduce the search space from the combinatorially large set of slates $\mathcal{S}_K$ to the original action space $\mathcal{A}$. For a given context x, this reduction consists of assigning a score $f_\theta(a, x)$ to each action a and recommend a slate composed of the top-K items with the highest scores. This leads to a $\mathcal{O}(\log|\mathcal{A}|)$ delivery time when we adopt the inner-product structure for $f_\theta(a, x)$. Aouali et al. (2023b) suggest a direct method approach to learn large scale slate recommendation systems. In the next chapter, we present the challenges of learning slate decision systems and propose solutions to accelerate their training.

Chapter 8. **Fast Offline Learning for Slate Recommendation.** In this chapter, we focus on accelerating the learning of slate policies, a ubiquitous building block of modern online systems. We begin by introducing the problem and analysing the existing algorithms, their common assumptions and limitations. We then propose a new class of algorithms, based on a novel relaxation that deals elegantly with the large scale constraints. The resulting method works with arbitrary rewards, has better statistical properties while achieving sub-linear training updates. We conduct large scale experiments and demonstrate that the proposed approach is orders of magnitude faster than the baselines while resulting in better performing policies. This chapter is based on the following publication:

- Otmane Sakhi, David Rohde, and Nicolas Chopin (2023b). Fast Slate Policy Optimization: Going Beyond Plackett-Luce. *Transactions on Machine Learning Research.*

1.2 List of Publications

Journal/Conference Articles

1. Otmane Sakhi, David Rohde and Nicolas Chopin (2023b). Fast Slate Policy Optimization: Going Beyond Plackett-Luce. *Transactions on Machine Learning Research.*

2. Otmane Sakhi, Pierre Alquier and Nicolas Chopin (2023a). PAC-Bayesian Offline Contextual Bandits with Guarantees. *Proceedings of the 40th International Conference on Machine Learning, 2023.*

3. Otmane Sakhi, David Rohde and Alexandre Gilotte (2023c). Fast Offline Policy Optimization for Large Scale Recommendation. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023.*

4. Otmane Sakhi, Stephen Bonner, David Rohde and Flavian Vasile (2020a). BLOB: A Probabilistic Model for Recommendation that Combines Organic and Bandit Signals. *KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Workshop Articles

1. Otmane Sakhi, Louis Faury and Flavian Vasile (2020b). Improving Offline Contextual Bandits with Distributional Robustness. *Proceedings of the ACM RecSys Workshop on Reinforcement Learning and Robust Estimators for Recommendation Systems, 2020.*

2. Otmane Sakhi, Stephen Bonner, David Rohde, Flavian Vasile (2019). Reconsidering Analytical Variational Bounds for Output Layers of Deep Networks. *4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.*

Preprints

1. Imad Aouali, Achraf Ait Sidi Hammou, Sergey Ivanov, Otmane Sakhi, David Rohde, Flavian Vasile (2023b). Probabilistic Rank and Reward: A Scalable Model for Slate Recommendation. *arXiv preprint.*

Tutorials

1. Imad Aouali, Amine Benhalloum, Martin Bompaire, Achraf Ait Sidi Hammou, Sergey Ivanov, Benjamin Heymann, David Rohde, Otmane Sakhi, Flavian Vasile, Maxime Vono (2022). Reward Optimizing Recommendation using Deep Learning and Fast Maximum Inner Product Search. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

2. Flavian Vasile, David Rohde, Olivier Jeunen, Amine Benhalloum, and Otmane Sakhi (2021). Recommender Systems Through the Lens of Decision Theory. *Companion Proceedings of the Web Conference 2021.*

3. David Rohde, Flavian Vasile, Sergey Ivanov, Otmane Sakhi (2020). Bayesian Value Based Recommendation: A modelling based alternative to proxy and counterfactual policy based recommendation. *Proceedings of the 14th ACM Conference on Recommender Systems.*

Chapter 2

Literature Review

2.1 The Landscape of Contextual Bandit

2.1.1 The Online Setting

A (stochastic)[1] contextual bandit is a powerful sequential decision-making framework where an agent interacts with an unknown environment for $T \in \mathbb{N}^*$ rounds. This environment provides contexts (user information, web page, etc) and a set of available actions $\mathcal{A}$ that our agent can make. In each round, the agent observes a context $x \in \mathcal{X}$, acts by taking an action a and receives a feedback; a reward $r \in \mathbb{R}^+$ that depends on both the action and the context observed, coming from a *fixed, but unknown distribution*. The particularity of this setting compared to classical supervised learning is that we observe partial feedback; we get access to the reward associated with the context and the action made by the agent and nothing more. Formally, for each round $t \in [T]$:

- The environment reveals a context $x_t \in \mathcal{X}$ coming from an unknown distribution ν.
- The agent acts on the context x_t by making action a_t. The agent is represented by a stochastic policy $\pi_t : \mathcal{X} \to \mathcal{P}(\mathcal{A})$, that given the context x_t, defines a probability distribution $\pi_t(\cdot|x_t) \in \mathcal{P}(\mathcal{A})$ over the space of available actions $\mathcal{A}$. Acting boils down to sampling from the policy given the context x_t; $a_t \sim \pi_t(\cdot|x_t)$.
- Making action a_t for the context x_t reveals a reward $r_t \in \mathbb{R}^+$ coming from an unknown distribution $r_t \sim p(\cdot|a_t, x_t)$.
- The feedback received r_t updates the policy π_t.

Every interaction helps the agent learn about the environment and improves it *online* to better act in the future. The contextual bandit framework is flexible and can model various problems. However, it is noteworthy to point out that it relies on the fundamental assumption that **the problem is stateless**: actions made by the agent do not affect the environment; for each round t, both contexts and rewards are drawn i.i.d. as the action a_t does not influence ν. This makes contextual bandit not suitable for problems that require long-term planning, for which we can use the more general framework of Reinforcement Learning. We direct the reader to Sutton and Barto (2018) for a great introduction to the field. With these assumptions in mind, we want our agent to achieve a goal that the practitioner is interested in. Depending on the application, we are interested in either maximising the expected cumulative reward after

[1] Different from the adversarial setting.

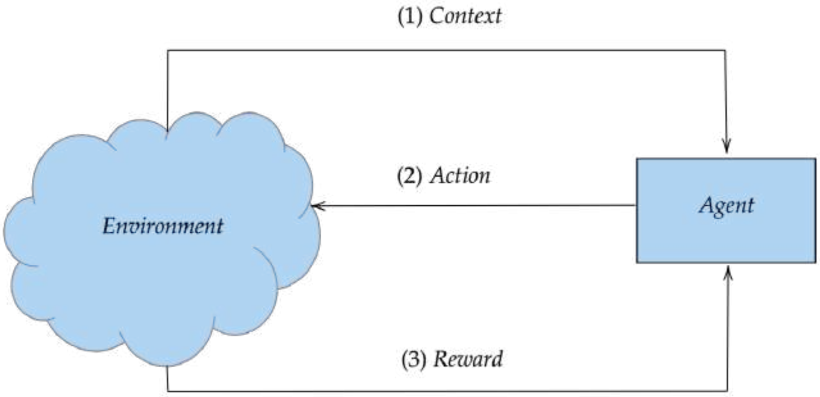


Figure 2.1: A Simple Illustration of the contextual bandit framework. One interaction consists of the environment revealing a context, the agent acting on the context and receiving a reward.

T rounds (when playing actions is costly, think about running Ad campaigns, when one does not own the display space and needs to pay for it) or identify the best arms given a confidence tolerance or a fixed interactions budget (when playing actions has little to no cost, think about a *casino owner* wanting to identify slot machines with high payouts to get rid of them).

Regret Minimisation. The regret of the agent (Auer et al., 2002) is defined as the gap between the highest expected cumulative reward (achieved by an optimal policy) and the cumulative reward the agent actually obtains after T round. Maximising the cumulative reward is equivalent to minimising the regret, the latter quantity however is better suited to theoretically compare the strategies to the best attainable outcome. Figure 2.2 illustrates the regret of a bandit strategy. Algorithms achieving optimal regret need to carefully balance between two conflicting objectives: increase their knowledge by playing new actions (exploration) and leverage the information acquired so far to enhance their performance (exploitation), giving rise to the well known explore-exploit dilemma (Lattimore and Szepesvári, 2020). Optimal strategies for regret minimisation are based on the *optimism in the face of uncertainty* principle, with the most notable strategies being **UCB**: Upper Confidence Bounds (Chu et al., 2011) and **TS**: Thomson Sampling (Agrawal and Goyal, 2013). In each round, these strategies construct (or update) a confidence interval around the true reward[2] and play the action with the highest "potential" outcome. We illustrate in Figure 2.3 a simplified view of the idea behind these algorithms.

Pure Exploration. Pure Exploration is a paradigm used within the contextual bandit framework to identify the best policy under practical constraints. This is suitable for applications where we do not necessarily need to exploit or gather reward to counterbalance the cost of playing actions. The constraints can be split into two types:

- **Fixed Confidence**: Given a tolerance δ, we want to identify the optimal policy with confidence at least $1 - \delta$ while reducing the number of interactions T as much as possible. Some algorithms used for this type of problem are variants of confidence interval strategies (Kalyanakrishnan et al., 2012; Degenne et al., 2019) and Track-and-Stop strategies (Garivier and Kaufmann, 2016).

[2] Thomson Sampling can also be cast within this framework (Abeille and Lazaric, 2017).

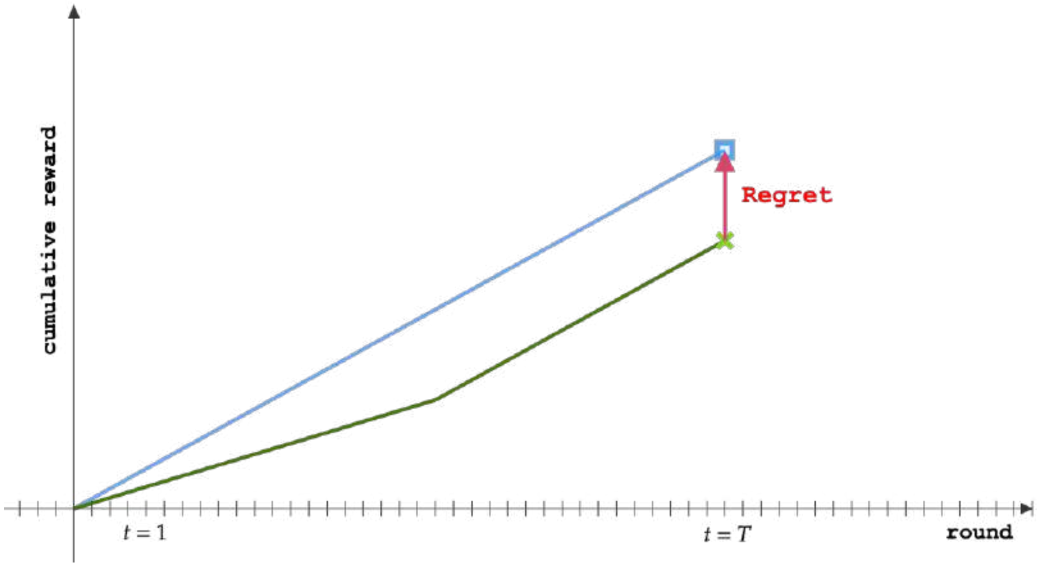


Figure 2.2: A Simple example of the regret of a strategy after T rounds: the difference (red line) between the cumulative reward of the optimal policy (blue curve) and the cumulative reward of our bandit strategy (the green curve). One can observe that starting a certain time, our strategy begins to play optimal actions (both the blue and green line start having the same slope).

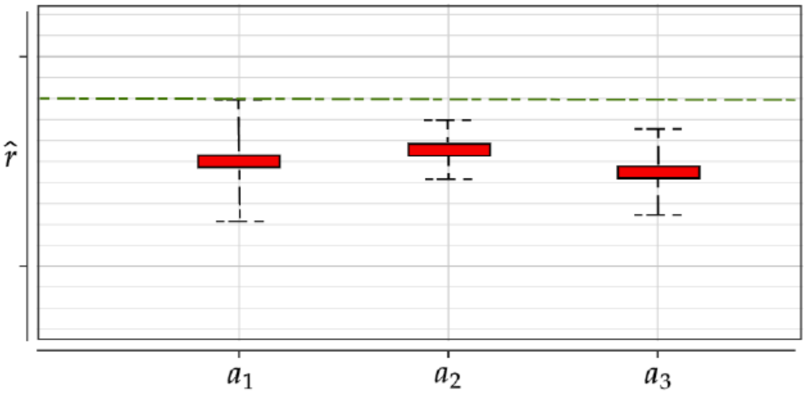


Figure 2.3: An Illustration of the principle of "optimism in face of uncertainty" with an example of a contextual bandit problem with $|\mathcal{A}| = 3$. At round t, we construct a confidence interval around the reward of each arm, and choose the arm with the highest potential payout. In this case, even if a_2 has the highest empirical mean, we choose the arm a_1 as it can have the best reward in the most "optimistic" case.

- **Fixed Interactions budget**: Given a number of interactions T, we want to maximise the probability of returning the optimal policy. One of the algorithms that deal with this type of constraint is the sequential halving algorithm developped in Karnin et al. (2013).

If it is by no mean our ambition to cover the rich literature of the bandit framework in this introduction, the reader can already imagine the endless applications and practical impact this modelling approach might have. All the strategies devised for these different applications benefit from strong theoretical guarantees (achieving low regret, finding optimal policies) while letting the agent learn by its own; besides setting parameters for the strategy, both acting and learning is done online, automatically by the agent. As attractive this online learning setting can be, there are some practical considerations that limit its viability, and motivate us to think about the problem differently:

- **Robust Infrastructure**: Deploying a bandit algorithm online, especially for large scale applications, requires a scalable and robust infrastructure, that is capable of handling hard engineering constraints (asynchronous and automatic updates, monitoring capabilities, etc) requiring a full rethinking of the model deployment pipeline. This can represent a big engineering cost that few companies are willing to pay.

- **Slower experimentation**: The same decision-making problem can be attacked by different bandit strategies, built on different assumptions, while having different hyperparameters to tune. Testing one strategy online requires the deployment of an agent that will learn by interacting with some traffic for enough rounds before convergence. If we can collect n interactions per day, and let us suppose that our bandit strategy need $7n$ interactions to converge, then we can only test out a bandit strategy per week (7 days) which renders experimentation really slow and costly.

- **Might be too costly**: Evaluating a bandit strategy offline before deployment is hard to do, making practitioners deploy agents "blindly'. This can result in unreasonable losses especially in the case of high risk applications. In addition, even if we choose the best suited bandit strategy to our problem, the level of exploration recommended by theory is often costly in the short-mid term. While a good level of exploration is beneficial for the long-term, it can result in immediate loss of revenue that might be detrimental to the business operated as it needs to comply with short-term revenue constraints.

With all these limitations taken into consideration, we want to adapt this framework to better answer the needs of industrial applications. In practice, we usually want to have full control of the amount of exploration done by the systems and prefer being able to manipulate it easily. In addition, businesses rarely face 'cold-start' problems; for the majority of the problems faced, one can leverage expert knowledge combined with non-bandit signal (information about contexts and actions) to design reasonable strategies even before the first interactions with the environment. The main challenge then shifts to the improvement of such strategies with data-driven approaches. It is highly desirable to being able to train the next strategy *offline* (as it reduces drastically the infrastructure prerequisites and accelerates experimentation) while having guarantees on its performance; before deploying the brand-new recommendation engine, one would like to make sure that it will generate at least as much revenue as its predecessor. This requires the development of a counterfactual reasoning, and the construction of specific estimators that allow us to answer the question: "What revenue would I have generated if I had acted differently?". In the hope of answering this question, the offline formulation of the contextual bandit framework was developed with the idea of leveraging logged interactions of an already deployed strategy, to confidently evaluate (What is the revenue generated of a given

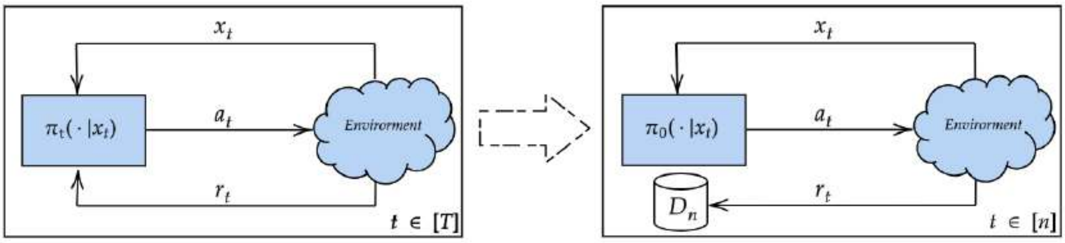


Figure 2.4: The difference between (Online) Contextual bandit and its offline formulation. The online approach (on the left) updates the model every time we observe a reward on an action. The objective is to minimize the regret in the long run. The offline setting (on the right) updates the model once based on the logged interactions of the policy π_0 with the environment. This update is done offline and the new strategy, if better, will be deployed in the future.

policy?) and learn (Find the policy that will maximize revenue) newly constructed strategies *offline*.

2.1.2 The Offline Setting

The offline contextual bandit setting is particularly interesting for industrial applications. It provides more control to practitioners, as they can evaluate and learn new policies, and fully decide on whether to deploy them online or not. In this formulation, the agent gathers data and is not updated after each interaction. Instead, this data is logged and is used by practitioners to design better performing agents for the next deployment. The current agent is represented by the policy π_0 which, in each round $t \in [n]$, acts on the context x_t by performing the action a_t and receives the feedback r_t. Figure 2.4 represents the difference between this formulation and the classical contextual bandit. All the n interactions are logged in the so-called bandit feedback dataset:

$$\mathcal{D}_n = \{x_i \sim \nu, a_i \sim \pi_0(\cdot|x_i), r_i \sim p(\cdot|x_i, a_i), \pi_0(a_i|x_i)\}_{i\in[n]}.$$

The goal in this formulation is often performance driven, as we want to find policies that minimize the risk; defined as the expected negative reward under the actions of the policy. For a given policy π, the risk is expressed as:

$$\begin{aligned} R(\pi) &= -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\mathbb{E}_{r\sim p(\cdot|x,a)}[r]\right] \\ &= -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right] \\ &= \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[c(a,x)\right]. \end{aligned}$$

with the cost $c(a,x)$ defined as $-\bar{r}(a,x)$. These notations produce the same definition and will be used exchangeably in the rest of the manuscript. As we cannot have access to the true expected risk, we proceed by building an estimator of this quantity to first evaluate the risk of any policy offline and learn reward maximizing policies in a second time.

Policy Evaluation. We want to be able to evaluate the performance of any given policy π and be able to compare it to the performance of the policy acting in production. The most reliable way to achieve this is to actually deploy the policy π and gather interactions under it to estimate its expected risk. Modern online decision systems rely on A/B tests (Kohavi et al., 2012), considered the "gold-standard" of evaluation practices. When conducted properly (Gupta et al., 2019), an A/B test can accurately estimate the effect of replacing the current policy “A” with the new candidate “B”. The common protocol begins by choosing a promising policy with the help of extensive offline experiments. The new policy is then deployed, alongside the current system and the A/B test is conducted to decide, whether the chosen candidate “B” improves and should replace the current system “A”. In large scale production systems, A/B tests require substantial engineering effort, constant monitoring and need several days to be properly analysed. Ideally, the offline selection process should produce excellent candidates that align with the online metrics, to avoid unnecessary A/B tests. Aligning offline and online performance is the goal of the research literature on policy evaluation. The challenge that arises from this approach is that we can only use data collected under the policy π_0 to evaluate, any, possibly different policy π. A common idea is to correct the bias of the estimation of the risk of new policies π with importance weighting (Chopin and Papaspiliopoulos, 2020), as we have:

$$\begin{aligned} R(\pi) &= -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right] \\ &= -\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x)}\left[\frac{\pi(a|x)}{\pi_0(a|x)}\bar{r}(a,x)\right], \end{aligned}$$

The expectation becomes computed under π_0 and thus can be approximated by the collected interactions, giving the well known IPS: Inverse Propensity Scoring estimator (Horvitz and Thompson, 1952) as a result:

$$\hat{R}_n^{\texttt{IPS}}(\pi) = -\frac{1}{n}\sum_{i=1}^{n}\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}r_i.$$

This estimator of the risk of π is unbiased when the support[3] of π is included in the support of π_0. This is a desirable property as it means that the estimator is easy to analyse, consistent and will converge to the true risk with enough samples. However, as this estimator relies on importance weighting, its variance depends on the disparity between the policy that we want to evaluate and the policy that gathered the data (Bottou et al., 2013), its use can be problematic when the new policy π differs drastically from π_0. In these cases, one would prefer an estimator that do not suffer from large variance problems. A common way to achieve this is to learn a model $r_\mathcal{M} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^+$ of the reward mean $\bar{r}$. Once we have a model $r_\mathcal{M}$, we can build a simple estimator of the risk of any policy from the following observation:

$$\begin{aligned} R(\pi) &= -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right] \\ &\approx -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[r_\mathcal{M}(a,x)\right]. \end{aligned}$$

This produces the DM: Direct Method estimator that writes:

$$\hat{R}_n^{\texttt{DM}}(\pi) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\pi(a|x_i)r_\mathcal{M}(a,x_i).$$

The DM estimator does not suffer from variance problems coming from the mismatch of both policies as it does not rely on importance weighting. It can evaluate any policy π, even if π and π_0 do not share the same support. The efficiency of this estimator however depends entirely on our ability to model the problem. If this estimator enjoys a well-behaved variance, its limitation comes from a potentially substantial bias, as it is generally hard to model the reward perfectly. As both estimators have complementary properties, we can mitigate their limitations by combining them. The DR: Doubly Robust estimator (Dudík et al., 2014) does

[3] $\text{supp}(\pi) = \{(x,a) \in \mathcal{X}\times\mathcal{A}, \pi(a|x) > 0\}$

that and results in an improved estimator. The idea behind the construction of this estimator stems from the following identity:

$$\begin{aligned} R(\pi) &= -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right] \\ &= -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x) - r_{\mathcal{M}}(a,x)\right] - \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[r_{\mathcal{M}}(a,x)\right] \\ &= -\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x)}\left[\frac{\pi(a|x)}{\pi_0(a|x)}\left(\bar{r}(a,x) - r_{\mathcal{M}}(a,x)\right)\right] - \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[r_{\mathcal{M}}(a,x)\right]. \end{aligned}$$

Which combines both the importance weighting technique and the use of a reward model $r_{\mathcal{M}}$, resulting in the DR estimator:

$$\hat{R}_n^{\texttt{DR}}(\pi) = -\frac{1}{n}\sum_{i=1}^{n}\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}\left(r_i - r_{\mathcal{M}}(a_i,x_i)\right) - \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\pi(a|x_i)r_{\mathcal{M}}(a,x_i).$$

The estimator obtained is unbiased under the same common support condition and enjoys a better variance (Nguyen-Tang et al., 2022). Research in the area of offline (also called off-policy) evaluation focuses on deriving estimators with an improved bias-variance trade-off, either by using different importance weighting techniques (Ionides, 2008; Swaminathan and Joachims, 2015b; Wang et al., 2017; Metelli et al., 2021) or by assuming a certain structure on the reward (Swaminathan et al., 2017; Saito and Joachims, 2022a; Saito et al., 2023). Building these estimators help us evaluate the performance of any policy π, thus they can be used as a training objective to find the best policy π offline.

Policy Learning. The ingredients to learn a policy are to choose an objective function; often a *regularized* off-policy estimator (Swaminathan and Joachims, 2015a; Ahmed et al., 2019; London and Sandler, 2019), and a policy class on which to optimize it. These choices dictate the approach that will be adopted and often result in different policy learning algorithms. Let $\Pi = \{\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})\}$ be the space of policies, and let us begin by introducing one of the simplest approaches. If we are confident about our ability to model the problem, and have built a reward model $r_{\mathcal{M}}$, we can proceed and learn a policy using the Direct Method. The idea stems from the following:

$$\begin{aligned} \arg\min_{\pi\in\Pi} R(\pi) &= \arg\min_{\pi\in\Pi} -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\bar{r}(a,x)\right] \\ &\approx \arg\min_{\pi\in\Pi} -\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[r_{\mathcal{M}}(a,x)\right]. \end{aligned}$$

By replacing the unknown mean reward $\bar{r}$ by our model $r_{\mathcal{M}}$, we can solve the unconstrained policy optimization problem and obtain the DM solution:

$$\forall (x,a) \quad \pi_{\texttt{DM}}(a|x) = \mathbb{1}\left[\arg\max_{a'\in\mathcal{A}} r_{\mathcal{M}}(a',x) = a\right].$$

For each context x, the DM policy chooses the action a that has the biggest reward according to our model $r_{\mathcal{M}}$. This approach is called the Direct Method because we can directly derive the optimal policy from the reward model. Sometimes, we want to enforce some constraint on the policy deployed. For example, some applications require policies that diversify the actions played for the same context, others need some exploration to better identify the best actions. This constraint is often encoded by adding a regularization to the learning objective. To achieve better diversification, we add an entropy regularization (Ahmed et al., 2019) and modify our optimization problem to solve the following:

$$\arg\min_{\pi\in\Pi}\left\{R(\pi) + \gamma\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[\log\pi(a|x)\right]\right\} \approx \arg\min_{\pi\in\Pi}\left\{\mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[-r_{\mathcal{M}}(a,x) + \gamma\log\pi(a|x)\right]\right\}$$

with γ a positive parameter that controls the diversity level of the policy. The solution of this optimization problem can be obtained analytically and is expressed as:

$$\begin{aligned}\forall(x,a) \quad \pi^{\gamma}_{\texttt{DM}}(a|x) &= \texttt{softmax}_{\mathcal{A}}\left(r_{\mathcal{M}}(a,x)/\gamma\right) \\ &= \frac{\exp(r_{\mathcal{M}}(x,a)/\gamma)}{\sum_{a'\in\mathcal{A}}\exp(r_{\mathcal{M}}(x,a')/\gamma)}.\end{aligned}$$

This policy has a positive probability mass on all actions, interpolating between a uniform distribution ($\gamma \to +\infty$) and $\pi_{\texttt{DM}}$ ($\gamma \to 0$). The policies derived with the direct method depend on the reward model, directing all our efforts towards building a $r_{\mathcal{M}}$ that reflects the properties of the true rewards and from which the policy derived fits our engineering constraints. Sometimes, our reward model $r_{\mathcal{M}}$ produces an optimal policy $\pi_{\texttt{DM}}$ that cannot be deployed due to application-dependent constraints (in low latency applications, finding the action with maximum reward for a particular context x can take more time than allowed). In these cases, we restrict our optimization problem to a space of policies that fits the requirements of our problem. Building a space of policies is usually done through the definition of:

- A *parametric* space of score functions $\mathcal{F}(\Theta) = \{f_\theta : \mathcal{X}\times\mathcal{A}\to\mathbb{R}, \theta\in\Theta\subset\mathbb{R}^d\}$ with d the dimension of the parameters. Given a $\theta\in\Theta$ and for a particular context x and action a, the value of $f_\theta(x,a)$ reflects the relevance of action a to the context x.

- A link function L that takes a score function f_θ and transforms it in order to define a policy π_θ. If we want to write:

  $$\forall(x,a), \quad \pi_\theta(a|x) = L(f_\theta(a,x)).$$

  L needs to be a positive, real valued function $L : \mathbb{R} \to \mathbb{R}^+$ that verifies the following condition:

  $$\forall(\theta,x), \quad \sum_{a'\in\mathcal{A}} L(f_\theta(a',x)) = 1.$$

  The space of functions verifying these conditions will be denoted by $\mathcal{L}$. Different link functions produce policies with different properties (Mei et al., 2020a,b; Sakhi et al., 2023a). We already saw from the DM example that the link function defining our policy can be an indicator or a softmax function depending on the objective we aim for. In general, we want smooth link functions that facilitate optimization making the softmax function (Mei et al., 2020b) a commonly adopted option.

The choice of the couple $(\mathcal{F}(\Theta), L\in\mathcal{L})$ is enough to define a parametric policy space $\Pi(\Theta)$ on which the optimization of our objective function can be done. Getting back to the Direct Method approach, we shift our focus to solving the following constrained optimization problem:

$$\begin{aligned}\operatorname*{arg\,min}_{\pi_\theta\in\Pi(\Theta)} R(\pi_\theta) &= \operatorname*{arg\,min}_{\pi_\theta\in\Pi(\Theta)} -\mathbb{E}_{x\sim\nu, a\sim\pi_\theta(\cdot|x)}\left[\bar{r}(a,x)\right] \\ &\approx \operatorname*{arg\,min}_{\pi_\theta\in\Pi(\Theta)} -\mathbb{E}_{x\sim\nu, a\sim\pi_\theta(\cdot|x)}\left[r_{\mathcal{M}}(a,x)\right].\end{aligned}$$

As we do not know if $\pi_{\texttt{DM}}\in\Pi(\Theta)$, we proceed by computationally solving the empirical counterpart of the objective:

$$\operatorname*{arg\,min}_{\pi_\theta\in\Pi(\Theta)}\left\{-\frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\pi_\theta(a|x_i)r_{\mathcal{M}}(a,x_i)\right\} = \operatorname*{arg\,min}_{\pi_\theta\in\Pi(\Theta)}\left\{\hat{R}^{\texttt{DM}}_n(\pi_\theta)\right\}.$$

Which can be interpreted as distilling the potentially complicated reward model $r_{\mathcal{M}}$ into a policy that fits our constraints. In this example, our learning objective was the `DM` estimator. As a general rule, off-policy learning objectives rely on optimizing a regularized risk estimator on a parametric policy class:

$$\operatorname*{arg\,min}_{\pi_\theta \in \Pi(\Theta)} \left\{ \hat{R}_n(\pi_\theta) + \lambda C(\pi_\theta) \right\}, \tag{2.1}$$

with $\hat{R}_n$ a risk estimator, λ a tunable parameter and $C(\pi_\theta)$ a regularization term that is either motivated by additional constraints we want to enforce (Ahmed et al., 2019; Schulman et al., 2015) or statistical learning theory arguments making the learning of these policies more principled (Swaminathan and Joachims, 2015a; Ma et al., 2019; London and Sandler, 2019). In the next section, we will develop the policy learning discussion more, with a particular focus on statistical learning tools that enable us to learn systems with performance certificates.

2.2 Performance Guarantees with Statistical Learning

Statistical learning theory (Vapnik, 1998) studies the problem of inference; that is, of gaining knowledge and making predictions based on a set of data. In particular, we are interested in the **PAC**: Probably Approximately Correct framework (Valiant, 1984), a branch of learning theory that investigates the problem of generalisation, answering the question of how well a predictor (or a family of predictors) can perform on unseen data. Developments of this branch improved our understanding of common learning paradigms, with contributions in supervised learning (Vapnik, 1991; Cortes and Vapnik, 1995; McAllester, 1998; Catoni, 2007; Germain et al., 2009), unsupervised learning (Bengio et al., 2013; Saunshi et al., 2019; Nozawa et al., 2020) and online learning (Even-Dar et al., 2002; Seldin et al., 2011; Haddouche and Guedj, 2022; Tirinzoni et al., 2023; Al-Marjani et al., 2023). Historically, supervised learning had attracted most attention and is best understood from this perspective. It is only natural to choose this learning paradigm to present some of the tools used by the PAC framework. In this setting, we are given a data set, and a loss to measure performance. We fix a set of predictors and look for a good predictor in this set, w.r.t to the loss defined. Formally, we have:

- A dataset $S_n = \{X_i \in \mathcal{X}, Y_i \in \mathcal{Y}\}_{i \in [n]}$ composed of n i.i.d. observations coming from an unknown joint distribution $p(\mathcal{X}, \mathcal{Y})$. $\mathcal{X}$ is the object set (text, image) and $\mathcal{Y}$ the label set (sentiment of the text, class of the image).

- A loss function $l : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ measuring the quality of the predictions, with the convention that $l(y, y) = 0$.

- We look for good predictors in $\mathcal{H}_\Theta = \{h_\theta : \mathcal{X} \to \mathcal{Y}, \theta \in \Theta\}$ a class of predictors, parameterized by θ coming from the parameter set Θ.

- We are interested in finding a predictor h from $\mathcal{H}_\Theta$ that will minimize the expected loss $l(h) = \mathbb{E}_{(X,Y)\sim\nu}[l(h(X), Y)]$. We denote by $l_n(h)$ the empirical loss estimate.

As it is usually impossible to have access to the true, expected loss, the PAC toolbox provide us with bounds that control this quantity for any predictor $h_\theta \in \mathcal{H}_\Theta$. PAC bounds give us the following result, holding with high probability over the data:

$$\forall \theta \in \Theta : l(h_\theta) \leq l_n(h_\theta) + \mathcal{O}\left(C_n(\mathcal{H}_\Theta)\right),$$

with $C_n(\mathcal{H}_\Theta)$ a measure of complexity (Vapnik, 1998; Zhou, 2002; Bartlett and Mendelson, 2003) of the class of predictors used. In our applications, we want to obtain a performance guarantee on our predictors with the help of these bounds. For a predictor $h_{\hat{\theta}}$, we want to control its true expected loss with high probability. We aim at obtaining a *performance guarantee/certificate*, which is a result of the form:

$$l(h_{\hat{\theta}}) \leq 0.12. \tag{2.2}$$

This result **guarantees us** (with high probability) that our predictor, will suffer a loss of at most 0.12. To obtain the smallest guarantees, we seek bounds that are tight and advocate for minimizing the right-hand side over all $\theta \in \Theta$ in order to control and minimize expected loss. For example, Maurer and Pontil (2009) derived an empirical Bernstein-type bound, and used the notion of covering number (Zhou, 2002) to control the loss of a class of predictors. For a tolerance $\delta \in]0,1]$, Their main result is a bound holding with probability $1-\delta$:

$$\forall \theta \in \Theta : l(h_\theta) \leq l_n(h_\theta) + \sqrt{\frac{18 v_n(h_\theta) \ln\left(\mathcal{M}_n(\mathcal{H}_\Theta)/\delta\right)}{n}} + \frac{15 \ln\left(\mathcal{M}_n(\mathcal{H}_\Theta)/\delta\right)}{n-1}, \tag{2.3}$$

with $v_n(h_\theta)$ the empirical variance of the loss estimate and $\mathcal{M}_n(\mathcal{H}_\Theta)$ a complexity measure defined in (Maurer and Pontil, 2009, Theorem 6). This complexity is intractable even for simple predictor classes, which means that its presence makes the bound unusable as-is for learning purposes. $\mathcal{M}_n(\mathcal{H}_\Theta)$ can be upper bounded by empirical quantities (Zhang, 2002) but this often results in loose and overly conservative bounds. In particular, this complexity is known to be very large for rich predictor classes (i.e. neural networks), making the bound vacuous. To circumvent this limitation, the usual approach is to identify useful quantities from the bound and propose a learning principle, replacing intractable quantities with tunable hyperparametes. This approach motivated numerous learning principles, such as Empirical Risk Minimization (Vapnik, 1991) and Structural Risk Minimization (Cortes and Vapnik, 1995). In this example, the **SVP** principle was derived from Equation (2.3) proposing to solve the following optimization problem:

$$\underset{\theta \in \Theta}{\arg\min} \left\{ l_n(h_\theta) + \lambda \sqrt{\frac{v_n(h_\theta)}{n}} \right\},$$

with λ a hyperparameter selected with cross-validation. Using these learning principles provide practitioners with tractable optimization objectives, but does not result in performance certificate like in Equation (2.2). Swaminathan and Joachims (2015a) adapted these results to the offline contextual bandit framework. See (Swaminathan and Joachims, 2015a, Table 1) for the differences between the supervised learning problem and the offline contextual bandit problem. Particularly, they based their analysis on `cIPS`: clipped IPS (Bottou et al., 2013) in order to respect the bounded assumption of the loss. This risk estimate was used to derive a bound similar to Equation (2.3) holding for policies π_θ in a policy class $\Pi(\Theta)$. The obtained bound (Swaminathan and Joachims, 2015a, Theorem 1) is intractable, and motivated the use of a similar learning principle.

The Distributionally Robust Optimization framework (Duchi et al., 2021) provides an intuitive approach to control the loss of our predictors. After observing the samples S_n, it treats the induced empirical distribution with scepticism and seeks a solution that minimizes the worst-case expected cost over a family of distributions, described in terms of an uncertainty ball (around the observed, empirical distribution). These tools were proven to be powerful for decision theory (Duchi and Namkoong, 2019) and in training robust classifiers (Madry et al., 2018). Let $\mathcal{U}_\epsilon(\hat{p}_n)$ be the uncertainty ball of radius ϵ, around the empirical distribution $\hat{p}_n$, and let h_θ be a predictor from $\mathcal{H}_\Theta$. Instead of studying the empirical loss $l_n(h_\theta)$, the DRO formulation focuses on the following, worst-case empirical estimator (Duchi et al., 2021):

$$l_n(h_\theta, \mathcal{U}_\epsilon(\hat{p}_n)) = \max_{q \in \mathcal{U}_\epsilon(\hat{p}_n)} \mathbb{E}_{(x,y)\sim q} \left[l(h_\theta(x), y) \right].$$

This framework is also called generalized, empirical likelihood as a well-chosen uncertainty ball recovers the empirical likelihood approach of Owen (2001). For a particular choice of the uncertainty set $\mathcal{U}_\epsilon^{\chi^2}(\hat{p}_n)$ (using the χ^2 divergence to quantify the distance from the empirical distribution), Duchi and Namkoong (2019) prove that the DRO, worst-case empirical estimator is equivalent to a *variance-regularized* empirical loss:

$$l_n(h_\theta, \mathcal{U}_\epsilon^{\chi^2}(\hat{p}_n)) = l_n(h_\theta) + \sqrt{\epsilon v_n(h_\theta)}.$$

This result means that minimizing the worst-case empirical loss is equivalent to solving the **SVP** principle. These tools were adapted to the problem of off-policy learning (Faury et al., 2020; Dai et al., 2020) and we develop them further in Chapter 3. Their use is motivated by asymptotic-coverage arguments (in the limit, the worst-case risk will cover the true risk) and their finite-sample analysis is loose, failing to produce satisfying performance guarantees (a result similar to Equation (2.2)) in practical scenarios (Dai et al., 2020).

If our objective is to know how a policy will perform before it interacts with the environment, deriving a learning principle is not enough. We are interested in obtaining **performance guarantees**; results similar to Equation (2.2), where we control with high confidence the risk of a trained policy $\pi_{\hat{\theta}}$:

$$R(\pi_{\hat{\theta}}) \leq -0.81. \quad \text{(The risk is in } [-1, 0]) \tag{2.4}$$

This result certifies that, in the worst case, our policy $\pi_{\hat{\theta}}$ will have a risk of -0.81. This performance guarantee give practitioners a way to identify promising policies that are worth A/B testing. For this same example, if the logging policy π_0 have a risk of -0.71, then Equation (2.4) alone guarantee us that the new learned policy improves on π_0. These results are desired in the offline policy learning context and can have a substantial impact on real world problems. Obtaining these results rely on the derivation of **tight** and **tractable** PAC bounds. Recently, PAC-Bayes bounds (McAllester, 1998; Catoni, 2007), a family of PAC bounds, promise the delivery of performance guarantees for difficult problems (Dziugaite and Roy, 2017) and present themselves as good candidates to answer this question. If the notion of complexity in PAC bounds limited their application to simple predictor classes, PAC-Bayes techniques can deal elegantly with any predictor class $\mathcal{H}_\Theta$ and are proven to provide performance guarantees even for well-known, over-parameterized neural networks (Dziugaite and Roy, 2017). However, an artefact of these bounds is that we need to change the quantities of interest. PAC bounds study the performance of a predictor h_θ in $\mathcal{H}_\Theta$ by controlling its loss $l(h_\theta)$. PAC-Bayes bounds however study randomized predictors; obtained by sampling in a set of basic predictors, according to some prescribed probability distribution. Formally, let $\mathcal{P}(\Theta)$ be the set of all probability distributions on Θ (equipped with its σ-algebra). let $Q \in \mathcal{P}(\Theta)$ a probability distribution over Θ (and thus $\mathcal{H}_\Theta$), PAC-Bayes bounds control the loss of randomized predictors, computed as :

$$\mathbb{E}_{\theta\sim Q}\left[l(h_\theta)\right].$$

In a supervised learning setting, this quantity can be interpreted as adopting the following procedure: for each sample $(X, Y) \sim \nu(\mathcal{X}, \mathcal{Y})$, we sample a predictor h_θ from Q, predict the label $Y^p = h_\theta(X)$ and then compute the loss $l(Y^p, Y)$. This is different from studying aggregated predictors (Breiman, 2001) where for each sample, we aggregate (by either voting or averaging) all predictor's results to predict the label. We present the differences introduced with the PAC-Bayesian approach for supervised learning. We have:

- A dataset $S_n = \{X_i \in \mathcal{X}, Y_i \in \mathcal{Y}\}_{i\in[n]}$ composed of n i.i.d. observations coming from an unknown joint distribution $p(\mathcal{X}, \mathcal{Y})$. $\mathcal{X}$ is the object set (text, image) and $\mathcal{Y}$ the label set (sentiment of the text, class of the image).
- A loss function $l : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ measuring the quality of the predictions, with the convention that $l(y, y) = 0$.
- We define $\mathcal{H}_\Theta = \{h_\theta : \mathcal{X} \to \mathcal{Y}, \theta \in \Theta\}$ a class of predictors, parameterized by θ coming from the parameter set Θ.
- **(PAC-Bayes)** We define a a set of probability distribution $\mathcal{M}(\Theta) \subseteq \mathcal{P}(\Theta)$ over Θ.
- **(PAC-Bayes)** We are interested in finding a good distribution $Q \in \mathcal{M}(\Theta)$ that will minimize the expected loss of the randomized predictor $\mathbb{E}_{\theta\sim Q}\left[l(h_\theta)\right]$.

This is achieved through the derivation of bounds holding for all distributions $Q \in \mathcal{M}(\Theta)$. Let $P \in \mathcal{P}(\Theta)$ a reference distribution that does not depend on the data S_n. The general form of PAC-Bayesian bounds is an inequality holding with high probability:

$$\boxed{\forall Q \in \mathcal{M}(\Theta) : \mathbb{E}_{\theta\sim Q}\left[l(h_\theta)\right] \leq \mathbb{E}_{\theta\sim Q}\left[l_n(h_\theta)\right] + \mathcal{O}\left(\mathcal{KL}\left(Q||P\right)\right),}$$

with $\mathcal{KL}$ the KL-divergence defined by:

$$\mathcal{KL}(Q||P) = \begin{cases} \int \ln\left\{\frac{dQ}{dP}\right\} dQ & \text{if } Q \text{ is } P\text{-continuous,} \\ +\infty & \text{otherwise.} \end{cases}$$

For example, we give below a simple PAC-Bayes bound (Catoni, 2007) to showcase the versatility of this framework. Let $P \in \mathcal{P}(\Theta)$ a reference distribution, $\delta \in]0, 1]$ a tolerance and $\lambda > 0$, we have with probability at least $1 - \delta$:

$$\boxed{\forall Q \in \mathcal{P}(\Theta) : \mathbb{E}_{\theta\sim Q}\left[l(h_\theta)\right] \leq \mathbb{E}_{\theta\sim Q}\left[l_n(h_\theta)\right] + \frac{\mathcal{KL}\left(Q||P\right) + \log 1/\delta}{\lambda} + \frac{\lambda}{8n} \qquad (2.5)}$$

Minimizing the r.h.s gives the smallest guarantees on aggregated predictors. We get to solve the following optimization problem:

$$\underset{Q\in\mathcal{P}(\Theta)}{\arg\min} \quad \mathbb{E}_{\theta\sim Q}\left[l_n(h_\theta)\right] + \frac{\mathcal{KL}\left(Q||P\right)}{\lambda},$$

which happens to be analytically tractable and we obtain the Gibbs distribution (Guedj, 2019) as a solution:

$$\forall \theta \in \Theta, \quad d\hat{Q}(\theta) \propto \exp\left(-\lambda l_n(h_\theta)\right) \times dP(\theta).$$

The bound also gives us an idea on the worst case performance of the randomized predictor according to $\hat{Q}$:

$$\mathbb{E}_{\theta\sim\hat{Q}}\left[l(h_\theta)\right] \leq \log\left(\mathbb{E}_{\theta\sim P}\left[\exp\left(-\lambda l_n(h_\theta)\right)\right]\right) + \frac{\log 1/\delta}{\lambda} + \frac{\lambda}{8n}. \tag{2.6}$$

Sampling from the distribution $\hat{Q}$ is mandatory if we want to use the randomized predictor, and computing $L(P) = \log\left(\mathbb{E}_{\theta\sim P}\left[\exp\left(-\lambda l_n(h_\theta)\right)\right]\right)$ is desired to have an idea on its performance. Both sampling from $\hat{Q}$ and approximating $L(P)$ can be done with (Markov Chain/Sequential) Monte Carlo (Chopin and Papaspiliopoulos, 2020). If these methods fail to scale to our problem, a usual solution is to restrict the probability set $\mathcal{M}(\Theta)$ to simple distributions. If $\Theta = \mathbb{R}^d$, a common choice is to set $\mathcal{M}(\Theta) = \left\{\mu \in \mathbb{R}^d, \mathcal{N}(\mu, I_d)\right\}$ to unit-variance, isotropic Gaussians. By fixing the reference distirbution to $P = \mathcal{N}(\mu_0, I_d)$, the previous bound becomes:

$$\forall \mu \in \mathbb{R}^d : \mathbb{E}_{\theta\sim\mathcal{N}(\mu,I_d)}\left[l(h_\theta)\right] \leq \mathbb{E}_{\theta\sim\mathcal{N}(\mu,I_d)}\left[l_n(h_\theta)\right] + \frac{||\mu-\mu_0||^2 + 2\log 1/\delta}{2\lambda} + \frac{\lambda}{8n}. \tag{2.7}$$

Obtaining a good randomized predictor boils down to computationally solving the optimization problem:

$$\underset{\mu\in\mathbb{R}^d}{\arg\min}\, \mathbb{E}_{\theta\sim\mathcal{N}(\mu,I_d)}\left[l_n(h_\theta)\right] + \frac{||\mu-\mu_0||^2}{2\lambda}.$$

This optimization problem looks like Variational Bayes objectives (Blei et al., 2017) for which a multitude of solutions were proposed to solve it efficiently (Xu et al., 2019). Once we have our solution, obtaining the worst case loss for the randomized predictor can be done by evaluating the bound. We can observe that, contrary to classical PAC bounds, PAC-Bayesian bounds are tractable and benefit from various computational tools to find their minimizers. They are also tight enough to be valuable in learning both simple (Germain et al., 2009) and complex predictors (Dziugaite and Roy, 2017) with guarantees. To increase the impact of these bounds, research in this area is focused on deriving new, tighter bounds (Mhammedi et al., 2019; Jang et al., 2023), loosening assumptions (Alquier and Guedj, 2018; Kuzborskij and Szepesvári, 2020; Haddouche and Guedj, 2023) and adapting them to various learning problems (Seldin et al., 2011; London and Sandler, 2019; Haddouche and Guedj, 2022). To make them even more viable, new disintegration techniques (Viallard et al., 2023) were also developed to allow these bounds to give strong performance guarantees on single predictors drawn from the learned distribution. If working with randomized predictors can be seen as a "bug" in most settings, it is considered a "feature" in policy optimization as both policies and randomized predictors are closely related.

- The procedure of a randomized predictor is the following: for each sample $(X, Y) \sim \nu(\mathcal{X}, \mathcal{Y})$, we sample a predictor h_θ from Q, predict the label $Y^p = h_\theta(X)$ and then suffer the loss $l(Y^p, Y)$.

- The procedure of a policy is the following: for each context $x \sim \nu(\mathcal{X})$, we sample an action a from $\pi(\cdot|x)$, and receive the reward $r \sim p(\cdot|x, a)$.

The procedures are similar and both objects can be related if we work with class of predictors that map contexts x to actions in $\mathcal{A}$. Indeed, instead of sampling directly from a distribution on the action set, we sample from a distribution on a predictor space, $\mathcal{H}_\Theta \subseteq \{h : \mathcal{X} \to \mathcal{A}\}$. As such, for a distribution, Q over $\mathcal{H}_\Theta$, the probability of an action, $a \in \mathcal{A}$, given a context, $x \in \mathcal{X}$, is the probability that a random predictor, $h_\theta \sim Q$, maps x to a; that is:

$$\pi_Q(a|x) = \mathbb{E}_{h\sim Q}\left[\mathbb{1}[h(x) = a]\right].$$

This result shows that a policy is a randomized predictor in disguise. This perspective was developed in Seldin et al. (2012), adopted by London and Sandler (2019) and later formalized in Sakhi et al. (2023a). This result is key to the analysis of London and Sandler (2019), that adapted McAllester (2003)'s bound to the offline contextual bandit setting. To achieve this, they clipped the propensity score and used the following risk estimator:

$$\hat{R}^\tau_n(\pi) = -\frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\max\{\pi_0(a_i|x_i), \tau\}} r_i,$$

with $\tau \in]0, 1]$. Their analysis resulted in the bound below. Given a reference distribution P and a tolerance parameter $\delta \in]0, 1]$, the following holds with probability at least $1 - \delta$, uniformly over all distributions $Q \in \mathcal{P}(\Theta)$:

$$\boxed{R(\pi_Q) \le \hat{R}^\tau_n(\pi_Q) + \frac{2(\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{\tau n} + \sqrt{\frac{2[\hat{R}^\tau_n(\pi_Q) + \frac{1}{\tau}](\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{\tau n}}.}$$

One can see that for offline contextual bandits, PAC-Bayes bounds control the quantity of interest, which is the risk of the policy directly. Working with randomized predictors for this problem matches perfectly our needs. Another connection between the PAC-Bayes framework and offline contextual bandits is that the reference distribution can be set naturally to match the logging policy π_0. Indeed, P can be chosen such as $\pi_0 = \pi_P$ to obtain a bound that encourages policies with low empirical risk that stay close to the logging policy π_0. All of these connections make PAC-Bayes the perfect candidate for guaranteed performance. The bound proposed in London and Sandler (2019) however, is not tight enough and produces vacuous results in practical scenarios (Sakhi et al., 2023a). London and Sandler (2019) avoided using the bound and derived a learning principle for parametrized softmax policies. This principle advocates for a L_2 regularization towards the parameter of π_0:

$$\arg\min_{\theta\in\Theta}\left\{\hat{R}^\tau_n(\pi_\theta) + \lambda||\theta - \theta_0||^2\right\}.$$

If this principle improves on **CRM** (Swaminathan and Joachims, 2015a), these results are far from being satisfying if we want to have guarantees on the learned policies. To this end, we continue the development of PAC-Bayes bounds for this problem in Chapters 4 and 5 to finally obtain tight bounds, that certify the performance of the new policies and can confidently improve on the logging policy π_0. These results are desired in production settings where we would like to propose a new system that will improve on the current production system with high probability. This is the case of online decision systems, in particular, recommender systems, where our goal is to always improve the quality of recommendation to better answer the needs of the users. In the next section, we cover the history of recommendation and present how the offline contextual bandit tools fit in the picture, playing a crucial role in redefining the modern internet experience.

2.3 Online Decision Systems: History of Recommendation

Online decision systems have revolutionized the way we interact with the vast ocean of content present on the internet. From search engines to recommender systems, they offer a personalized experience by efficiently exploring the overwhelming amount of information and filtering it to cater to the specific needs of the users. Although these systems are now ubiquitous, it was not always the case during the emergence of the internet. Democratizing the access to web-based information resulted in an exponential increase in the quantity of available data. This increase alone did not upgrade the internet experience, as having access to non structured, vast amount of information is not beneficial unless we have tools to efficiently explore it. This issue attracted research interest which gave birth to the field of IR: Information Retrieval (Rijsbergen, 1979). A natural application of IR is web search engines, now considered an integral part of the internet experience. In their simplest form, these engines take in queries like “Is it normal to be depressed during COVID?” and produce an ordering of, hopefully relevant web pages as a result. If we are more ambitious, we would like to know what happens when our query is incomplete as we need implicit information to better answer it? What happens when we do not have an explicit query at all? What happens when we do not know which musical artist can be interesting to listen to or which movie we would like to watch? In such scenarios, the field of Recommender Systems comes into play, providing a needed solution to these challenges. The concept of filtering and recommending information to users has been around for some time, with early examples dating back to the 1990s. Belkin and Croft (1992) analyzed the two notions of Information Filtering and Information Retrieval, arguing that the latter constitutes the fundamental technology behind Search Engines, while Recommender Systems are built with ideas rooted in the former. In the same year, Goldberg et al. (1992) proposed the "Tapestry" system allowing users, through a graphical interface, to explicitly rate items and view recommendations based on their preferences and the ratings of other users with similar tastes. The term "Collaborative Filtering" first appeared in this work to denote that the information extracted from other users preferences, combined with your preferences (explaining the collaborative part) would be used to infer what the system should recommend (explaining the filtering part) to you. During the same period, content-based filtering also emerged, where recommendations are made based on item features or attributes. Despite its simplicity, creating an operational content-based recommender system, even for basic applications, was a significant challenge, as it required a deep (not in a machine learning sense) understanding of the topic under consideration and the factors influencing the relationship between users and the topics themselves. While modern machine learning tools, emerging from the combination of accurate modelling and powerful computations (Blei et al., 2003; Vaswani et al., 2017b), can now extract valuable factors from the content being recommended, this was not the case in the 1990s. One of the earliest successful real-world projects in this area was the Music Genome Project, which aimed to capture the essence of music through its properties. This project represents any song with over 450 properties and describes the interplay between each one of them. Once we obtain the song's representations, the recommendation procedure follows a natural design. When a user likes a song, the procedure attributes positive values to its specific properties, promoting similar songs (with similar properties) and bringing them to the user's attention. Collaborative filtering and content-based filtering are built on distinct principles, each with its own strengths and weaknesses. Content-based filtering relies on a comprehensive understanding of the recommended content, and therefore does not necessarily require input from other users. In contrast, collaborative filtering depends heavily on user interactions to identify individuals with similar preferences. Content-based filtering may, however, have limitations when it comes to generating novel or diverse recommendations, since it is primarily based on an understanding of the properties of the content. Fortunately, both

$$R = \underbrace{\left.\begin{bmatrix} ? & \mathbf{1} & \dots & \mathbf{5} & ? \\ \mathbf{4} & \mathbf{2} & \dots & ? & ? \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{5} & ? & \dots & \mathbf{4} & ? \\ ? & ? & \dots & \mathbf{5} & \mathbf{2} \end{bmatrix}\right\} U}_{I} \implies \hat{f}(R, M) = \begin{bmatrix} \textcolor{red}{4} & \mathbf{1} & \dots & \mathbf{5} & \textcolor{red}{2} \\ \mathbf{4} & \mathbf{2} & \dots & \textcolor{red}{3} & \textcolor{red}{3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{5} & \textcolor{red}{1} & \dots & \mathbf{4} & \textcolor{red}{1} \\ \textcolor{red}{3} & \textcolor{red}{2} & \dots & \mathbf{5} & \mathbf{2} \end{bmatrix}$$

Figure 2.5: An example of a rating matrix completion problem. The recommendation procedure $\hat{f}$ is tasked to predict the missing ratings of the incomplete user-item matrix R using the information provided by the matrix R and some metadata M about the items, if available.

$$L = \left\{ \begin{matrix} u_1 & t_{u_1}^1 & I_1 \\ u_1 & t_{u_1}^2 & I_1 \\ u_1 & t_{u_1}^3 & I_2 \\ u_2 & t_{u_2}^1 & I_2 \\ \vdots & \vdots & \vdots \\ u_n & t_{u_n}^j & I_n \end{matrix} \right\} \implies IF = \begin{bmatrix} 1 & 1 & \dots & ? & ? \\ ? & 1 & \dots & ? & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ? & ? & \dots & 1 & ? \\ 1 & ? & \dots & ? & ? \end{bmatrix}$$

Figure 2.6: Typical logs of views of items by users transformed into an implicit feedback matrix. The user u_1 viewed the item I_1 twice, its row is highlighted with orange in the IF matrix where these views were deduplicated and transformed to a positive signal for the item I_1 (column highlighted with yellow).

approaches offer distinct benefits, and the most successful recommender systems in use combine the strengths of both methods (Vasile et al., 2016; Jeunen et al., 2020). The early recommender systems were often limited by the availability of data, as well as by the computational resources needed to process that data. However, the ideas behind them laid the groundwork for more sophisticated recommendation paradigms that would emerge in the years to come. Even if it is by no means our ambition to provide an exhaustive covering of the recommendation research landscape in this introduction, we want to give the reader in the next paragraphs different perspectives on how recommendation is modelled.

2.3.1 Recommendation as Preference Prediction

The "Tapestry" system, which was introduced earlier, approached the recommendation problem as predicting the rating that a user would give to an item. This approach gained further popularity with the work of Resnick et al. (1994) within the GroupLens Research Lab, which provided a complete architecture to support research in this area. The idea is to have different users rate items and gather this information in a dataset. Since asking each user to rate all items is not feasible (think about massive movie catalogues, for example), users are randomly exposed to a few items for which they give a rating as shown in Harper and Konstan (2015). These ratings are then compiled and represented in a user-item rating matrix R of size $U \times I$ with U and I respectively the number of users and the number of items. Each entry in R, $R_{u,i}$, represents the given or missing rating of user u to item i. The goal is to learn a procedure $\hat{f}$ that can predict the missing ratings and complete the matrix R. In addition to the ratings' dataset, some metadata about the items M (relevant properties of the items) is often available (Harper and Konstan, 2015). This allows practitioners to explore different ways to combine the users ratings data (collaborative filtering) and item specific data (content-based filtering) to obtain a procedure that produces the most accurate predictions. The quality of the predictions is typically measured by assessing the difference between the true and estimated ratings on a separate test set (Salakhutdinov and Mnih, 2007). Figure 2.5 visualises an example of an incomplete rating dataset and the expected output of the procedure $\hat{f}$. Given R and/or M as input, the learned procedure $\hat{f}$ generates potential ratings for every user item pair. These complete ratings are then used to identify items that may be of interest to each user by selecting the ones with high predicted ratings. The underlying assumption of this approach is that items with higher ratings are considered suitable candidates for recommendation. This framework is referred to in the literature as the "explicit feedback" setting, as it requires users to explicitly provide ratings on a predefined scale. Recommendation based on explicit feedback has had a huge practical success, and were responsible for the success of many tech companies. For example, Netflix, a *DVD* rental service at the time, launched a competition rewarding a million US dollars to whoever achieves the smallest reconstruction error of their rating dataset. Despite its success, the "explicit feedback" paradigm suffers from significant limitations. The fundamental premise of the approach is to build a method that, gathers in an unbiased manner, genuine ratings that accurately reflect the user's true appreciation of items. However, obtaining this data requires the system to explicitly ask users to rate items, which can be costly and may be detrimental to the user's experience. To deal with this issue, these systems provide a non-intrusive way to rate an item; a like button to express if they loved the content they interacted with, or a rating system for the product bought from a retail store. These methods are integrated in the system and do not necessarily harm the user experience, but they give the entire freedom to the user to rate an item or not. This introduces an additional bias to the rating matrix as the presence of a rating is influenced by the decision of the user, making ratings "Missing Not At Random" (MNAR) (Yang et al., 2018). For example, once a user watches a movie, how much they enjoyed the movie influences directly the likelihood that they will leave a rating. Additionally, new ratings of an item tend to be biased by all the previous ratings that item received, making it hard to measure how much a new user really likes an item. Actually, in depth studies have shown that most ratings collected by these systems are biased, and correlate poorly with the true interest of users, as evidenced by Zhang et al. (2017). A potentially better signal to consider is the organic behaviour of users. By exploiting the information that is inherent to a user interacting with an item, we can avoid the need for explicit ratings. Indeed, we can reasonably assume that a user will mostly view retail product pages of items they are interested in, or movies and series that they think they will enjoy. This information is denoted in the literature by the "implicit feedback" as it is not asked directly from the user but reflects to a certain extent its interests through his organic interactions with the system. Implicit-feedback recommendation took the industry by storm and dominated the industrial applications in recent years. For example, Gomez-Uribe and Hunt (2016) describe the recommender system recently used by Netflix and show that they moved from their heavy dependence on rating feedback to a simpler feedback mechanism, focusing primarily on signal acquired from interaction data. This interaction data can come in different forms depending on the nature of the service provided. For online recommender systems, the most common form of interaction is a view/visit (or multiple views/visits) of an item by a user. In general, these views are logged, processed and deduplicated to build a matrix of binary, positive-only signal as shown in Figure 2.6. This simple organic signal differs from the explicit rating given by users, as it cannot encode negative information. When a user interacts with an item, we assume that the user is interested in the item. In the other hand, when an interaction is missing from the data, we do not know whether this means that the user is simply unaware of the item, or whether it is irrelevant to the user. The absence of negative signal motivated new collaborative filtering algorithms, sometimes augmented with item-related data,

$$L = \left\{\begin{matrix} u_1 & t_{u_1}^1 & I_1 \\ u_1 & t_{u_1}^2 & I_1 \\ u_1 & t_{u_1}^3 & I_2 \\ u_2 & t_{u_2}^1 & I_2 \\ \vdots & \vdots & \vdots \\ u_n & t_{u_n}^j & I_n \end{matrix}\right\} \Longrightarrow \left\{\begin{matrix} S_{u_1} = [I_{t_{u_1}^1}, I_{t_{u_1}^2}, I_{t_{u_1}^3}, \cdots] \\ \vdots \\ S_{u_n} = [I_{t_{u_n}^1}, I_{t_{u_n}^2}, I_{t_{u_n}^3}, \cdots] \end{matrix}\right\}$$

Figure 2.7: Typical logs of views of items by users transformed into user sessions, taking time into consideration. The user u_1 is represented by the session S_{u_1} constructed as a time-ordered list of the user u_1 interactions.

and moved the evaluation of such systems from computing a reconstruction error to ranking metrics which are deemed more useful in this scenario (Zangerle and Bauer, 2022). In particular cases, available organic behaviour might encode negative feedback. For example, video streaming platforms interpret a short watch time on a video as a negative feedback. However, this kind of signal is application-dependent and differs from the classical implicit feedback setting these methods are designed to solve.

2.3.2 Recommendation as Next-Event Prediction

Modelling Recommendation as Preference Prediction proved valuable in industrial applications as it extracts strong signal from the data we have on the users and the items. This approach relies on the underlying assumption that all the historical interactions are equally important to the user's current preference, which may not be true depending on the application. As shown in Figure 2.6, this modelling approach discards temporal information present in the logs L to construct the interaction matrix. This signal can be valuable as in a multitude of applications, a user's choice of items not only depends on long-term historical preference, but also on short-term and more recent preferences. In particular, choices always have time-sensitive context; for instance, "recently viewed" or "recently purchased" items are often more relevant than others. These short-term preferences are embedded in the user's most recent interactions, but may account for only a small proportion of historical interactions. These considerations have prompted the exploration and development of a new class of recommendation algorithms: known as Session-Based recommendation algorithms, seeing recommendation as *NEP: Next-Event Prediction* (Wang et al., 2021b). This approach heavily relies on the user's most recent interactions, rather than the entire user's historical preferences. Formally, a user session is composed of multiple interactions that happen together in a continuous period of time. For instance, products purchased in a single transaction or item viewed in a single shopping session. Depending on the application, the time window adopted varies and these sessions can occur on the same day, or across several days, weeks, or months. Figure 2.7 visualises an example transforming the raw logs of user interactions into user sessions. This paradigm wants to answer the following question: "If a user interacted with these items in this order, what should we recommend next?" Recommendation is framed as finding the item that will give the most probable sequence, increasing the likelihood that our algorithm completes the user session. The *BLOB* model (introduced in Chapter 6), especially its organic part, makes explicitly this assumption to extract signal from the user session. The evaluation of such systems can be done offline by splitting these user sessions, taking the first part as input, and measuring how well we can predict the second part of the session. Good recommendations in an e-commerce scenario

$$L = \left\{\begin{matrix} u_1 & t^1_{u_1} & V_1 \\ u_1 & t^2_{u_1} & V_1 \\ u_1 & t^3_{u_1} & V_2 \\ u_2 & t^1_{u_2} & V_2 \\ \vdots & \vdots & \vdots \\ u_n & t^j_{u_n} & V_n \end{matrix}\right\}, \quad B = \left\{\begin{matrix} u_1 & a_1 & r_1 & t_{a_1} \\ u_2 & a_2 & r_2 & t_{a_2} \\ u_3 & a_3 & r_3 & t_{a_3} \\ \vdots & \vdots & \vdots & \\ u_n & a_n & r_n & t_{a_n} \end{matrix}\right\}$$

Figure 2.8: We come across two different logs in modern recommendation. L represents the organic behaviour of the users without any intervention of the recommender systems; views of items by users navigating the website. B represents the data collected from the interactions of the recommender system with the users. We observe a signal r_n (sale) after taking action a_n (the item recommended) for user u_n.

help the user find items that complement his latest purchases, or in a streaming platform find music that fit exactly the mood of his last listening session. Poor recommendations will, at best, have no impact, and in the general case, result in a negative user experience. Many approaches were designed to tackle the next event prediction task. The simplest approach recommends the item that most frequently co-occurs with the last item in the session, as presented in Ludewig and Jannach (2018). This heuristic can be a good baseline, but does not capture the complex dependencies present in the user's session history. Advanced approaches view user sessions as sentences, drawing inspiration from recent advances in Natural Language Processing to capture semantic complexity between the different items, with intuitive approaches suggested in Vasile et al. (2016), which infers information about items in a similar fashion to the popular language model "word2vec" by Mikolov et al. (2013). More recently, deep learning approaches based on recurrent neural networks (Hidasi et al., 2015) and graph neural networks (Wu et al., 2019) have been adapted to the problem, achieving state-of-the-art results for session completion tasks.

2.3.3 Recommendation as an Interaction

In the previous sections, we presented two different paradigms of how the recommendation problem is modelled. These approaches cast recommendation into proxy problems that are easier to solve; a good recommendation is an item that is likely to be viewed by the user, or an item that complement well the user session. In the majority of applications, we are interested in aligning recommendation with business value. As discussed in Jannach and Jugovac (2019), the performance of modern recommender systems can be measured by its ability to retain users, increases their engagement or have positive impact on sales. Solving recommendation as a session completion task for example might correlate with these business metrics but is only considered a poor proxy for the real goal that modern recommender systems want to solve. In particular, the performance of an algorithm trained with the presented paradigms may be very different from its actual performance once deployed (Garcin et al., 2014). Instead of focusing on predicting the occurrence of interactions between users and items, a better modelling approach needs to focus on the impact of the recommender system and use the observed outcomes to update the model, making the system better aligned with business metrics at each new deployment. Within this paradigm, we want to answer the following question; "Can we make the recommender system more aligned with business value (generate more clicks, sales, etc) in the next deployment?". The sequential nature of the problem, where we need to observe the outcome of the decisions taken by the system and update it accordingly, motivated research to

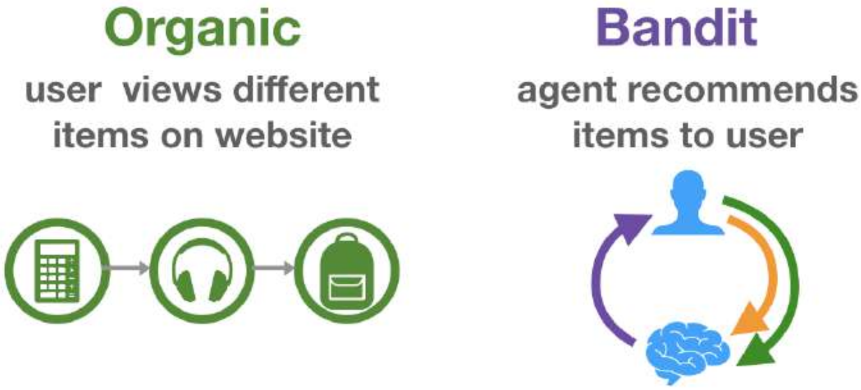

Organic
user views different items on website
Bandit
agent recommends items to user

Figure 2.9: Difference between Organic behaviour and Bandit feedback.

framing recommendation as a sequential decision-making problem under uncertainty (Roijers et al., 2013), with both Reinforcement Learning and Contextual Bandit considered suitable candidates for the task. These frameworks model an agent that encounters a state/context, performs an action and observes a short term/long term outcome depending on the application. In our case, a recommendation system encounters a user, chooses which items to recommend to this user and observes a complex signal. As such, real-world recommendation systems can be perfectly described within these frameworks. In the majority of scenarios, the online pipeline can log the interactions between the recommender system and the users. In addition, we also have in our disposal data on the intrinsic behaviour of our users with different items without the intervention of the system.

E-Commerce Website. Let us study a simplified example of an e-commerce website, where the goal is to build a recommender system that generates more sales. A customer will typically navigate the website, searches for some items that he is interested in. In this customer session, the recommender system intervenes and shows an item to the user in the hope to answer his needs, we then observe the interaction between the user and the recommended item, and see if it ends up in a sale. This use case summarizes the typical dynamics encountered in some modern recommender systems (Rohde et al., 2018). In this example, the system logs different information about the user behaviour, that can be split into two classes, detailed in the following:

- **Organic Behaviour** represents the natural interaction of the user with the website. From searching for a specific item, to viewing products on a sub category of an e-commerce website. This feedback is organic to the user and gives us information about users and items regardless of the impact of the recommender system.

- **Bandit Feedback** represents an intervention of our system, one where we have the opportunity to show the user an item and observe the outcome of the interaction, in this case, if the user did purchase the recommended item or not.

The reader can refer to Figure 2.8 to have an idea of how the data collected looks like, and Figure 2.9 for a high level illustration of the difference between these signals. Roughly speaking, if classical recommendation paradigms modelled the problem using the organic signal; how likely the item is to be viewed by the user, to achieve better alignment with business metrics, which is in this case, generating more sales, our focus should be directed to the **Bandit**

Feedback that represents the outcomes that we are interested in. The main challenge inherent to learning from this data originates from the fact that available observations are biased towards actions favoured by the initial recommender system - the one that was previously deployed and acting online. Addressing this bias gives birth to a new class of Recommender Systems, either designed within the Contextual Bandit Framework to handle action-level business metrics (CTR: Click-Through-Rate, Dwell Time, watch time, etc) as in Chen et al. (2019a); Ma et al. (2020), or designed within the Reinforcement Learning Framework to handle long-term/timeline level business metrics (Sales, Engagement, Retention, etc) as in Afsar et al. (2022); Chen et al. (2022); Wang et al. (2022). This opens exciting avenues of research as industrial recommender systems come with their own challenges; i.e. adapting these methods to large, discrete action spaces. Indeed, recommendation in practice aims at identifying the need of users and answering it by choosing an action from a potentially massive catalogue. These systems are tasked to deliver recommendations rapidly, and are constrained to particular architectures to achieve this. After taking the time to introduce the recommendation research landscape, we will cover in what follows the various constraints imposed on recommender engines that deal with large catalogues.

2.4 Constraints of Large Scale Recommender Systems

Recommendation is the act of accurately understanding the need of a user and answering it. This process is achieved by both learning about the users and the catalogue of items from which recommendations are done. Different paradigms can model the recommendation problem, but they all follow the same conception. Learning a recommender engine usually boils down to defining a score function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ that attributes for any user (context) x and action (item) a a score $f(x, a)$, with the convention that a higher score means better relevance. Once we are convinced with the scoring function, delivering recommendations for a user x online is done through the identification of the highest scoring actions. Our system can deliver one item or multiple items depending on the application. Let K be the number of items needed, the recommendation process is formalized by solving the following:

$$[a_1, ..., a_K] = \underset{a' \in \mathcal{A}}{\arg\overset{K}{\text{sort}}} \left\{ f(a, x) \right\}, \tag{2.8}$$

with the operator $\arg\text{sort}^K_{a' \in \mathcal{A}}$ returning the K highest scoring actions. This sorting operation has a linear complexity on the size of the action space $\mathcal{O}(|\mathcal{A}| \log K)$ and cannot be adopted in a large scale production environment. A simple solution to this problem is to build a two stage decision system (Borisyuk et al., 2016). The idea is to first generate a *small* subset of potential action candidates $\mathcal{A}_{\text{sub}} \subset \mathcal{A}$ with $|\mathcal{A}_{\text{sub}}| \ll |\mathcal{A}|$, and then score this subset instead to select the top-K actions in $\mathcal{A}_{\text{sub}}$ leading to a $\mathcal{O}(|\mathcal{A}_{\text{sub}}| \log K)$ delivery time. This is generally achieved by pre-constructing the subset $\mathcal{A}_{\text{sub}}$, so as the complexity of this step does not add to the time complexity of the delivery. This approach has a major shortcoming. Usually, the candidate generation step and the scoring models are not trained jointly, which leads to delivering suboptimal actions; the subset $\mathcal{A}_{\text{sub}}$ may not contain the highest scoring actions.

A more satisfying approach is to avoid candidate generation and rely instead on the structure of the score function to accelerate the sorting step on the entirety of the action space. Modern large scale recommendation systems adopt the two-tower model (Huang et al., 2013; Li et al., 2022). This architecture consists of having two encoders, that embed users and items in the same space, followed by an Approximate Nearest Neighbour (ANN) (Wang et al., 2021a) search to select top items given the user's embedding. This architecture allows the delivery of recommendation in logarithmic time. Some score functions are compatible with this approach:

$$\forall(x,a) \quad f_\theta(a,x) = h_\Xi(x)^\intercal \beta_a \tag{MIPS}$$

$$\forall(x,a) \quad f_\theta(a,x) = -||h_\Xi(x) - \beta_a||^2 \tag{L_2}$$

with $\theta = [\Xi, \beta]$ the parameters of the encoders. The MIPS: Maximum Inner Product Search is the most adopted one. In this structure, the score function becomes an inner product between a context embedding $h_\Xi(x)$ and an action embedding β_a, both residing in a latent space $\mathbb{R}^l$ of dimension $l \ll |\mathcal{A}|$. With this architecture, Equation (2.8) can be solved with *approximate* MIPS: Maximum Inner Product Search algorithms (Shrivastava and Li, 2014) in a time complexity of $\mathcal{O}(\log|\mathcal{A}|)$ instead of $\mathcal{O}(|\mathcal{A}|)$, rendering fast decision-making possible without additional considerations. This alleviates the need for a candidate generation step and promises the delivery of optimal actions under the learned score function.

The second aspect to take into account is the training of such systems. Recommendation engines need to constantly understand the ever-shifting needs of the users. These systems are updated frequently, which makes fast training highly desirable. Large scale collaborative filtering-based recommender systems rely on the understanding and factorization of the user-item interaction matrix into a latent space. Accelerating the learning within this paradigm relies on the acceleration of matrix manipulation techniques, such as the inversion or factorization of a matrix. Motivated by large scale recommendation applications, various techniques were proposed to approximate the inverse of a matrix (Steck, 2020), accelerate weighted matrix factorization (He et al., 2016; Chen et al., 2020), alternating least square (Hastie et al., 2015) and Singular Value Decomposition (Janeković and Bojanjac, 2021; Boulle and Townsend, 2022). These advancements helped scale common collaborative filtering approaches to large catalogues. Modelling recommendation as next-event prediction benefits from advances in Natural Language processing, as both methods rely on efficiently handling sequences of tokens/items coming from a large corpus/catalogue. Notable acceleration techniques are based on contrastive learning (Xie et al., 2021), negative sampling (Tanielian and Vasile, 2019; Chen et al., 2023) and practically all accelerated, sequence-based deep learning methods (Vaswani et al., 2017a; Zandieh et al., 2023).

Our work frames the recommendation problem as an interaction, and formalizes it with the help of offline contextual bandit tools. Learning within this framework can be achieved through the direct method or importance weighting objectives. The direct method relies on regressing a score function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ on the observed reward. This method benefits naturally from recent advances of deep learning methods that can scale the training of large models on large datasets (Shen et al., 2023). On the other hand, we have the importance weight path that learns a policy directly. Handling policies over large catalogues involves computing sums over the whole action space. In particular, we need to be extra-careful when computing/approximating gradients of our objectives because this operation scales linearly in $|\mathcal{A}|$, which can drastically slow down the optimization routine. The question of scaling general off-policy learning objectives attracted little attention; Chen et al. (2019a) learned a production-ready policy with an `IPS`-based objective without any focus on the computational aspect. We are interested in this question and want to provide general acceleration methods. If every learning objective has its own expression and properties, a large panel can be written under a unified framework. For a policy π, we can recover a multitude of well-known estimators and objectives by the following expression:

$$\hat{R}_n^{\texttt{LIN}-\hat{r}}(\pi) = -\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{a\sim\pi(\cdot|x_i)}\left[\hat{r}(a, x_i)\right]. \tag{2.9}$$

with $\hat{r}$ a reward estimator, that can also include a regularizer. For example, the previously defined risk estimators are obtained by the following:

$$\forall(x,a) \quad \hat{r}_{\texttt{DM}}(a,x) = r_{\mathcal{M}}(a,x) \implies \hat{R}_n^{\texttt{DM}}(\pi)$$

$$\forall(x,a) \quad \hat{r}_{\texttt{IPS}}(a,x) = \begin{cases} r_i/\pi_0(a_i|x_i) & \text{if } (a,x) = (a_i,x_i), \\ 0 & \text{otherwise.} \end{cases} \implies \hat{R}_n^{\texttt{IPS}}(\pi)$$

$$\forall(x,a) \quad \hat{r}_{\texttt{DR}}(a,x) = \begin{cases} (r_i - r_{\mathcal{M}}(a,x))/\pi_0(a_i|x_i) + r_{\mathcal{M}}(a,x) & \text{if } (a,x) = (a_i,x_i), \\ r_{\mathcal{M}}(a,x) & \text{otherwise.} \end{cases} \implies \hat{R}_n^{\texttt{DR}}(\pi).$$

The `LIN` notation in the estimator stands for *linear* as these estimators are linear in the policy evaluated. This family can recover commonly adopted estimators (Horvitz and Thompson, 1952; Dudík et al., 2014; Wang et al., 2017; Saito and Joachims, 2022a; Saito et al., 2023; Aouali et al., 2023a), and principled learning objectives (London and Sandler, 2019; Sakhi et al., 2023a).

We are particularly interested in learning objectives of this form because they write down nicely as an expectation of the quantity $\hat{r}$ under actions coming from π. This means that even if we cannot compute the expectation exactly (we want to avoid a sum over a large action spaces $|A| \gg 1$), one can still approximate it to a desired precision if we can sample efficiently from π. The last two chapters of the thesis develop sublinear optimization routines to learn MIPS-based policies with objectives of the form in Equation (2.9).

Part I

Offline Learning with Performance Guarantees

Chapter 3

Offline Learning with Distributionally Robust Optimization

Abstract

This chapter extends the Distributionally Robust Optimization (DRO) approach for offline contextual bandits laid out in Faury et al. (2020). Specifically, we leverage this framework to introduce a *convex* reformulation of the Counterfactual Risk Minimization principle introduced in Swaminathan and Joachims (2015a). Besides relying on convex programs, the proposed approach is compatible with stochastic optimization, and can therefore be readily adapted to the large data regime. Our procedure relies on the construction of asymptotic confidence intervals for offline contextual bandits through the DRO framework. By leveraging known asymptotic results of robust estimators, we also show how to automatically calibrate such confidence intervals, which in turn removes the burden of hyperparameter selection for policy optimization. We present empirical results supporting the effectiveness of our approach.

Contents

3.1 Introduction

3.1.1 Contextual Bandits.

The Contextual Bandit (CB) framework is a formalization of an important sequential decision-making problem, with impactful applications in recommender systems (Li et al., 2010; Valko et al., 2014), mobile health (Tewari and Murphy, 2017) and clinical trials (Villar et al., 2015). It describes a repeated game between a decision-maker and an environment. The latter sequentially reveal sets of available actions to the former, along with some additional side information for each action. Such additional information is assumed to carry informative signal about the intrinsic value (or *reward*) of its associated action. Informally, the goal of the decision-maker is to discover an efficient strategy (or *policy*) to select, given a context, a nearly optimal action.

3.1.2 Offline Policy Learning.

The Contextual Bandit optimization literature can be divided into two streams. The first studies its *online* formulation, where the focus lies on the exploration/exploitation trade-off for *regret* minimization (Lattimore and Szepesvári, 2020). This chapter is concerned with the second, known as Off-Policy learning, or Batch Learning from Bandit Feedback (Swaminathan and Joachims, 2015a) (BLBF). This setting is arguably better suited for applications in real-life situations. The optimization is performed *offline* and based on historical data, typically obtained by logging the interactions between an older version of the current policy and the environment. The learning problem consists in leveraging this data (necessarily biased towards actions favoured by the logged policy) to discover new strategies of greater performance.

3.1.3 Prior work and limitations.

The first step in addressing the Offline policy learning problem is to remove the intrinsic bias introduced by the logging policy (Bottou et al., 2013). This however can come at the price of building high-variance estimates (Swaminathan and Joachims, 2015a) for the performance of the current policy, which in turns can lead to high *post-decision* regret. To address such challenges, Swaminathan and Joachims (2015a) introduced the Counterfactual Risk Minimization (CRM) principle. It combines debiasing through importance re-weighting (Horvitz and Thompson, 1952) with a modified policy-selection process that penalizes policies with high-variance estimates. Recently, Faury et al. (2020) proposed a generalization of the CRM principle through the Distributionally Robust Optimization (DRO) framework. This led them to the development of a new BLBF algorithm, obtained through a specialization of this general framework. However, while the methods introduced in both Swaminathan and Joachims (2015a) and Faury et al. (2020) offer some desirable theoretical guarantees for off-line policy optimization, their respective implementation suffers from important caveats. Namely, they rely on optimizing *non-convex* objectives, which is notably hard from a theoretical perspective. Further, these objective are not well-suited for stochastic optimization (useful when the logged data is large and cannot fit in memory), as obtaining unbiased stochastic gradients for these objective is not straight-forward. Last, they rely on the selection of rather sensitive hyperparameters, of which the (approximate) automatic calibration (through asymptotic arguments, for instance) is unknown.

3.1.4 Contributions.

In this chapter, we further investigate the DRO framework for offline CB introduced in Faury et al. (2020). This leads to a reformulation of the CRM principle that boils down to solving a convex problem. This reformulation brings the first fully stochastic algorithm of CRM solving this objective for large logged datasets. Our approach relies on the construction of asymptotic confidence intervals for offline CB through the DRO framework. Leveraging known asymptotic results for DRO we show how to automatically calibrate such confidence intervals, which in turns remove the need for hyperparameter optimization for policy optimization. We validate our approach through extensive experiments on standard datasets for this task.

3.2 Preliminaries

3.2.1 Notations

In the following, we will write 1_n to be the n-dimensional vector which entries are all equal to $1/n$. For any positive integer m, Δ_m denotes the m-dimensional simplex. φ is a real-valued convex function. We denote $\varphi^\star(s) = \sup_{x\in\mathbb{R}}(xs - \varphi(x))$ the Fenchel conjugate of φ. For two distributions p and q, $q \ll p$ means that q is absolutely continuous with respect to p ($supp(q) \subset supp(p)$).

The notation d_φ refers to the f-divergence associated to φ, that is defined for discrete distributions p and q in Δ_m by:

$$d_\varphi(q,p) = \sum_{i=1}^{n} p_i \varphi\left(\frac{q_i}{p_i}\right) \quad q \ll p.$$

3.2.2 Setting

We will use $x \in \mathcal{X}$ to denote a context and $a \in \mathcal{A}$ an action, where $|\mathcal{A}|$ denotes the number of available actions. Given a context x, each action is associated with a cost $c(x,a) \in [-1,0]$[1], with the convention that better actions have smaller cost. The cost function c is unknown. A decision maker is represented by its policy π, which maps each context $x \in \mathcal{X}$ to the $|\mathcal{A}|$-dimensional simplex $\Delta_{|\mathcal{A}|}$. Assuming that the contexts are stochastic and follow an unknown distribution ν, we define the *risk* of the policy π as the expected cost one suffers when playing actions according to π:

$$R(\pi) = \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[c(x,a)\right].$$

The learning problem is to find a policy π with the smallest risk. In most real world problems, it is not reasonable to expect having the luxury of testing out several policies to compare their empirical risk and retain whichever policy has the smallest. This issue is usually circumvented by forecast the risk of a given policy thanks to some existing interaction data. This is formalized through a *logging policy* π_0 which has already been deployed in the environment (e.g a previous version of a recommender system that the practitioner is trying to improve) for which we assume we have the following interactions:

$$\mathcal{D}_n = \{x_i, a_i \sim \pi_0(x_i), \pi_0(x_i,a_i), c(a_i,x_i)\}_{i\in[n]}.$$

Based on this data, one can build an unbiased (under mild assumptions) estimator of the risk of any policy π through the use of importance weights (Horvitz and Thompson, 1952):

$$\hat{R}_n^{\texttt{IPS}}(\pi) := \frac{1}{n}\sum_{i=1}^{n} r_\pi(x_i,a_i)$$

where $r_\pi(x_i,a_i) := \omega_\pi(a_i|x_i)c(x_i,a_i)$ and $\omega_\pi(a|x) := \pi(a|x)/\pi_0(a|x)$. This estimate is commonly referred to as the IPS (Inverse Propensity Scoring) risk.

[1] We make this assumption for ease of exposition. It can be explicitly enforced by re-scaling the cost function.

3.2.3 Counterfactual Risk Minimization

The IPS estimator of the risk $R(\pi)$ has potentially high variance, which depends on the disparity between π and π_0 (Swaminathan and Joachims, 2015a, Section 4). Hence, directly sorting candidate policies thanks to their IPS risk is hazardous (as it boils down to comparing estimators with potentially high and different variances) and is known to be suboptimal.

Swaminathan and Joachims (2015a) proposed to add an *empirical variance* term to the IPS risk in order to penalize policies with high-variance estimate. Coined Counterfactual Risk Minimization (CRM) principle, this suggests finding the policy which minimizes:

$$\text{Risk}_n^{\lambda}(\pi) = \hat{R}_n^{\texttt{IPS}}(\pi) + \lambda\sqrt{\frac{\widehat{\text{Var}}_n(\pi)}{n}} \tag{3.1}$$

where λ is a tunable hyperparameter and $\widehat{\text{Var}}_n(\pi) = \frac{1}{n-1}\sum_{i=1}^{n}\Big(r_\pi(x_i,a_i) - \hat{R}_n(\pi)\Big)^2$ is the empirical variance of $\hat{R}_n(\pi)$. This policy selection process is based on *variance-sensitive* confidence intervals for the true risk obtained via empirical Bernstein bounds (Maurer and Pontil, 2009).

3.2.4 Generalization through DRO

Based on a similar intuition, Faury et al. (2020) recently introduced the idea of using Distributionally Robust Optimization (DRO) tools for this policy optimization problem. They showed that for a particular class of φ-divergence, the robust risk $\text{RobustRisk}_n^{\varphi}(\pi,\epsilon)$ defined as:

$$\text{RobustRisk}_n^{\varphi}(\pi,\epsilon) := \sup_{q\in\Delta_n^{\varphi}(\epsilon)} \sum_{i=1}^{n} q_i r_\pi(x_i,a_i) \tag{3.2}$$

where $\Delta_n^{\varphi}(\epsilon) := \{q \in \Delta_n \,|\, d_\varphi(q, 1_n) \le \epsilon\}$ is a variance-sensitive (asymptotic) upper-bound for the true risk. It is therefore well-suited for the policy optimization task, and generalizes (in some sense) the CRM approach (Faury et al., 2020, Lemma 3). For the KL-divergence, the $\text{RobustRisk}_n^{\text{KL}}(\pi,\epsilon)$ has a closed-formed which can be directly minimized (Faury et al., 2020, Lemma 4):

$$\text{RobustRisk}_n^{\text{KL}}(\pi,\epsilon) = \sum_{i=1}^{n} \frac{\exp(r_\pi(x_i,a_i)/\gamma)}{\sum_j \exp(r_\pi(x_j,a_j))/\gamma)} r_\pi(x_i,a_i)$$

with γ being a tunable hyperparameter.

3.2.5 Limitations and contributions

Variance penalization of the IPS objective, and its DRO generalization, benefit from a strong theoretical justification and yield better performing policies. These algorithms, that were proven to outperform the simple IPS objective empirically in Faury et al. (2020); Swaminathan and Joachims (2015a) can still be improved as they suffer from huge limitations.

- Contrary to the IPS objective, which is linear (and therefore convex) in π, both the CRM principle (adding a square root variance penalization) and DRO as it was introduced in Faury et al. (2020) (the adversary distribution is exponential on the loss) break this convexity, resulting in problems that are way harder to optimize, meaning that the benefits from such formulations are always down weighted with the non-convexity induced.

Divergence	$\varphi(t)$	$d_\varphi(q\|p)$	$\varphi^*(s)$
χ-Square	$(t-1)^2$	$\sum_{i=1}^n \frac{(q_i-p_i)^2}{p_i}$	$\begin{cases} s+s^2/4 & s \geq -2 \\ -1 & s \leq -2 \end{cases}$
Kullback-Leibler	$t\log t - t + 1$	$\sum_{i=1}^n q_i \log(q_i/p_i)$	$e^s - 1$
Burg entropy	$-\log t + t - 1$	$\sum_{i=1}^n p_i \log(p_i/q_i)$	$-\log(1-s)$, $s < 1$
Hellinger distance	$(\sqrt{t}-1)^2$	$\sum_{i=1}^n \left(\sqrt{p_i} - \sqrt{q_i}\right)^2$	$\frac{s}{1-s}$, $s \leq 1$

Table 3.1: Some coherent φ-divergences and their characterizations.

- Another important limitation of such objectives is its scalability in the context of large datasets. Both formulations are not well-adapted to the stochastic gradient descent algorithm, and naively using it will lead to biased estimates of the gradients. The proposed algorithm in Faury et al. (2020) works only in the batch setting as the adversary distribution needs all the data to be normalized, and even if Swaminathan and Joachims (2015a) suggested a relaxation of CRM amenable to stochastic gradients, it is built on a majorisation-minimisation algorithm that needs to load the entire dataset at certain times, which is not practical in the large data regime.

- These algorithms also come with hyperparameters that need careful tuning as their choice drastically impact the performance of the obtained policy, making the whole optimization procedure even harder. Swaminathan and Joachims (2015a) treats λ, the weight of the variance penalty, as a hyperparameter, while Faury et al. (2020) treat ϵ, the maximum distance between the adversarial and the nominal distribution, which has a close connection to λ, as a hyperparameter as well.

Both algorithms are deemed impractical in real life, as they have a non-convex loss surface to optimize, are not applicable to huge datasets and need heavy hyperparameter tuning. This work tries to circumvent these limitations through a more careful treatment of the DRO formulation, leading to general algorithms that treat all these caveats in a well-defined and unified framework.

3.3 Policy Evaluation and Optimization

3.3.1 Policy Evaluation: Confidence Intervals

In this section, we briefly review and discuss how the robust risk can lead to the construction of confidence intervals for the true risk. This is a crucial step for policy optimization, as it will consist in minimizing the high-probability upper-bound on the true risk provided by the policy evaluation procedure. Designing tight confidence interval for the risk can also be a goal in itself, in order to fully evaluate the potential benefits/risks of deploying a given policy (e.g provide an offline metric for A/B testing). In the following, we build on Faury et al. (2020) and consider *coherent* φ-divergence - i.e we will assume that φ satisfies the conditions of Assumption 1 in Faury et al. (2020). We provide some examples of such functions (and their associated divergence measure) in Table 3.1.

Under such conditions, one can show that the robust risk accounts for the variance of the IPS risk estimator (Faury et al., 2020, Lemma 2). Further, DRO can be leveraged to build asymptotic confidence intervals for the robust-risk. Indeed, let us introduce the following problem that defines an Optimistic Risk $\text{OptimisticRisk}_n^\varphi(\pi, \epsilon)$:

$$\text{OptimisticRisk}_n^\varphi(\pi, \epsilon) := \inf_{q \in \Delta_n^\varphi(\epsilon)} \sum_{i=1}^n q_i r_\pi(x_i, a_i) \tag{3.3}$$

We have the following result, extending Lemma 1 of Faury et al. (2020).

Lemma 3.3.1. *[Asymptotic Confidence Interval] Let $\delta \in [0,1)$. For $\alpha \in (0,1)$ denote ρ_α the $(1-\alpha)$-quantile of the one-dimensional χ^2 distribution. We also denote $RCI_n^\varphi(\pi,\epsilon)$, the interval* $[\text{OptimisticRisk}_n^\varphi(\pi,\epsilon), \text{RobustRisk}_n^\varphi(\pi,\epsilon)]$ *Then:*

$$\lim_{n\to\infty} \mathbb{P}\left[R(\pi) \in RCI_n^\varphi(\pi, \frac{\varphi'(1)\rho_\alpha}{2n})\right] \geq 1-\delta. \tag{3.4}$$

The proof of this lemma can be found in Duchi et al. (2021). This result states that the interval $[\text{OptimisticRisk}_n^\varphi(\pi,\epsilon), \text{RobustRisk}_n^\varphi(\pi,\epsilon)]$ is a asymptotic $(1-\delta)$ confidence interval for the true risk, when the size of the ambiguity-set ϵ is set to $\frac{\varphi'(1)\rho_\alpha}{2n}$. We will show in Section 3.4.1 that despite being asymptotic, this interval is empirically *tight* and displays satisfying coverage, motivating its use in real-life applications. It turns out that the programs for computing the robust and optimistic risk (Equation (3.2) and (3.3), respectively) can be efficiently solved. We will only review here the computation of the robust risk, however a similar reasoning holds for the optimistic risk. Notice that the objective in Equation (3.2) is linear in the variable q, which acts as a re-weighting for the counterfactual costs. Further, the constraint set $\{q \in \Delta_n \,|\, d_\varphi(q, 1_n) \leq \epsilon\}$ is convex. The program is therefore *convex* and can henceforth be solved efficiently. In this work, we rather rely on the dual formulation, which allows for efficient solving and is well adapted to the stochastic setting. We rely on the following result to characterize the robust risk.

Lemma 3.3.2 (Dual program for the robust risk)**.** *Let:*

$$g_\pi(\beta,\gamma) = \beta + \gamma\epsilon + \frac{1}{n}\sum_{s=1}^{n} (\gamma\varphi)^\star\left(r_\pi(x_i,a_i) - \beta\right) \tag{3.5}$$

where $(\gamma\varphi)^\star(s) = \gamma\varphi^\star(s/\gamma)$, with the convention that $(0\varphi)^\star(s) = +\infty$ is $s > 0$ and 0 otherwise. The function $(\pi,\beta,\gamma) \to g_\pi(\beta,\gamma)$ is convex and:

$$\text{RobustRisk}_n^\varphi(\pi,\epsilon) = \inf_{\beta,\gamma\geq 0} g_\pi(\beta,\gamma) \tag{\textbf{DRO-PE}}$$

This robust program characterization can be extracted from more general results - see for instance (Ben-Tal et al., 2013, Section 4). We provide a detailed proof in the following for the sake of completeness.

Proof. Recall the definition of the robust risk:

$$\text{RobustRisk}_n^\varphi(\pi,\epsilon) := \sup_{q\in\Delta_n} \left\{ \sum_{i=1}^{n} q_i \omega_\pi(x_i,a_i)c(x_i,a_i) \quad \text{s.t} \quad d_\varphi(q, 1_n) \leq \epsilon \right\} \tag{\textbf{P}}$$

where:

$$\begin{cases} \Delta_n = \left\{ p \in \mathbb{R}_n^+ \,\middle|\, \sum_{i=1}^{n} p_i = 1 \right\} \\ \mathbb{1}_n = \frac{1}{n}(1\dots1)^\intercal \in \mathbb{R}^n \\ d_\varphi(q,p) = \sum_{i=1}^{n} p_i\varphi\left(\frac{q_i}{p_i}\right) \quad \forall q \ll p \in \Delta_n \end{cases}$$

Note that the program (**P**) optimizes a linear objective under convex constraints (since φ is convex). Further, when $\epsilon > 0$, the candidate $q = \mathbb{1}_n$ is *strictly* feasible. Therefore, Slater's condition holds and (**P**) enjoys strong duality. Writing down its Lagrangian, we obtain the following equivalence:

$$\begin{aligned}
\text{RobustRisk}_n^{\varphi}(\pi, \epsilon) &= \sup_{q \succeq 0} \inf_{\beta, \gamma \geq 0} \sum_{i=1}^{n} q_i \omega_\pi(x_i, a_i) c(x_i, a_i) + \beta \left(1 - \sum_{i=1}^{n} q_i \right) + \gamma \left(\epsilon - \frac{1}{n} \sum_{i=1}^{n} \varphi(nq_i) \right) \\
&= \inf_{\beta, \gamma \geq 0} \sup_{q \succeq 0} \sum_{i=1}^{n} q_i \omega_\pi(x_i, a_i) c(x_i, a_i) + \beta \left(1 - \sum_{i=1}^{n} q_i \right) + \gamma \left(\epsilon - \frac{1}{n} \sum_{i=1}^{n} \varphi(nq_i) \right) \\
&= \inf_{\beta, \gamma \geq 0} \beta + \gamma \epsilon + \frac{1}{n} \sum_{i=1}^{n} \sup_{q_i \geq 0} \left\{ (nq_i) \omega_\pi(x_i, a_i) c(x_i, a_i) - \gamma \varphi(nq_i) \right\} \qquad (3.6)
\end{aligned}$$

where the first equality is a consequence of strong duality, and the second is obtained through simple re-arranging. If $\gamma \neq 0$, easy computations lead to:

$$\begin{aligned}
\text{RobustRisk}_n^{\varphi}(\pi, \epsilon) &= \inf_{\beta, \gamma \geq 0} \beta + \gamma \epsilon + \frac{\gamma}{n} \sum_{i=1}^{n} \sup_{q_i \geq 0} \left\{ (nq_i) \frac{\omega_\pi(x_i, a_i) c(x_i, a_i)}{\gamma} - \varphi(nq_i) \right\} \\
&= \inf_{\beta, \gamma \geq 0} \beta + \gamma \epsilon + \frac{\gamma}{n} \sum_{i=1}^{n} \varphi^\star \left(\frac{\omega_\pi(x_i, a_i) c(x_i, a_i)}{\gamma} \right)
\end{aligned}$$

by using the definition of $\varphi^\star$. The limit conditions announced in the Lemma are easily checked by computing the dual function when $\gamma = 0$. We therefore obtain the equality announced by using the definition of g_π:

$$\text{RobustRisk}_n^{\varphi}(\pi, \epsilon) = \inf_{\beta, \gamma \geq 0} g_\pi(\beta, \gamma)$$

The convexity of g_π can be obtained two ways; (1) by noticing that g_π is obtained through convexity-transforming transformations of a *perspective* function (Combettes, 2018), or (2) by noticing thanks to Equation (3.6) that:

$$(\pi, \beta, \gamma) \to \sum_{i=1}^{n} \sup_{q_i \geq 0} \left\{ (nq_i) \omega_\pi(x_i, a_i) c(x_i, a_i) - \gamma \varphi(nq_i) \right\}$$

is convex as a sum of supremum of linear (and hence convex) functions. ■

In a few words, Lemma 3.3.2 states the the robust risk can be efficiently computed by solving a two-dimensional convex program. When n is reasonably small (in other words, when the dataset $\mathcal{D}_n$ fits in memory), coordinate descent (with exact line search) provides an efficient, principled tool for computing the robust risk. The program (**DRO-PE**) is also well-suited for the large-data regime - *e.g* large n, as it naturally adapts to stochastic optimization. Indeed, the function g (Equation (3.5)) is composite and unbiased gradients of this objective are easily obtainable. Stochastic gradient descent methods (e.g Ruder (2016) for a modern overview) therefore provide efficient and flexible solutions for this problem (up to some mild modifications to account for the fact that the g is not smooth for γ in a neighborhood of 0). To sum-up, we showed here how the DRO method could be used to build confidence intervals for the true risk, by simply relying on solving convex programs. This confidence intervals are however only asymptotic; we will show in Section 3.4 that, still, they provide sufficient empirical coverage while being much tighter than their finite-time counterparts.

3.3.2 Policy Optimization: Towards a Convex Objective

General principle:

The CRM principle casts policy optimization in a theoretically sound framework, and interestingly enough is closely related to the robust risk defined throughout the chapter. Relying on results from Duchi et al. (2021), Faury et al. (2020) showed that the robust risk provides an asymptotic approximation to the variance regularized empirical risk:

$$\text{RobustRisk}_n^{\varphi}(\pi, \lambda^2/n) = \hat{R}_n(\pi) + \lambda\sqrt{\frac{\widehat{\text{Var}}_n(\pi)}{n}} + o\left(1/\sqrt{n}\right).$$

The original CRM principle for policy optimization (Equation (3.1)) can therefore be rethought as the minimization of a φ-robust risk. Using the dual formulation of the robust risk given in Equation (**DRO-PE**), the policy optimization objective becomes :

$$\inf_{\pi} \text{RobustRisk}_n^{\varphi}(\pi, \epsilon) = \inf_{\pi,\beta,\gamma\geq 0} g_\pi(\beta, \gamma) \qquad (\textbf{DRO-PO})$$

Note that as a consequence of Lemma 3.3.2, this objective is *convex*. It can therefore be minimized in principled ways, while enjoying similar guarantees as the original CRM objective. Naturally, one can expect such an important transformation of the optimization properties of the policy improvement objective to lead to greater practical performances.

There are several ways to solve the policy improvement objective (**DRO-PO**). For example, we can obtain (sub-)gradients of the robust-risk (w.r.t to the policy π) by first solving the policy evaluation program (**DRO-PE**). Formally, solving for $(\beta^\star, \gamma^\star) \in \arg\max_{\beta,\gamma} g_\pi(\beta, \gamma)$ allows to compute the adversarial re-weighting $q^\star$ through the following conversion, which holds up to a normalization constant:

$$\forall i \in [n], \quad q_i^\star = \varphi^\star\left(\frac{r_\pi(x_i, a_i) - \beta^\star}{\gamma^\star}\right).$$

A sub-gradient of the robust risk is then easily computable by differentiating:

$$\pi \to \sum_{i=1}^{n} q_i^\star r_\pi(x_i, a_i).$$

A more interesting approach is to solve the dual program jointly for:

$$(\pi^\star, \beta^\star, \gamma^\star) \in \arg\max_{\pi,\beta,\gamma} g_\pi(\beta, \gamma),$$

feeding gradient of the function $(\beta, \gamma, \pi) \to g_\pi(\beta, \gamma)$ to a gradient optimizer. In all of our experiments, we solve the dual program with the L-BFGS solver in the batch setting, and with the Adam solver in the stochastic setting as this formulation is composite and allows for stochastic optimization naturally.

Towards Better Behaved Objectives:

Variance Reduction. The methods presented so far rely on vanilla IPS. It is well known that this estimator suffers from large variance which can lead to poor performances - whatever the policy optimization algorithm used. Fortunately, our method easily extend to other estimators, so long that they remain convex in π - this is the case for the Clipped IPS[2] or the Doubly Robust

[2] as long as the costs are negative

(Dudík et al., 2014) estimator. Still, it would be useful to extend our method to estimators that actively reduce variance. A candidate for this task is the self-normalized importance sampling estimator of Swaminathan and Joachims (2015b). This estimator is unfortunately not convex in π, which goes against the efforts undertaken in this chapter to maintain well-behaved optimization tasks. We provide here an alternative which uses a simple additive control variate (instead of a multiplicative one). Formally, we rely on the following estimator:

$$\text{Risk}_{n,\rho}(\pi) = \frac{1}{n}\sum_{i=1}^{n} r_\pi^\rho(x_i, a_i)$$

where $r_\pi^\rho(x_i, a_i) = (c(x_i, a_i) - \rho)\,\omega_\pi(a_i|x_i) + \rho$. A robust version of this estimator easily follows, and enjoys the same convex properties of the IPS robust risk. The variance-reduction property of the additive control variate is presented in the following Lemma.

Lemma 3.3.3 (Propensity weights as an additive control variate)**.** *For all ρ,* $\text{Risk}_{n,\rho}(\pi)$ *is an unbiased estimator of $R(\pi)$, achieving a better variance than naive IPS whenever:*

$$2\frac{Cov(r_\pi, \omega_\pi)}{\mathbf{V}(\omega_\pi)} \leq \rho \leq 0$$

and with optimal variance reduction $\mathbf{V}(r_\pi^{\rho^*}) = (1 - corr(r_\pi, \omega_\pi)^2)\mathbf{V}(r_\pi)$ attained at

$$\rho^* = \frac{Cov(r_\pi, \omega_\pi)}{\mathbf{V}(\omega_\pi)}$$

In addition, if the cost is independent of the propensity weights, we obtain $\rho^* = \mathbb{E}[c]$.

In practice, we don't know how to derive ρ^* analytically. A straightforward way to obtain an unbiased estimator of ρ^* is to use regression slope estimation, or the cost's empirical mean under π_0 if the independence conditions stated in Lemma 3.3.3 are met. In our experiments, we follow this second strategy, assuming for simplicity that such independence holds.

Parametric Policies. In practice, the actions/contexts space is extremely large and directly optimizing the objective with respect to the policy (as a $\mathbb{R}^{|\mathcal{X}|\times K}$ matrix) is unreasonable. In such cases, policies are parametrized to drastically reduce the complexity of the problem. This usually breaks convexity, even in the simplest case of log-linear policies - that is, policies of the form $\pi_\theta(a|x) \propto \exp(\theta^T f(x, a))$, for $f(x, a)$ a given joint feature map. In this case, the objective becomes a negative sum of log-concave functions resulting in a non-convex optimization surface. Building on Roux (2017), one can bypass this non-convexity by constructing a *tight* convex upper bound of the original objective, relying on the following lemma :

Lemma 3.3.4 (Convex upper-bound for log-concave policies)**.** *Let π_θ be a log-concave (w.r.t θ) policy. For a given θ_0, let* $\text{Risk}_n^{up}(\pi_\theta)$ *be defined as :*

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\pi_{\theta_0}(a_i|x_i)}{\pi_0(a_i|x_i)}(1 + \log[\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_0}(a_i|x_i)}])c(a_i|x_i)$$

$\text{Risk}_n^{up}(\pi_\theta)$ *is a* convex *upper bound of the IPS risk. The closer θ_0 to θ, the tighter the upper bound, with equality at $\theta_0 = \theta$.*

We can use Lemma 3.3.4 to obtain a proxy of our initial objective, building on an iterative procedure that only uses convex losses throughout the whole optimization process. Once again, we can build a robust version of this estimator which can be efficiently optimized. Note that here, the robust estimator will also be convex w.r.t the parametrization θ as soon as the policy is log-concave.

3.4 Experiments

We here describe the experimental results, backing up the idea that an improved optimization landscape for policy optimization naturally leads to improved practical performances. We work with the four φ-divergence presented in Table 3.1. We employ the classical supervised to bandit conversion (Agarwal et al., 2014). Formally, denote $x \in \mathcal{X}$ a given input vector and $y \in \{0,1\}^L$ its label, and $\mathcal{D}^\star = \{(x_1, t_1), \dots, (x_m, t_m)\}$ a given multi-label dataset. We create the logging policy by training it on a fraction of $\mathcal{D}^\star$ with half of its labels shuffled[3]. We then create the logged interactions $\mathcal{D}_n$ by going repeating P times the following procedure: for every (x_i, t_i) in the supervised dataset, sample $a_i \sim \pi_0(x)$ and log the cost $c(a_i, x_i) = \|a_i - t_i\|_1$. Following Swaminathan and Joachims (2015a), we denote by P the *replay count*.

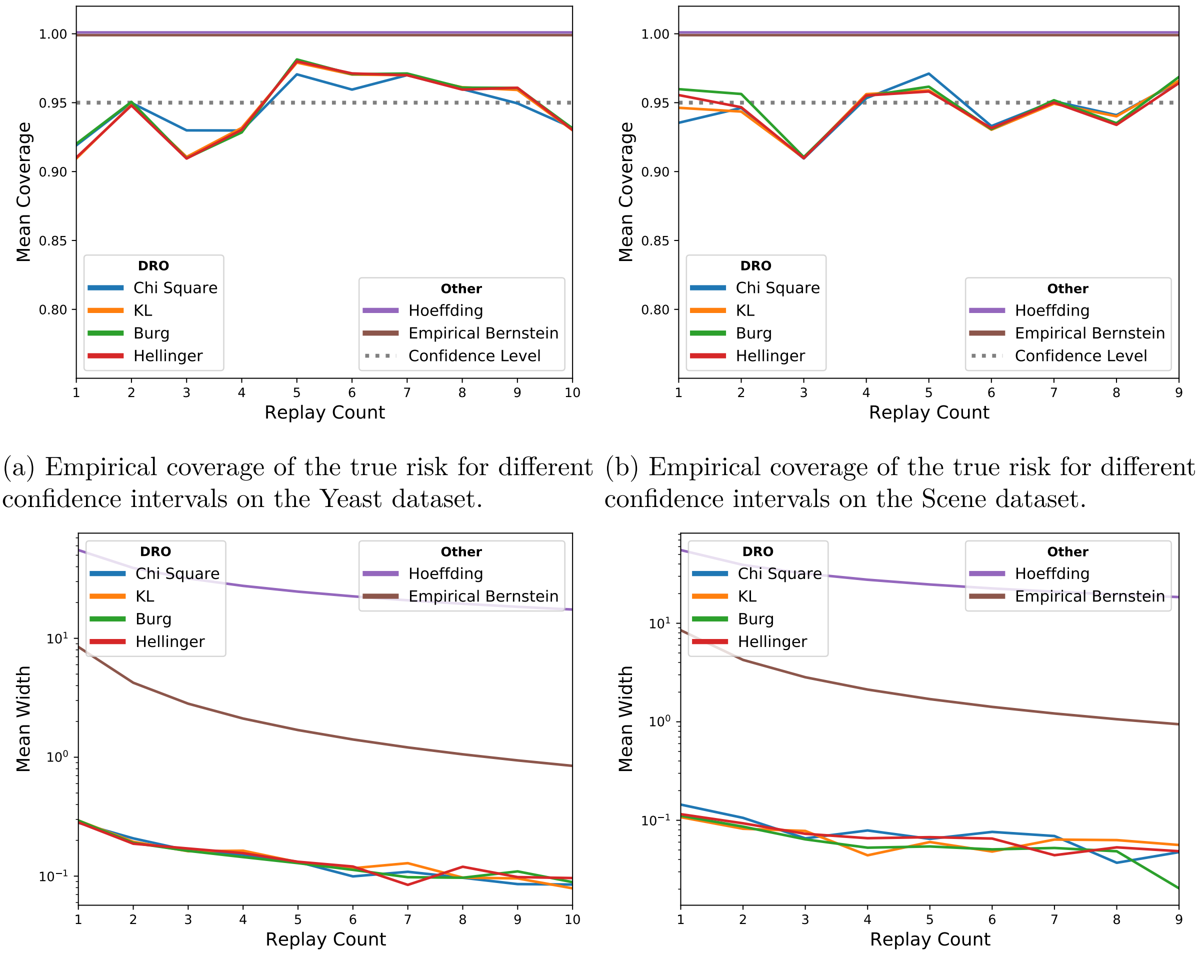
3.4.1 DRO Confidence Intervals

We start this experimental section by performing a *sanity check* on the (asymptotic) confidence intervals that we based our policy optimization method on. Formally, we evaluate the finite-time validity of Equation (3.4). Being asymptotic, we can safely expect DRO-based confidence intervals to be smaller than their finite-time counterparts - i.e confidence intervals based on Hoeffding or empirical Bernstein tail inequalities (see Thomas et al. (2015) for their application to policy evaluation). We however wish to check that they provide reasonable coverage in non-asymptotic regimes - that is, that the true value of the risk belongs to the interval often enough. To do so, we train a policy π on a subset of $\mathcal{D}^\star$ (randomly chosen, so the policy π trains on different examples than π_0) and evaluate the empirical mean coverage and of width of DRO-based intervals. In Figure 3.1, we present such results on two datasets: Yeast and Scene, taken from the LibSVM repository and standard for the policy optimization task (Swaminathan and Joachims, 2015a). The empirical coverage and width are reported for increasing values of the replay count P, or equivalently for increasing values of the historic data size n. The confidence level is set to $\delta = 0.95$ in all experiments. As expected, the asymptotic DRO-based confidence intervals are by orders of magnitude smaller than finite-time ones. Nevertheless, we observe that they provide almost exact $(1-\delta)$ coverage, and it is therefore safe to use them even in the small data regime. As a side comment, we observe that all four φ-divergence lead to very similar results.

3.4.2 Policy Optimization

We report here the results for policy optimization, for which we follow the experimental procedure of Swaminathan and Joachims (2015a). The supervised dataset $\mathcal{D}^\star$ is split into three parts (train, validation and test). A logging policy is trained on a random fraction (5%) of $\mathcal{D}_*$ with half of its labels shuffled and used to collect the history $\mathcal{H}_n$ by running it through the training data $P = 4$ times. For DRO-based algorithms, the validation set is not used, since no hyper-parameter needs to be tuned (we use the value recommended by the asymptotic analysis

[3] to avoid near optimal logging policies.



(a) Empirical coverage of the true risk for different confidence intervals on the Yeast dataset.

(b) Empirical coverage of the true risk for different confidence intervals on the Scene dataset.

(c) Mean width of the true risk for different confidence intervals on the Yeast dataset.

(d) Mean width of the true risk for different confidence intervals on the Scene dataset.

Figure 3.1: Finite-time evaluation (coverage and width) for asymptotic DRO-based confidence intervals.

for ε, with a fixed confidence level at $\delta = 0.05$). For POEM and its stochastic approximation, the parameter λ is selected by cross-validation on the validation data.

Clipped IPS. We report in Figure 3.2 the risks of the policy returned by the different algorithms trained using the clipped IPS estimator CIPS with M set to $\sqrt{n}$ recommended in Ionides (2008). We compare its performance to POEM and DRO on three multilabel LibSVM datasets: Scene, Yeast and Mediamill[4]. As in Swaminathan and Joachims (2015a), all policies are parametrized linearly, with a softmax output activation layer. We present results for both batch algorithms optimized by L-BFGS ($\mathcal{B}$) and their stochastic versions optimized by Adam for 10 epochs ($\mathcal{S}$). Results are averaged over 20 random repetitions. For batch algorithms, one can notice that DRO-based methods provide either similar or better empirical results than POEM on the considered datasets, while being hyper-parameter free (which is not the case of POEM). On the Yeast dataset, the improvement is quite significative for two of the four φ-divergence (KL and Hellinger). It seems however that there is no consistency in the relative performance of the different divergences. This can be troublesome in practice, as to the best of our knowl-

[4] preprocessed to reduce the number of actions

Algorithm	Scene	Yeast	Mediamill*
CIPS-$\mathcal{B}$	1.25 (0.10)	5.08 (0.13)	2.62 (0.13)
POEM-$\mathcal{B}$	1.21 (0.12)	5.00 (0.12)	2.48 (0.13)
DRO-$\mathcal{B}$-χ^2	1.25 (0.25)	4.96 (0.08)	2.42 (0.05)
DRO-$\mathcal{B}$-KL	1.23 (0.10)	4.77 (0.10)	2.44 (0.05)
DRO-$\mathcal{B}$-Burg	1.26 (0.23)	4.89 (0.12)	2.42 (0.05)
DRO-$\mathcal{B}$-Hellinger	1.21 (0.11)	4.75 (0.15)	2.43 (0.05)
CIPS-$\mathcal{S}$	1.29 (0.09)	5.05 (0.10)	2.52 (0.13)
POEM-$\mathcal{S}$	1.28 (0.09)	5.00 (0.11)	2.47 (0.10)
DRO-$\mathcal{S}$-χ^2	1.27 (0.09)	4.93 (0.09)	2.42 (0.10)
DRO-$\mathcal{S}$-KL	1.30 (0.09)	4.93 (0.10)	2.42 (0.08)
DRO-$\mathcal{S}$-Burg	1.27 (0.09)	4.96 (0.11)	2.38 (0.09)
DRO-$\mathcal{S}$-Hellinger	1.31 (0.08)	4.92 (0.08)	2.44 (0.06)

Figure 3.2: **Policy optimization results**: Performance of the trained policy on the test set.

Algorithm	Scene	Yeast	Mediamill*
POEM-$\mathcal{B}$	1.21 (0.12)	5.00 (0.12)	2.48 (0.13)
DRO-$\mathcal{B}$-div	1.20 (0.10)	4.80 (0.15)	2.42 (0.05)
POEM-$\mathcal{S}$	1.28 (0.09)	5.00 (0.11)	2.47 (0.10)
DRO-$\mathcal{S}$-div	1.30 (0.09)	4.95 (0.10)	2.40 (0.08)

Figure 3.3: **Policy optimization results**: Performance of DRO-div on the test set.

edge there is no obvious nor preferable choice of divergences given a dataset. A solution to this problem is to cross-validate this choice, either over a discrete or a continuous parametrization of the divergence considered here (such as the parameter of a Cressie-Read divergence). Finally, we note that the stochastic algorithms present the same trend, with DRO-based algorithms resulting similar or better performance and providing the first fully stochastic algorithm for CRM contrary to POEM-$\mathcal{S}$ that needs to load in memory the entire dataset at every epoch (*e.g.* every time an upper-bound on the true objective is constructed). Meaning that not only DRO-$\mathcal{S}$ provides better results on the experiments presented, it is significantly faster. Figure 3.3 provides the results of DRO-div, which uses the validation set to choose the best divergence to use among the 4 presented in Table 3.1. This is still more efficient than POEM as we cross validate 4 divergences instead of a grid over the values of λ. We can see that this approach gives similar results than POEM on the Scene dataset, and outperforms it on the two other datasets, choosing automatically the divergence based on its performance on the validation set.

Additive Control Variate. In this set of experiments, we call IPS-ACV, the robustified estimator derived from Lemma 3.3.3 with ρ being equal to the empirical mean of the cost under the logging policy. We expect that this estimator will not only behave better than clipped IPS, but it will also benefit from the CRM principle as it was observed in Swaminathan and Joachims (2015b) when using the SNIPS estimator. We report the results in Figure 3.4 of IPS-ACV, IPS-ACV-POEM, and IPS-ACV-DRO-div which uses the validation set to choose the best divergence. We can observe that IPS-ACV improves drastically over clipped IPS on the three datasets considered and benefits slightly from CRM and DRO, meaning that even if we can achieve better variance with an estimator, we might still need to rank the policies by the variance induced to derive a more robust objective to optimize. This behavior was also observed in the experiments conducted with the SNIPS estimator in Swaminathan and Joachims (2015b).

Algorithm	Scene	Yeast	Mediamill*
IPS-ACV-$\mathcal{B}$	0.78 (0.06)	4.20 (0.16)	2.37 (0.12)
IPS-ACV-POEM-$\mathcal{B}$	0.78 (0.06)	4.16 (0.16)	2.25 (0.11)
IPS-ACV-DRO-$\mathcal{B}$-div	0.78 (0.06)	4.16 (0.15)	2.20 (0.08)

Figure 3.4: **Policy optimization results**: Performance of the trained policies using the robustified estimator.

3.5 Related Work

The Offline contextual bandit problem has received increasing interest in the past years as its an important formulation to real-world decision problems. The first step is to define a good estimator to evaluate different policies offline. The simplest estimator for offline contextual bandits is the "Inverse Propensity Score" (IPS) approach which is unbiased, but suffers from high variance depending on the discrepancy between the logging policy and the policy we want to evaluate, a natural extension of this estimator is the clipped IPS (Bottou et al., 2013) that tries to trade off bias for variance. The Self-Normalized IPS (SNIPS) Swaminathan and Joachims (2015b) estimate uses the propensity weights as a multiplicative control variate resulting in a biased estimator but with a better mean squared error. Defining better estimators can also be achieved by building a model of the reward first to define a doubly robust estimator (Dudík et al., 2014; Su et al., 2020) or by switching (Wang et al., 2017) between a doubly robust estimator and direct application of a reward estimator to optimize mean square error. All of these methods focus on defining better estimators for policy evaluation, which makes its optimization more stable. Another line of work focus more on the learnability of the problem, the Counterfactual Risk Minimization principle Swaminathan and Joachims (2015a) provides variance sensitive upper bounds on the true risk using empirical bernstein inequalities, Faury et al. (2020) tackles the problem with distributional robustness and exposes its link with sample variance penalization and London and Sandler (2019) uses PAC-Bayes learning theory to have guarantees on the risk of the learned policy. A recent work identifies the optimization difficulty of such objectives Chen et al. (2019b) and defines convex surrogates resulting in better learned policies. The distributionally robust optimization framework has a rich literature (Duchi et al., 2021; Ben-Tal et al., 2013) that focuses mainly on the statistical learning aspects, Faury et al. (2020) introduced the idea of using this framework in the context of offline policy optimization, and we further investigate its use in this work to define objectives that are convex on the policy, that keeps the same statistical properties that sample variance penalization provides, and that are amenable to stochastic gradient descent, making the optimization problem scalable to large datasets. Concurrently to this work, DRO tools were investigated in Dai et al. (2020) to achieve confident policy evaluation in the offline Reinforcement Learning framework (Sutton and Barto, 2018). If we focused on both evaluation and learning aspects for CB based on asymptotic arguments, Dai et al. (2020) provide finite sample analysis, and generalize our evaluation approach to the RL setting with little focus on the learning aspect.

3.6 Conclusion

In this work, we leverage the Distributionally Robust Optimization framework to provide a computationally friendly alternative to the counterfactual risk minimization principle, keeping the same statistical properties and giving access to algorithms that are convex on the policy, faster than POEM and that can avoid hyperparameter tuning if we choose a divergence beforehand. The experiments conducted confirm that DRO can benefit any risk estimator. It provides asymptotic confidence intervals that are tight, have good coverage properties in finite-time, and that can be exploited to learn policies with improved performance.

Chapter 4

Offline Learning with PAC-Bayesian Theory

Abstract

This chapter introduces a new principled approach for off-policy learning in contextual bandits. Unlike previous work, our approach does not derive learning principles from intractable or loose bounds. We analyse the problem through the PAC-Bayesian lens, interpreting policies as mixtures of decision rules. This allows us to propose novel generalization bounds and provide tractable algorithms to optimize them. We prove that the derived bounds are tighter than their competitors, and can be optimized directly to confidently improve upon the logging policy *offline*. Our approach learns policies with guarantees, uses all available data, and does not require tuning additional hyperparameters on held-out sets. We demonstrate through extensive experiments the effectiveness of our approach in providing performance guarantees in practical scenarios.

Contents

4.1 Introduction

Online industrial systems encounter sequential decision problems as they interact with the environment and strive to improve based on the received feedback. The contextual bandit framework formalizes this mechanism, and proved valuable with applications in recommender systems (Valko et al., 2014) and clinical trials (Villar et al., 2015). It describes a game of repeated interactions between a system and an environment, where the latter reveals a context that the system interacts with, and receives a feedback in return. While the online solution to this problem involves strategies that find an optimal trade-off between exploration and exploitation to minimize the *regret* (Lattimore and Szepesvári, 2020), we are concerned with its offline formulation (Swaminathan and Joachims, 2015a), which is arguably better suited for real-life applications, where more control over the decision-maker, often coined *the policy*, is needed. The learning of the policy is performed offline, based on historical data, typically obtained by logging the interactions between an older version of the decision system and the environment. By leveraging this data, our goal is to discover new strategies of greater performance.

There are two main paths to address this learning problem. The *direct method* (Sakhi et al., 2020a; Jeunen and Goethals, 2021) attacks the problem by modelling the feedback and deriving a policy according to this model. This approach can be praised for its simplicity (Brandfonbrener et al., 2021), is well-studied in the offline setting (Nguyen-Tang et al., 2022) but will often suffer from a bias as the feedback received is complex and the efficiency of the method directly depends on our ability to understand the problem's structure. We will consider the second path of off-policy learning, or IPS: *inverse propensity scoring* (Horvitz and Thompson, 1952), where we learn the policy directly from the logged data after correcting its bias with importance sampling (Owen and Zhou, 2000). As these estimators (Bottou et al., 2013; Swaminathan and Joachims, 2015b) can suffer from a variance problem as we drift away from the logging policy, the literature gave birth to different learning principles (Swaminathan and Joachims, 2015a; Ma et al., 2019; London and Sandler, 2019; Faury et al., 2020) motivating penalizations toward the logging policy. These principles are inspired by generalization bounds, but introduce a hyperparameter λ to either replace intractable quantities (Swaminathan and Joachims, 2015a) or to tighten a potentially vacuous bound (London and Sandler, 2019). These approaches require tuning λ on a held-out set and sometimes fail at improving the previous decision system (London and Sandler, 2019; Chen et al., 2019b). In this chapter, we analyse off-policy learning from the PAC-Bayesian perspective (McAllester, 1998; Catoni, 2007). We aim at introducing a novel, theoretically-grounded approach, based on the direct optimization of newly derived tight generalization bounds, to obtain guaranteed improvement of the previous system ***offline***, without the need for held-out sets nor hyperparameter tuning. We show that our approach is perfectly suited to this framework, as it naturally incorporates information about the old decision system and can confidently improve it.

4.2 Preliminaries

4.2.1 Setting

We use $x \in \mathcal{X}$ to denote a context and $a \in \mathcal{A} = [K]$ an action, where K denotes the number of available actions. For a context x, each action is associated with a cost $c(x, a) \in [-1, 0]$, with the convention that better actions have smaller cost. The cost function c is unknown. Our decision system is represented by its policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$ which given $x \in \mathcal{X}$, defines a probability distribution over the discrete action space $\mathcal{A}$ of size K. Assuming that the contexts are stochastic and follow an unknown distribution ν, we define the *risk* of the policy π as the expected cost one suffers when playing actions according to π:

$$R(\pi) = \mathbb{E}_{x \sim \nu, a \sim \pi(\cdot|x)} \left[c(x,a) \right].$$

The learning problem is to find a policy π which minimizes the risk. This risk can be naively estimated by deploying the policy online and gathering enough interactions to construct an accurate estimate. Unfortunately, we do not have this luxury in most real-world problems, as the cost of deploying bad policies can be extremely high. We can obtain instead an estimate by exploiting the logged interactions collected by the previous system. Indeed, the previous system is represented by a *logging policy* π_0 (e.g a previous version of a recommender system that we are trying to improve), which gathered interaction data of the following form:

$$\mathcal{D}_n = \{x_i, a_i \sim \pi_0(\cdot|x_i), c_i\}_{i \in [n]}, \quad \text{with } c_i = c(x_i, a_i).$$

Given this data, one can build various estimators, with the clipped IPS Bottou et al. (2013) the most commonly used. It is constructed based on a clipping of the importance weights or the logging propensities to mitigate variance issues (Bottou et al., 2013). We are more interested in clipping the logging probabilities, as we need objectives that are linear in the policy π for our study. The cIPS estimator is given by:

$$\hat{R}_n^{\tau}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\max\{\pi_0(a_i|x_i), \tau\}} c_i \tag{4.1}$$

with $\tau \in [0,1]$ being the clipping factor. Choosing $\tau \ll 1$ reduces the bias of cIPS. We recover the classical IPS estimator (Horvitz and Thompson, 1952) (unbiased under mild conditions) by taking $\tau = 0$. Another estimator with better statistical properties is the doubly robust estimator (Ben-Tal et al., 2013), which uses the importance weights as control variates to reduce further the variance of the cIPS estimators. This estimator is asymptotically optimal (Farajtabar et al., 2018) (in terms of variance) amongst the class of unbiased and consistent off-policy estimators.

We consider a simplified version of this estimator, which replaces the use of a model $\hat{c}$ of the cost by one parameter $\xi \in [-1, 0]$ that can be chosen freely. We define the control variate clipped IPS, or cvcIPS as follows:

$$\hat{R}_n^{\tau,\xi}(\pi) = \xi + \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\max\{\pi_0(a_i|x_i), \tau\}} (c_i - \xi). \tag{4.2}$$

The cvcIPS estimator can be seen as a special case of the doubly robust estimator when the cost model $\hat{c} = \xi$ is constant and $\tau = 0$. cIPS is recovered by setting $\xi = 0$. This simple estimator is deeply connected to the *SNIPS* estimator (Swaminathan and Joachims, 2015b) and was shown to be more suited to off-policy learning as it mitigates the problem of propensity overfitting (Joachims et al., 2018).

4.2.2 Related Work: Learning Principles

The literature so far has focused on deriving new principles to learn policies with good online performance. The first line of work in this direction is CRM: Counterfactual Risk minimization Swaminathan and Joachims (2015a) which adopted SVP: Sample Variance Penalization Maurer and Pontil (2009) to favour policies with small empirical risk and controlled variance. The intuition behind it is that the variance of cIPS depends on the disparity between π and π_0, making the estimator unreliable when π drifts away from π_0. The analysis focused on the cIPS estimator and used uniform bounds based on empirical Bernstein inequalities Maurer and Pontil (2009), where intractable quantities were replaced by a tuning parameter λ, giving the following learning objective:

$$\arg\min_{\pi} \left\{ \hat{R}^{\tau}_n(\pi) + \lambda \sqrt{\frac{\hat{V}_n(\pi)}{n}} \right\} \tag{4.3}$$

with $\hat{V}_n(\pi)$ the empirical variance of the cIPS estimator. A majorisation-minimisation algorithm was provided in Swaminathan and Joachims (2015a) to solve Equation (4.3) for parametrized softmax policies. In the same spirit, Faury et al. (2020); Sakhi et al. (2020b) generalize SVP using the distributional robustness framework, showing that the CRM principle can be retrieved with a particular choice of the divergence and provide asymptotic coverage results of the true risk. Their objectives are competitive with SVP while providing simple ways to scale its optimization to large datasets.

Another line of research, closer to our work, uses PAC-Bayesian bounds to derive learning objectives in the same fashion as Swaminathan and Joachims (2015a). Indeed, London and Sandler (2019) introduce the Bayesian CRM, motivating the use of L_2 regularization towards the parameter θ_0 of the logging policy π_0. The analysis uses McAllester (2003)'s bound, is conducted on the cIPS estimator and controls the L_2 norm by a hyperparameter λ, giving the following learning objective for parametrized softmax policies:

$$\arg\min_{\theta} \left\{ \hat{R}^{\tau}_n(\pi_\theta) + \lambda ||\theta - \theta_0||^2 \right\}. \tag{4.4}$$

London and Sandler (2019) minimize a convex upper-bound of objective (4.4) (by taking a log transform of the policy) which is amenable to stochastic optimization, giving better results than (4.3) while scaling better to the size of the dataset.

Limitations. The principles found in the literature are mainly inspired by generalization bounds. However, the bounds from where these principles are derived either depend on intractable quantities Swaminathan and Joachims (2015a) or are not tight enough to be used as-is London and Sandler (2019). For example, Swaminathan and Joachims (2015a) derive a generalisation bound (see Theorem 1 in Swaminathan and Joachims (2015a)) for offline policy learning using the notion of covering number. This introduces the complexity measure $\mathcal{Q}_\Pi(n, \gamma)$ that cannot be computed (even for simple policy classes) making their bound intractable. This forces the introduction of a hyperparameter λ that needs further tuning. Unfortunately, this approach suffers from numerous problems:

- **No Theoretical Guarantees.** Introducing the hyperparameter λ in Equations (4.3) and (4.4) gives tractable objectives, but loses the theoretical guarantees given by the initial bounds. These objectives do not necessarily cover the true risk, and optimizing them can lead to policies worse than the logging π_0. Empirical evidence can be found in Chen et al. (2019b); London and Sandler (2019) where the SVP principle in Equation (4.3) fails to improve on π_0.

- **Inconsistent Strategy.** These principles were first introduced to mitigate the suboptimality of learning with off-policy estimators, deemed untrustworthy for their potential high variance. The strategy minimizes the objectives for different values of $\{\lambda_1, ..., \lambda_m\}$, generating a set of policy candidates $\{\pi_{\lambda_1}, ..., \pi_{\lambda_m}\}$, from which we select the best policy π_{λ_*} according to the same untrustworthy, high variance estimators on a held-out set. This makes the selection strategy used inconsistent with what these principles are claiming to solve.

- **Tuning requires additional care.** Tuning λ needs to be done on a held-out set. This means that we need to train multiple policies (computational burden) on a fraction of the data (data inefficiency), and select the best policy among the candidates using off-policy estimators (variance problem) on the held-out set.

In this chapter, we derive a coherent principle, that learns policies using all the available data and provides guarantees on their performance, without the introduction of new hyperparameters.

4.2.3 Learning With Guaranteed Improvements

Our first concern in most applications is to improve upon the actual system π_0. As $\mathcal{D}_n$ is collected by π_0, we have access to $R(\pi_0)$[1]. Given a new policy π, we want to be confident that the improvement $\mathcal{I}(\pi, \pi_0) = R(\pi_0) - R(\pi)$ is positive before deployment.

Let us suppose that we are restricted to a class of policies Π, and have access to a generalization bound that gives the following result with high probability over draws of $\mathcal{D}_n$:

$$R(\pi) \leq \mathcal{UB}_n(\pi) \quad \forall \pi \in \Pi.$$

with $\mathcal{UB}_n$ an empirical upper bound that depends on $\mathcal{D}_n$. For any π, we define the guaranteed improvement:

$$\mathcal{GI}_{\mathcal{UB}_n}(\pi, \pi_0) = R(\pi_0) - \mathcal{UB}_n(\pi).$$

We can be sure of improving $R(\pi_0)$ offline if we manage to find $\pi \in \Pi$ that achieves $\mathcal{GI}_{\mathcal{UB}_n}(\pi, \pi_0) > 0$ as the following result will hold with high probability:

$$\mathcal{I}(\pi, \pi_0) \geq \mathcal{GI}_{\mathcal{UB}_n}(\pi, \pi_0) > 0.$$

To obtain such a policy, we look for the minimizer of $\mathcal{UB}_n$ over the class of policies Π as:

$$\pi^*_{\mathcal{UB}_n} \in \arg\min_{\pi \in \Pi} \mathcal{UB}_n(\pi) = \arg\max_{\pi \in \Pi} \mathcal{GI}_{\mathcal{UB}_n}(\pi, \pi_0).$$

We define the best guaranteed risk and the best guaranteed improvement follows:

$$\mathcal{GR}^*_{\mathcal{UB}_n} = \mathcal{UB}_n(\pi^*_{\mathcal{UB}_n})$$
$$\mathcal{GI}^*_{\mathcal{UB}_n}(\pi_0) = R(\pi_0) - \mathcal{GR}^*_{\mathcal{UB}_n}.$$

A *theoretically-grounded* strategy to improve π_0 will be to deploy $\pi^*_{\mathcal{UB}_n}$ if we obtain a positive guaranteed improvement $\mathcal{GI}^*_{\mathcal{UB}_n}(\pi_0) > 0$, otherwise continue collecting data with the current system π_0.

This strategy will always produce policies that are at least as good as π_0, optimizes directly a bound over all data and does not require held-out sets nor new hyperparameters. However, the tightness of the bounds $\mathcal{UB}_n$ will play an important role. Indeed, If we fix $\mathcal{D}_n$ and π_0, $\mathcal{GI}^*_{\mathcal{UB}_n}(\pi_0)$ will only depend on the minimum of $\mathcal{UB}_n$, motivating the derivation of bounds that are tractable and tight enough to achieve the smallest minimum possible.

In this regard, we opt for the PAC-Bayesian framework to tackle this problem as it is proven to give tractable, non-vacuous bounds even in difficult settings (Dziugaite and Roy, 2017). Its paradigm also fits our application, as we can incorporate information about the previous system π_0 in the form of a prior; see Alquier (2021) for a recent review.

[1] up to a small $\mathcal{O}(1/\sqrt{n})$ approximation error.

Contributions. We advocate for a theoretically grounded strategy that uses generalization bound to improve π_0 with guarantees. So far, the existing bounds are either intractable (Swaminathan and Joachims, 2015a) or can be proven to be suboptimal (London and Sandler, 2019). In this work,

- we derive new, tractable and tight generalization bounds using the PAC-Bayesian framework. These bounds are fully tractable unlike Swaminathan and Joachims (2015a)'s bound and are tighter than London and Sandler (2019)'s bound.

- we provide a way to optimize our bounds over a particular class of policies and show empirically that they can guarantee improvement over π_0 in practical scenarios.

4.3 Motivating PAC-Bayesian tools

As previously discussed, in the contextual bandit setting, we seek a policy that minimizes the expected cost:

$$R(\pi) = \mathbb{E}_{x\sim\nu, a\sim\pi(\cdot|x)}\left[c(x,a)\right].$$

The minimizer of this objective over the unrestricted space of policies is a deterministic decision rule defined by:

$$\forall x, a \quad \pi^*(a|x) = \mathbb{1}[\operatorname*{argmin}_{a'} c(x,a') = a].$$

Given a context x, the solution will always choose the action that has the minimum cost. However, as the function c is generally unknown, we instead learn a parametric score function $f_\theta \in \mathcal{F}_\Theta = \{f_\theta : \mathcal{X} \times [K] \to \mathbb{R}, \theta \in \Theta\}$ that encodes the action's relevance to a context x. Given a function f_θ, we define the decision rule d_θ by:

$$d_\theta(a|x) = \mathbb{1}[\operatorname*{argmax}_{a'} f_\theta(x,a') = a].$$

These parametric decisions rules will be the building blocks of our analysis. We view stochastic policies as smoothed decision rules, with smoothness induced by distributions over the space of score functions $\mathcal{F}_\Theta$. Given a context x, instead of sampling an action a directly from a distribution on the action set, we sample a function f_θ from a distribution over $\mathcal{F}_\Theta$ and compute the action as $a = \operatorname{argmax}_{a'} f_\theta(x,a')$. With this interpretation, for any distribution Q on $\mathcal{F}_\Theta$, the probability of an action, $a \in \mathcal{A}$, given a context $x \in \mathcal{X}$, is defined as the expected value of d_θ over Q, that is:

$$\begin{aligned}\pi_Q(a|x) &= \mathbb{E}_{\theta\sim Q}\left[d_\theta(a|x)\right] \\ &= \mathbb{P}_Q\left(\operatorname*{argmax}_{a'} f_\theta(x,a') = a\right).\end{aligned}$$

Policies as mixtures of decision rules. This perspective on policies was introduced in Seldin et al. (2011) and later developed in London and Sandler (2019). Constructing policies as *mixtures of deterministic decision rules* does not restrict the class of policies our study applies to. Indeed, if the family $\mathcal{F}_\Theta$ is rich enough (e.g, neural networks), we give the following theorem that proves that any policy π can be written as a mixture of deterministic policies.

Theorem 4.3.1. *Let us fix a policy π. Then there is a probability distribution Q_π on the set of all functions $f : \mathcal{X} \times [K] \to \{0,1\}$ such that*

$$\forall x, a \quad \pi(a|x) = \mathbb{E}_{f \sim Q_\pi}\left[\mathbb{1}\left[\operatorname*{argmax}_{a'} f(x,a') = a\right]\right].$$

A formal proof of Theorem 4.3.1 is given in Appendix 4.8.1. This means that adopting this perspective on policies does not narrow the scope of our study. For a policy π_Q defined by a distribution Q over $\mathcal{F}_\Theta$, we observe that by linearity, its true risk can be written as:

$$R(\pi_Q) = \mathbb{E}_{\theta \sim Q}[R(d_\theta)].$$

Similarly, clipping the logging propensities in our empirical estimators allows us to obtain a linear estimator in π. For instance, we can estimate empirically the risk of the policy π_Q with cvcIPS (as it generalizes cIPS) and obtain:

$$\hat{R}_n^{\tau,\xi}(\pi_Q) = \mathbb{E}_{\theta \sim Q}[\hat{R}_n^{\tau,\xi}(d_\theta)].$$

By linearity, one can see that both the true and empirical risk of a policy π_Q can also be interpreted as the average risk of decision rules drawn from the distribution Q. This duality is at the heart of our analysis and paves the way nicely to the PAC-Bayesian framework, which studies generalization properties of the average risk of randomized predictors Alquier (2021). If we fix a reference distribution P over $\mathcal{F}_\Theta$ and define the KL divergence from P to Q as:

$$\mathcal{KL}(Q||P) = \begin{cases} \int \ln\left\{\frac{dQ}{dP}\right\} dQ & \text{if } Q \text{ is } P\text{-continuous}, \\ +\infty & \text{otherwise}, \end{cases}$$

we can construct with the help of PAC-Bayesian tools bounds holding for the average risk of decision rules over any distribution Q;

$$\mathbb{E}_{\theta \sim Q}[R(d_\theta)] \leq \mathbb{E}_{\theta \sim Q}[\hat{R}_n^{\tau,\xi}(d_\theta)] + \mathcal{O}\left(\mathcal{KL}(Q||P)\right).$$

Our objective will be to find tight generalisation bounds of this form as this construction, coupled with the linearity of our objective and estimator, allows us to obtain tight bounds holding for any policy π_Q;

$$R(\pi_Q) \leq \hat{R}_n^{\tau,\xi}(\pi_Q) + \mathcal{O}\left(\mathcal{KL}(Q||P)\right).$$

The PAC-Bayesian Paradigm. Before we dive deeper into the analysis, we want to emphasize the similarities between the PAC-Bayesian paradigm and the offline contextual bandit problem. This learning framework proceeds as follows: Given a class of functions $\mathcal{F}_\Theta$, we fix a prior (reference distribution) P on $\mathcal{F}_\Theta$ before seeing the data, then, we receive some data $\mathcal{D}_n$ which help us learn a better distribution Q over $\mathcal{F}_\Theta$ than our reference P. With the previous perspective on policies, the prior P, even if it can be any data-free distribution, will be our logging policy (i.e. $\pi_0 = \pi_P$), and we will use the data $\mathcal{D}_n$ to learn distribution Q, thus a new policy π_Q that improves the logging policy π_0.

4.4 PAC-Bayesian Analysis

4.4.1 Bounds for clipped IPS

The clipped IPS estimator (Bottou et al., 2013) is often studied for offline policy learning (Swaminathan and Joachims, 2015a; London and Sandler, 2019) as it is easy to analyse, and have a negative bias (once the cost is negative) facilitating the derivation of learning bounds.

London and Sandler (2019) adapted McAllester (2003)'s bound to derive their learning objective. We state a slightly tighter version in Proposition 4.4.1 for the cIPS estimator. The proof of this bound cannot be adapted naively to the cvcIPS estimator, because once $\xi \neq 0$, the bias of the estimator, an intractable quantity that depends on the unknown distribution ν, is no longer negative and needs to be incorporated in the bound, making the bound itself intractable.

Proposition 4.4.1. *Given a prior P on $\mathcal{F}_\Theta$, $\tau \in (0,1]$, $\delta \in (0,1]$, the following bound holds with probability at least $1-\delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:*

$$R(\pi_Q) \leq \hat{R}_n^\tau(\pi_Q) + \frac{2(\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{\tau n} + \sqrt{\frac{2[\hat{R}_n^\tau(\pi_Q) + \frac{1}{\tau}](\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{\tau n}}.$$

The upper bound stated in the previous proposition will be denoted by $\mathcal{LS}_n^{P,\delta,\tau}(\pi_Q)$. When there is no ambiguity, we will also drop P, δ, τ (all fixed) and only use $\mathcal{LS}_n(\pi_Q)$.

McAllester (2003)'s bound can give tight results in the $[0,1]$-bounded loss case when the empirical risk is close to 0 as one obtains fast convergence rates in $O(1/n)$. However, its use in the case of offline contextual bandits is far from being optimal. Indeed, to achieve a fast rate in this context, one needs $\hat{R}_n^\tau(\pi_Q) + \frac{1}{\tau} \approx 0$. This is hardly achievable in practice especially when n is large and $\tau \ll 1$. To defend our claim, let us suppose that for each context x, there is one optimal action a_x^* for which $c(x, a_x^*) = -1$ and it is 0 otherwise. Let us write down the clipped IPS:

$$\hat{R}_n^\tau(\pi) = \frac{1}{n}\sum_{i=1}^n \frac{\pi(a_i|x_i)}{\max\{\pi_0(a_i|x_i), \tau\}} c_i \geq -\frac{1}{\tau}.$$

To get equality, we need:

$$\forall i \in [n], \quad c_i = -1, \quad \pi_0(a_i|x_i) \leq \tau, \quad \pi(a_i|x_i) = 1.$$

If n is large, the first condition on the costs means that π_0 is near optimal and the played actions a_i are optimal. For this, we get that $\forall i \in [n], \pi_0(a_i|x_i) \approx 1$. This, combined with the second condition on π_0 gives that $\tau \approx 1$. In practice, π_0 is never the optimal policy and $\tau \ll 1$ which makes the fast rate condition $\hat{R}_n^\tau(\pi) + \frac{1}{\tau} \approx 0$ unachievable. In the majority of scenarios, as we penalize π_Q to stay close to π_0 through the KL divergence, we will have $\hat{R}_n^\tau(\pi_Q) \in [-1, 0]$, thus $\hat{R}_n^\tau(\pi_Q) + \frac{1}{\tau} \approx \frac{1}{\tau}$, giving a limiting behaviour:

$$\mathcal{LS}_n(\pi_Q) = \hat{R}_n^\tau(\pi_Q) + \mathcal{O}\left(\frac{1}{\tau}\sqrt{\frac{\mathcal{KL}(Q||P)}{n}}\right).$$

If we want to get tighter results, we need to look for bounds with better dependencies on τ and n. In our pursuit of a tighter bound, we derive the following result:

Proposition 4.4.2. *Given a prior P on $\mathcal{F}_\Theta$, $\tau \in (0,1]$, $\delta \in (0,1]$, the following bound holds with probability at least $1-\delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:*

$$R(\pi_Q) \leq \min_{\lambda > 0}\left\{\frac{1 - \exp\left(-\tau\lambda\hat{R}_n^\tau(\pi_Q) - \frac{1}{n}\left[\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}\right]\right)}{\tau(e^\lambda - 1)}\right\}.$$

We will denote by $\mathcal{C}_n^{P,\delta,\tau}(\pi_Q)$ the upper bound stated in this proposition. When there is no ambiguity, we will also drop P, δ, τ (all fixed) and only use $\mathcal{C}_n(\pi_Q)$.

This is a direct application of Catoni (2007)'s bound to the bounded loss cIPS while exploiting the fact that its bias is negative. A full derivation can be found in Appendix 4.8.2. Note that Proposition 4.4.2 cannot be applied to the cvcIPS estimator ($\xi \neq 0$) as its bias is non-negative and intractable. To be able to measure the tightness of this bound, we estimate its limiting behaviour to understand its dependency on τ and n. We derive in Appendix 4.8.3 the following result:

$$\mathcal{C}_n(\pi_Q) = \hat{R}_n^\tau(\pi_Q) + \mathcal{O}\left(\frac{\mathcal{KL}(Q||P)}{\tau n}\right).$$

This shows that $\mathcal{C}_n$ has a better dependency on n compared to $\mathcal{LS}_n$. Actually, we can prove that $\mathcal{C}_n$ will always give tighter results, as the next theorem states that it is smaller than $\mathcal{LS}_n$ in all scenarios.

Theorem 4.4.1. *For any $\mathcal{D}_n \sim (\mu, \pi_0)^n$, any distributions P, Q, any $\tau \in (0,1], \delta \in (0,1]$, we have:*

$$\mathcal{C}_n^{P,\delta,\tau}(\pi_Q) \leq \mathcal{LS}_n^{P,\delta,\tau}(\pi_Q).$$

One can refer to Appendix 4.8.4 for the full proof. This result confirms that the bound given by Proposition 4.4.2 is *theoretically* tighter than the bound in Proposition 4.4.1, making $\mathcal{LS}_n$ unusable if we seek tight guarantees on $R(\pi_Q)$.

4.4.2 Going beyond clipped IPS

The cvcIPS estimator in Equation (4.2) generalizes cIPS, and can behave better as it achieves improved variance with a well-chosen ξ. To study its learning properties, we derive a **novel** *Bernstein-type* PAC-Bayesian bound that holds for the cvcIPS estimator. Let g be the function

$$g : u \to \frac{\exp(u) - 1 - u}{u^2}.$$

We define the conditional bias $\mathcal{B}_n^\tau$ and the conditional second moment $\mathcal{V}_n^\tau$:

- $\mathcal{B}_n^\tau(\pi) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\pi(.|x_i)}\left[\mathbb{1}[\pi_0(a|x_i) < \tau]\left(1 - \frac{\pi_0(a|x_i)}{\tau}\right)\right]$
- $\mathcal{V}_n^\tau(\pi) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\pi(.|x_i)}\left[\frac{\pi_0(a|x_i)}{\max\{\pi_0(a|x_i), \tau\}^2}\right].$

With these definitions, we can state our proposition:

Proposition 4.4.3. *Given a prior P on $\mathcal{F}_\Theta$, $\xi \in [-1, 0], \tau \in (0, 1]$, $\delta \in (0, 1]$ and a set of strictly positive scalars $\Lambda = \{\lambda_i\}_{i \in [n_\Lambda]}$. The following bound holds with probability at least $1 - \delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:*

$$R(\pi_Q) \leq \hat{R}_n^{\tau,\xi}(\pi_Q) - \xi\mathcal{B}_n^\tau(\pi_Q) + \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{4\sqrt{n}}{\delta}}{2n}}$$
$$+ \min_{\lambda \in \Lambda}\left\{\frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}}{\lambda n} + \lambda l_\xi g\left(\lambda b_\xi\right)\mathcal{V}_n^\tau(\pi_Q)\right\}$$

with $l_\xi = \max\left[\xi^2, (1+\xi)^2\right]$, $b_\xi = (1+\xi)/\tau - \xi$.

The choice of Λ as well as the full proof of a more general version of Proposition 4.4.3 can be found in Appendix 4.8.5. The upper bound given by Proposition 4.4.3 (with P, τ, δ fixed) will be denoted by $\mathcal{CBB}_n^{\xi}(\pi_Q)$.

Proposition 4.4.3 covers the general cvcIPS estimator ($\xi \neq 0$) and its objective is decomposable into a sum, making it amenable to stochastic first order optimization (Robbins and Monro, 1951a). However, minimizing it requires access to the logging policy π_0. This is reasonable as π_0 represents the currently deployed decision system, which we want to improve. In the limit, the bound is estimated to behave like:

$$\mathcal{CBB}_n^{\xi}(\pi_Q) = \hat{R}_n^{\tau,\xi}(\pi_Q) - \xi \mathcal{B}_n^{\tau}(\pi_Q) + \mathcal{O}\left(\left(\frac{1}{2\sqrt{2}} + \sqrt{l_\xi \mathcal{V}_n^{\tau}(\pi_Q)}\right)\sqrt{\frac{\mathcal{KL}(Q||P)}{n}}\right)$$

A derivation of this result can be found in Appendix 4.8.5. We have $\mathcal{V}_n^{\tau} \leq 1/\tau$ and expect this bound to give tight results when $\mathcal{V}_n^{\tau}(\pi_Q) \ll 1/\tau$. However, once π_0 is uniform and $\tau = 1/K$, we can never have the previous condition as:

$$\forall \pi, \quad \mathcal{V}_n^{\tau}(\pi) = 1/\tau.$$

This means that the worst regime for $\mathcal{CBB}_n^{\xi}$ is when π_0 is uniform, and even in that case, this bound should be tighter than $\mathcal{LS}_n$ as it has a better dependency on τ. We can also get an intuition about the impact of ξ, in particular:

- $\xi = 0$ recovers cIPS. This estimator has the best dependency on the bias (it nullifies the impact of $\mathcal{B}_n^{\tau}(\pi_Q)$) and the worst dependency on $\mathcal{V}_n^{\tau}(\pi_Q)$ as $l_\xi = 1$.

- $\xi = -0.5$ obtains the best dependency on $\mathcal{V}_n^{\tau}(\pi_Q)$ as l_ξ reaches its minimum and make the bound suffer only half the bias $\mathcal{B}_n^{\tau}(\pi_Q)$.

- $\xi = -1$ in the other hand, has both the worst dependencies on the bias and the variance, and can be considered a bad choice. Actually, this value of ξ shifts the costs to be always positive ($\forall i, c_i - \xi \geq 0$), which is known to make the off-policy risk estimators not suited for policy learning (Swaminathan and Joachims, 2015a, Section 4.1).

These observations point to the importance of the choice of ξ, which can drastically change the behaviour of the bound. We will empirically study the impact of two candidate values of $\xi \in \{0, -0.5\}$ on the tightness of the bound.

4.5 Restricting the Space of Policies

Our PAC-Bayesian bounds hold for any policy π_Q. However, to obtain policies of practical use, we ask for some desired properties that are summarized in the points below.

Sampling. Being able to efficiently sample actions from our policy is crucial, as the decisions taken by our online system boil down to sampling. For a given context x, we have:

$$a \sim \pi_Q(\cdot|x) \iff a = \operatorname*{argmax}_{a'} f_\theta(x, a'),\ \theta \sim Q.$$

The complexity of sampling from π_Q depends on the difficulty of sampling from Q.

Computing propensities. Computing propensities for a given pair (x, a) is essential for off-policy evaluation. A generic estimate can be obtained by:

$$\hat{\pi}_Q^{\text{naive}}(a|x) = \frac{1}{S}\sum_{i=1}^{S} d_{\theta_i}(a|x)$$

with $\{\theta_i\}_{i=1}^S$ samples from Q. This estimator will behave badly once we deal with large action spaces and/or high-dimensional distributions parameters. Ideally, we would like to exploit the family of distributions Q considered and the form of the function f_θ to come up with a better behaved estimator for the propensities.

Numerical optimization. If we restrict our study to a parametric family $\mathcal{Q}_\Psi = \{Q_\psi, \psi \in \Psi\}$, computing gradients will be essential to minimising the bounds. For a given pair (x, a), we can compute for any parameterized distribution Q_ψ the score function gradient estimator (Williams, 1992) of $\pi_{Q_\psi}(a|x)$:

$$\begin{aligned}\nabla_\psi \pi_{Q_\psi}(a|x) &= \nabla_\psi \mathbb{E}_{\theta\sim Q_\psi}[d_\theta(a|x)] \\ &= \mathbb{E}_{\theta\sim Q_\psi}[d_\theta(a|x)\nabla_\psi \log Q_\psi(\theta)].\end{aligned}$$

This gradient suffers from a variance problem (Xu et al., 2019) and we might need to choose a specific family of distributions $\mathcal{Q}_\Psi$, or/and specify a form of f_θ to obtain $\psi \to \pi_{Q_\psi}(a|x)$ with better behaved gradients.

London and Sandler (2019) restricted their study to Mixed Logit policies (Hensher and Greene, 2003). These policies are easy to sample from, and have easy to compute propensities and gradients. However, their learning properties are demonstrated to be suboptimal (Mei et al., 2020a). To this end, we adopt another class of policies that we deem better behaved for our objective. We discuss the reasons behind this choice in detail in Appendix 4.8.7.

4.5.1 Linear Independent Gaussian Policies

As mentioned previously, even if our analysis is valid for all distributions Q and any form of f_θ, we need to restrict our space to obtain practical policies. We restrict f_θ to:

$$\forall x, a \quad f_\theta(x, a) = \phi(x)^T\theta_a \tag{4.5}$$

with ϕ a fixed transform[2] over the contexts. This form of f_θ is widely used in this context (Faury et al., 2020; Swaminathan and Joachims, 2015a). This results in a parameter θ of dimension $d = p \times K$ with p the dimension of the features $\phi(x)$ and K the number of actions. We also restrict the family of distributions $\mathcal{Q}_{d+1} = \{Q_{\boldsymbol{\mu},\sigma} = \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_d), \boldsymbol{\mu} \in \mathbb{R}^d, \sigma > 0\}$ to independent Gaussians with shared scale. With these choices of f_θ and $\mathcal{Q}$, the induced $\pi_{\boldsymbol{\mu},\sigma}$, that we call **LIG: Linear Independent Gaussian** policies, will provide fast sampling and easily computable propensities and gradients. Indeed, sampling from $\pi_{\boldsymbol{\mu},\sigma}$ will reduce to sampling from a normal distribution $\theta \sim Q_{\boldsymbol{\mu},\sigma}$ and computing $a = \operatorname{argmax}_{a'} f_\theta(x, a')$. When it comes to estimating the propensity of a given x, we can suggest another expression of $\pi_{\boldsymbol{\mu},\sigma}(a|x)$ that reduces the computation to a one dimensional integral:

$$\begin{aligned}\pi_{\boldsymbol{\mu},\sigma}(a|x) &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}\left[\prod_{a'\neq a}\Phi\left(\epsilon + \frac{\phi(x)^T(\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma||\phi(x)||}\right)\right] \\ &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}\left[G_{\boldsymbol{\mu},\sigma}(\epsilon, a, x)\right],\end{aligned} \tag{4.6}$$

[2] The analysis can be naturally extended to the more general case where ϕ_ψ is a parameterized neural network that we learn.

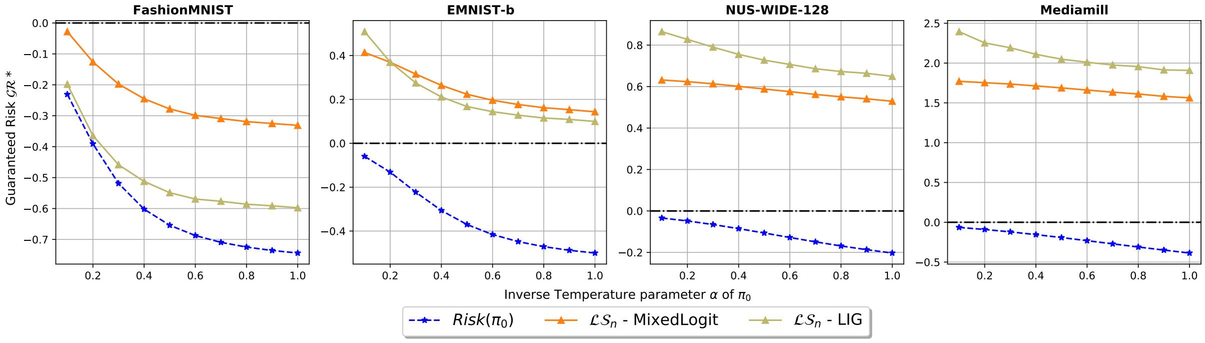


Figure 4.1: The Guaranteed Risk given by $\mathcal{LS}_n$ optimized over **Mixed Logit** and **LIG** policy classes while changing π_0. The $\mathcal{LS}_n$ bound fails to guarantee improvement ($\mathcal{GR}^*_{\mathcal{LS}_n} > R(\pi_0)$) in all scenarios considered.

with Φ the cumulative distribution function of the standard normal. See Appendix 4.8.6 for a full derivation. The computation of $\pi_{\boldsymbol{\mu},\sigma}(a|x)$ becomes easier as one dimensional standard normal integrals can be well approximated. The gradient can also be derived from this new expression:

$$\nabla_{\boldsymbol{\mu},\sigma}\pi_{\boldsymbol{\mu},\sigma}(a|x) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}\left[\nabla_{\mu,\sigma}G_{\mu,\sigma}(\epsilon, a, x)\right]$$

which can be seen as a one dimensional reparametrization trick gradient, and is known to behave better than the score function gradient estimator (Xu et al., 2019).

Optimising the bounds. For their practicality, we focus on the class of **LIG** policies to optimise the bounds. As these policies are built with Gaussian distributions Q, we also adopt Gaussian priors[3] $P = \mathcal{N}(\boldsymbol{\mu_0}, \sigma_0 I_d)$ to obtain an analytical expression for $\mathcal{KL}(Q||P)$. We state the bounds for **LIG** policies with Gaussian priors in Appendix 4.8.8. Optimizing for **LIG** policies, the best guaranteed risk defined in Section 4.2.3 and the minimizer $\pi^*_{\mathcal{UB}_n}$ for the different bounds $\mathcal{UB}_n \in \{\mathcal{LS}_n, \mathcal{C}_n, \mathcal{CBB}^{\xi}_n\}$ are given by:

$$\mathcal{GR}^*_{\mathcal{UB}_n} = \min_{\pi_{\boldsymbol{\mu},\sigma}} \mathcal{UB}_n(\pi_{\boldsymbol{\mu},\sigma}) \tag{4.7}$$

$$\pi^*_{\mathcal{UB}_n} = \arg\min_{\pi_{\boldsymbol{\mu},\sigma}} \mathcal{UB}_n(\pi_{\boldsymbol{\mu},\sigma}). \tag{4.8}$$

The best guaranteed improvement $\mathcal{GI}^*(\pi_0)$ with these bounds follows as the difference between $R(\pi_0)$ and $\mathcal{GR}^*$.

4.6 Experiments

We are interested in studying the effectiveness of the proposed bounds in providing guaranteed improvement of the previously deployed system π_0 after collecting $\mathcal{D}_n$.

Experimental Setup. For ease of exposition, we use a softmax logging policy of parameter $\mu_0 \in \mathbb{R}^{p\times K}$:

$$\pi^{\mathcal{S}}_{\boldsymbol{\mu_0}}(a|x) \propto \exp(\phi(x)^T \boldsymbol{\mu_0}_a).$$

$\mu_{0_a} \in \mathbb{R}^p$ is the parameter associated with action a. The policy $\pi^{\mathcal{S}}_{\boldsymbol{\mu_0}}$ is used to generate the logged interactions data $\mathcal{D}_n$ and its parameter $\boldsymbol{\mu_0}$ constructs the reference distribution $P = \mathcal{N}(\boldsymbol{\mu_0}, I_d)$

[3] The prior uses the parameters of the logging policy π_0.

for all bounds. We adopt the standard supervised-to-bandit conversion to generate logged data in all of our experiments (Swaminathan and Joachims, 2015a). We use two mutliclass datasets: **FashionMNIST** (Xiao et al., 2017) and **EMNIST-b** (Cohen et al., 2017), alongside two multilabel datasets: **NUS-WIDE-128** (Chua et al., 2009) with 128-VLAD features (Spyromitros-Xioufis et al., 2014) and **Mediamill** (Snoek et al., 2006) to empirically validate our findings. The statistics of the datasets are described in Table 4.1 in Appendix 4.9.1; N the size of the training split, K the number of actions and p the dimension of the features $\phi(x)$. We take a small fraction (5%) of the training data that will only be used to learn $\boldsymbol{\mu_0}$ in a supervised manner. With $\boldsymbol{\mu_0}$ obtained, we introduce an inverse temperature parameter α to our softmax logging policy $\pi^{\mathcal{S}}_{\alpha\boldsymbol{\mu_0}}$, giving a prior $P = \mathcal{N}(\alpha\boldsymbol{\mu_0}, I_d)$. Changing α allows us to cover logging policies with different entropies ($\alpha \approx 0$ gives a uniform π_0 and $\alpha \approx 1$ gives a peaked π_0). We run $\pi_{\alpha\boldsymbol{\mu_0}}$ on the rest ($n_c = 0.95N$) of the training data to generate $\mathcal{D}_n$. For a context x and an action a, we define in our setting the cost as $c = -\mathbb{1}[a \in y]$ with y the set of true labels for x. Learning $\boldsymbol{\mu_0}$ on a split different from the one logged allows us to use the previous bounds as the parameter $\boldsymbol{\mu_0}$ does not depend on the logged interactions, making the reference distribution P data-free. We set the allowed uncertainty to $\delta = 0.05$. For all datasets, τ will be set to $\tau = 1/K$ to get no bias when the logging policy is close to uniform. We use Adam (Kingma and Ba, 2014) with a learning rate of 10^{-3} for 100 epochs to optimize the bounds w.r.t their parameters. More details on the training procedure can be found in Appendix 4.9.2.

$\mathcal{LS}_n$ is not tight enough. We demonstrated in this work that the $\mathcal{LS}_n$ bound used by London and Sandler (2019) is suboptimal, as it generally shows a worse dependence on τ and can be shown to be theoretically dominated by $\mathcal{C}_n$ (Theorem 4.4.1). However, we want to verify if it can guarantee the improvement of the logging policy π_0. Given logged data $\mathcal{D}_n$ generated by π_0, we optimize the $\mathcal{LS}_n$ bound with both the **LIG** (Equation (4.7)) and **Mixed Logit** (Theorem 3 in London and Sandler (2019)) policy classes. In Figure 4.1, we plot $R(\pi_0)$, the true risk of π_0 alongside the best guaranteed risk $\mathcal{GR}^*_{\mathcal{LS}_n}$ given by the two policy classes, while changing the logging policies π_0 (going from uniform to peaked policies by changing α). We can observe, for the two policy classes and all scenarios considered, that this bound fails at providing guaranteed improvement as its guaranteed risk $\mathcal{GR}^*_{\mathcal{LS}_n}$ is always smaller than $R(\pi_0)$ and is vacuous ($\mathcal{GR}^*_{\mathcal{LS}_n} \geq 0$) for some datasets. This bound is not tight enough to be used with our strategy.

$\mathcal{C}_n$ and $\mathcal{CBB}^{\xi}_n$ do guarantee improvement. Given logged data $\mathcal{D}_n$, we optimize $\mathcal{C}_n$ and $\mathcal{CBB}^{\xi}_n$ for **LIG** policies (Equation (4.7)) and plot in Figure 4.2 the guaranteed risk $\mathcal{GR}^*$, the true risk of the minimizer $R(\pi^*)$ as well as the positive guaranteed improvement by the bound $\max(\mathcal{GI}^*, 0)$. To answer our question, we are interested in the first row (best guaranteed risk $\mathcal{GR}^*$) and last row (best guaranteed improvement $\mathcal{GI}^*$) of Figure 4.2. For the $\mathcal{CBB}^{\xi}_n$ bound, we study particularly the two values of $\xi \in \{0, -\frac{1}{2}\}$. We can observe that contrary to $\mathcal{LS}_n$, our bounds can guarantee improvement over π_0 in the majority of scenarios. The $\mathcal{C}_n$ bound gives great results when π_0 is close to uniform ($\alpha \to 0$) but sometimes fails (when $\alpha \to 1$) at improving the logging policy, and one can observe that in the context of **FashionMNIST** and **EMNIST-b**. As for the $\mathcal{CBB}^{\xi}_n$ bound, we can observe that choosing $\xi = -\frac{1}{2}$ consistently give the best results as it reduces considerably the dependency on $\mathcal{V}^{\tau}_n$. Note that $\mathcal{CBB}^{\xi}$ with $\xi = -\frac{1}{2}$ never fails to produce guaranteed improvement across all settings. Having a uniform logging policy π_0 is the worst regime for this bound, as $\mathcal{V}^{\tau}_n$ reaches its highest value $1/\tau$. This is empirically confirmed with our experiments. We can see in Figure 4.2 that, for small values of α, the $\mathcal{CBB}^{\xi}_n$ bound suffers the most; $\mathcal{CBB}_n(\xi = 0)$ is always worse than $\mathcal{C}_n$ and $\mathcal{CBB}_n(\xi = -\frac{1}{2})$ produces worse guarantees than $\mathcal{C}_n$ in the **NUS-WIDE-128** dataset. Once we drift away from

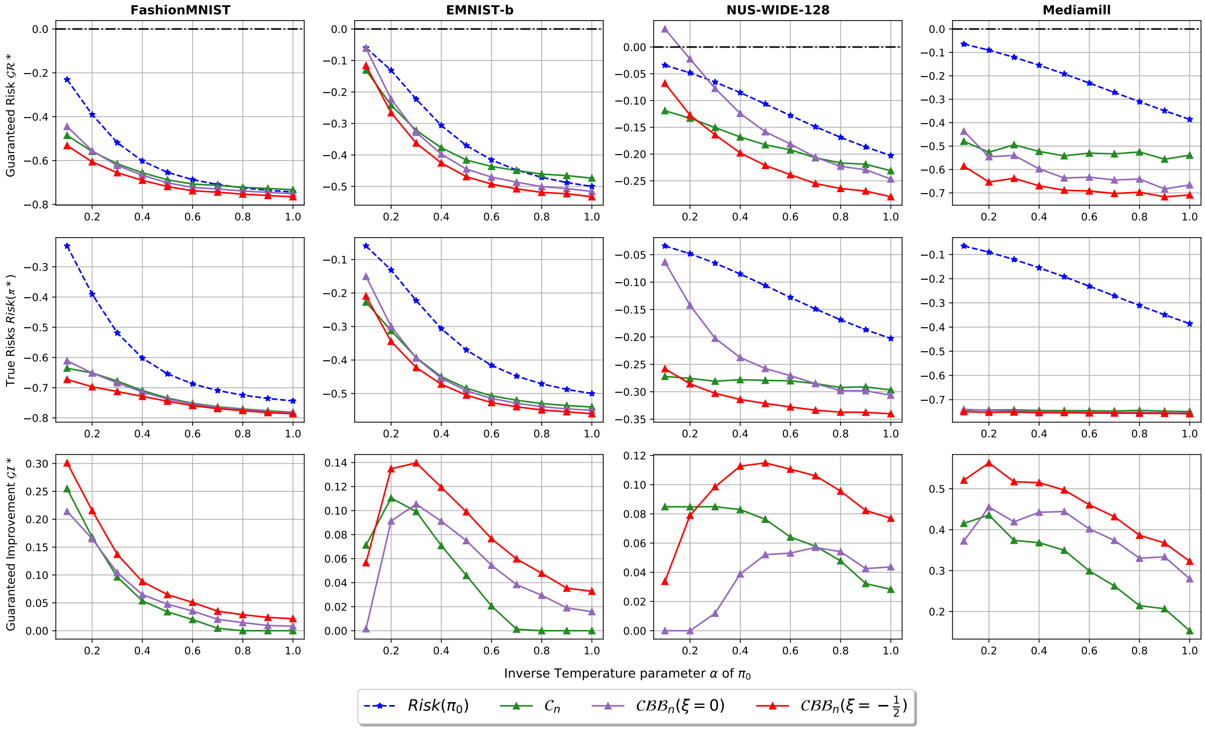


Figure 4.2: Behavior of the guaranteed risk $\mathcal{GR}^*$ (↓ is better), the risk of the minimizer $R(\pi^*)$ (↓ is better) and the guaranteed improvement $\mathcal{GI}^*$ (↑ is better) given by the bounds (optimized with **LIG** policies) while changing π_0. We can observe that the newly proposed bounds can efficiently improve on π_0, with $\mathcal{CBB}_n^{\xi}$ ($\xi = -\frac{1}{2}$) giving the best results.

uniform logging policies, the $\mathcal{CBB}^{\xi}$ bound, especially with $\xi = -\frac{1}{2}$ gives the best guarantees on all the datasets considered.

Tighter bounds give the best true risk. In real-world problems, we cannot have access to $R(\pi^*)$ before deployment. In our experiments, we can compute $R(\pi^*)$ on the test sets as we have access to the true labels[4]. We are interested in this quantity, as we want to make sure that the bounds giving low guaranteed risk $\mathcal{GR}^*$ will produce policies π^* with low true risk $R(\pi^*)$. Even if the limiting behaviours of the bounds give an intuition of how they compare, this will further confirm that the gap between the bounds is not linked to constants but to quantities valuable to learning π^*. The second row on Figure 4.2 confirms that the bounds with the best $\mathcal{GR}^*$ reliably give the best $R(\pi^*)$ in all settings.

Take away. These experiments confirm that the policies π^* obtained by optimising our newly proposed bounds improve, with high confidence, the logging policies π_0. The results also suggest the use of the variance sensitive bound $\mathcal{CBB}_n(\xi = -1/2)$ for its consistent results across the different scenarios. However, if computing expectations under π_0 is difficult, one can adopt $\mathcal{C}_n$ as it showed great results.

[4] up to a small $\mathcal{O}(1/\sqrt{n_t})$ approximation error.

4.7 Conclusion

In this work, we introduce a new theoretically grounded strategy for offline policy optimization. This approach is based on generalization bounds, uses all the available data and does not require additional hyperparameters. Leveraging PAC-Bayesian tools, we provide novel generalization bounds tight enough to make our strategy viable, giving practitioners a principled way to confidently improve over the previous decision system offline. Our results can nicely be extended to learning efficient policies over slates (Swaminathan et al., 2017) or continuous action policies (Kallus and Zhou, 2018). We believe that our work brings us closer to offline policy learning with online performance certificates. In the future, we would like to investigate tighter bounds for this problem and loosen our assumptions; e.g. to remove the need for having access to the logging policy π_0.

4.8 Appendix: Technical Results

4.8.1 Policies as mixtures of deterministic decision rules

As described in the chapter, a policy π takes a context $x \in \mathcal{X}$ and defines a probability distribution over the K-dimensional simplex Δ_K. In our work, we reinterpret policies as mixtures of deterministic decision rules.

Let f be the function that encodes the relevance of the action to the context x. Given a distribution Q over the functions $f \in \mathcal{F}_\Theta = \{f_\theta, \theta \in \Theta\}$, we define a policy as:

$$\forall x, a \quad \pi_Q(a|x) = \mathbb{E}_{f \sim Q}\left[\mathbb{1}\left[\operatorname*{argmax}_{a'} f(x, a') = a\right]\right].$$

A natural question is: can any policy π be written in this form?

In general, the answer depends on the set $\mathcal{F}_\Theta = \{f_\theta, \theta \in \Theta\}$ we are considering. When the class $\mathcal{F}_\Theta$ is rich enough, answer is yes, as proven by the following theorem.

Theorem 4.8.1. *Let us fix a policy π. Let*

$$\mathcal{G} = \{g : \mathcal{X} \times \mathcal{A} \to \{0, 1\} \textit{ such that } \forall x, \exists! a, g(x, a) = 1\}.$$

Then, there is a σ-algebra $\mathcal{S}$ on $\mathcal{G}$ and a probability distribution Q_π on $(\mathcal{G}, \mathcal{S})$ such that

$$\forall x, a \quad \pi(a|x) = \mathbb{E}_{f \sim Q_\pi}\left[\mathbb{1}\left[\operatorname*{argmax}_{a'} f(x, a') = a\right]\right].$$

Proof. Fix a policy π. Define the set $\Omega = [K]^{\mathcal{X}}$. That is, an element ω of Ω is a family of elements of $[K]$ indexed by $\mathcal{X}$: $\omega = (\omega_x)_{x \in \mathcal{X}}$. Define the set of cylinders

$$\mathcal{C} = \left\{A \subset \Omega : A = \prod_{x \in \mathcal{X}} A_x \text{ and } \operatorname{card}(\{x : A_x \neq [K]\}) < \infty\right\}.$$

For such a set $A = \prod_{x \in \mathcal{X}} A_x$ we define

$$P_\pi(A) = \prod_{x : A_x \neq \Omega} \left[\sum_{a \in A_x} \pi(a|x)\right].$$

Note in particular that, for a fixed $x \in \mathcal{X}$ and $a \in [K]$, we have

$$P_\pi(\{\omega \in \Omega : \omega_x = a\}) = \pi(a|x). \tag{4.9}$$

Then, Kolmogorov extension theorem guarantees that there is a unique extension of P_π to the σ-field $\mathcal{D}$ generated by $\mathcal{C}$, that is $\mathcal{D} = \sigma(\mathcal{C})$. We have thus built a probability space $(\Omega, \mathcal{D}, P_\pi)$.

Now, for any $\omega = (\omega_x)_{x\in\mathcal{X}}$, we define the function $f_\omega : \mathcal{X}\times\mathcal{A} \to \{0,1\}$ by $f_\omega(x,a) = \mathbb{1}[\omega_x = a]$. Define, for any $C \in \mathcal{D}$, $S_C := \{f_\omega, \omega \in C\}$ and $Q_\pi(S_C) = P_\pi(C)$, and finally put $\mathcal{S} = \{S_C, C \in \mathcal{D}\}$. As the function $F : \omega \mapsto f_\omega$ is a bijection from Ω to $\mathcal{G}$, $\mathcal{S}$ is a σ-field and Q_π is a probability distribution. We have thus equiped $\mathcal{G}$ with a σ-field $\mathcal{S}$ and a probability Q_π: $(\mathcal{G}, \mathcal{S}, Q_\pi)$ is a probability space. Now, we check that

$$\begin{aligned}
\mathbb{E}_{f\sim Q_\pi}\left[\mathbb{1}\left[\operatorname*{argmax}_{a'} f(x,a') = a\right]\right] &= Q_\pi\left(\left\{f \in \mathcal{G} : \operatorname*{argmax}_{a'} f(x,a') = a\right\}\right) \\
&= P_\pi\left(\left\{\omega \in \Omega : \operatorname*{argmax}_{a'} f_\omega(x,a') = a\right\}\right) \\
&= P_\pi\left(\{\omega \in \Omega : \omega_x = a\}\right) \\
&= \pi(a|x)
\end{aligned}$$

thanks to (4.9). This ends the proof. ■

4.8.2 Proof of Proposition 1

Proposition 4.4.2 is a direct application of Catoni (2007)'s bound (see Theorem 3 in Letarte et al. (2019)) to the rescaled cIPS $0 \le 1 + \tau \cdot \hat{R}^\tau_n(\cdot) \le 1$ with deterministic decision functions d_θ.

Proof. Let us fix a prior P over $\mathcal{F}_\Theta$ and $\tau \in (0,1]$. For any $\delta \in (0,1]$, we have with probability at least $1-\delta$ over draws of $\mathcal{D}_n \sim (\nu, \pi_0)^n$: for any Q that is P-continuous, any $\lambda > 0$:

$$1 + \tau\mathbb{E}_{(Q,\nu,\pi_0)}\left[\hat{R}^\tau_n(d_\theta)\right] \le \frac{1}{(1-e^{-\lambda})}\left(1 - \exp\left[-\lambda\cdot(1+\tau\mathbb{E}_Q[\hat{R}^\tau_n(d_\theta)]) - \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right]\right).$$

By linearity of the expectation and our clipped estimator $\hat{R}^\tau_n(\cdot)$, we get:

$$1 + \tau\cdot\mathbb{E}_{(\nu,\pi_0)}\left[\hat{R}^\tau_n(\pi_Q)\right] \le \frac{1}{(1-e^{-\lambda})}\left(1 - \exp\left[-\lambda\cdot(1+\tau\cdot\hat{R}^\tau_n(\pi_Q)) - \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right]\right).$$

Rearranging the terms gives:

$$\begin{aligned}
\mathbb{E}_{(\nu,\pi_0)}\left[\hat{R}^\tau_n(\pi_Q)\right] &\le \frac{e^{-\lambda}}{\tau(1-e^{-\lambda})}\left(1 - \exp\left[-\lambda\cdot\tau\cdot\hat{R}^\tau_n(\pi_Q) - \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right]\right) \\
&\le \frac{1}{\tau(e^{\lambda}-1)}\left(1 - \exp\left[-\lambda\cdot\tau\cdot\hat{R}^\tau_n(\pi_Q) - \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right]\right).
\end{aligned}$$

The last step is to exploit the fact that the bias of $\hat{R}^\tau_n(\cdot)$ is negative (because the cost $c \le 0$), we have for any π:

$$\begin{aligned}
\mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x)}\left[\hat{R}^\tau_n(\pi)\right] &= \mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x)}\left[\frac{\pi(a|x)}{\max(\pi_0(a|x), \tau)}c(x,a)\right] \\
&\ge \mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x)}\left[\frac{\pi(a|x)}{\pi_0(a|x)}c(x,a)\right] = R(\pi)
\end{aligned}$$

which gives the result stated in Proposition 4.4.2 by taking the minimum over $\lambda > 0$:

$$R(\pi_Q) \leq \min_{\lambda>0} \frac{1}{\tau(e^\lambda - 1)} \left(1 - \exp\left[-\lambda \cdot \tau \cdot \hat{R}_n^\tau(\pi_Q) - \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right]\right).$$

■

4.8.3 Limiting behavior of $\mathcal{C}_n$

To build an intuition of the dependency of $\mathcal{C}_n$ on both n and τ, we can linearize the bound by exploiting the well-known inequality $1 - \exp(-x) \leq x$ for $x \in [-1, 1]$. This gives:

$$\mathcal{C}_n(\pi_Q) \leq \min_{\lambda>0} \frac{\lambda}{(e^\lambda - 1)} \hat{R}_n^\tau(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{\tau(e^\lambda - 1)n}.$$

As the upper bound is a minimum over $\lambda > 0$, fixing $\lambda = \frac{1}{50}$ for example still gives a valid upper bound. This value leads to $\frac{\lambda}{(e^\lambda - 1)} \approx 1$ and $e^\lambda - 1 \approx \frac{1}{50}$, giving an approximated behaviour of the upper bound on $\mathcal{C}_n$ of:

$$\begin{aligned} \mathcal{C}_n(\pi_Q) &\leq \hat{R}_n^\tau(\pi_Q) + 50 \cdot \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{\tau n} \\ &\leq \hat{R}_n^\tau(\pi_Q) + \mathcal{O}\left(\frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{\tau n}\right). \end{aligned}$$

This result shows that $\mathcal{C}_n$ improves the dependency on n compared to $\mathcal{LS}_n$.

4.8.4 Proof of Theorem 1

Proof. We fix $\mathcal{D}_n \sim (\mu, \pi_0)^n, \tau \in (0, 1]$. To prove Theorem 4.4.1, we use the equality stated in Theorem 3 from Letarte et al. (2019) applied to the rescaled cIPS $0 \leq \hat{\mathcal{L}}_n(\cdot) = 1 + \tau \cdot \hat{R}_n^\tau(\cdot) \leq 1$. For any distribution P, any distribution Q that is P-continuous, $\delta \in (0, 1]$, we have:

$$\sup_{0\leq p\leq 1}\left\{p : kl(\hat{\mathcal{L}}_n(\pi_Q)||p) \leq \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right\} = 1 + \tau \cdot \mathcal{C}_n^{P,\delta,\tau}(\pi_Q)$$

with $\mathcal{C}_n^{P,\delta,\tau}(\pi_Q) := \min_{\lambda>0} \frac{1 - e^{-\tau\lambda\Gamma_n^\tau(Q,\lambda,\delta)}}{\tau(e^\lambda - 1)}$, $\Gamma_n^\tau(Q, \lambda, \delta) = \hat{R}_n^\tau(\pi_Q) + \frac{\mathcal{KL}(Q||P)+\ln\frac{2\sqrt{n}}{\delta}}{\tau\lambda n}$ and $kl(q||p) = q\log(\frac{q}{p}) + (1 - q)\log(\frac{1-q}{1-p})$, the KL divergence between two Bernoulli variables of parameters p and q. This means that:

$$kl(\hat{\mathcal{L}}_n(\pi_Q)||1 + \tau \cdot \mathcal{C}_n^{P,\delta,\tau}(\pi_Q)) \leq \frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta}}{n}.$$

By leveraging the following inequality: $p \leq q + \sqrt{2qkl(q||p)} + 2kl(q||p)$ for $p \leq q$, we get:

$$1 + \tau\mathcal{C}_n^{P,\delta,\tau}(\pi_Q) \leq 1 + \tau\hat{R}_n^\tau(\pi_Q) + \sqrt{\frac{2[1 + \tau\hat{R}_n^\tau(\pi_Q)](\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{n}} + \frac{2(\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{n}.$$

Giving the result of Theorem 4.4.1:

$$\mathcal{C}_n^{P,\delta,\tau}(\pi_Q) \leq \hat{R}_n^\tau(\pi_Q) + \sqrt{\frac{2[\frac{1}{\tau} + \hat{R}_n^\tau(\pi_Q)](\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{\tau n}} + \frac{2(\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n}}{\delta})}{\tau n}.$$

■

4.8.5 Bernstein-Type bound beyond the i.i.d. case

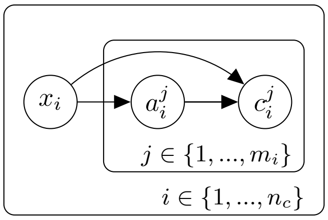


Figure 4.3: The "Multiple Interactions" Setting.

In a multitude of applications, the i.i.d. assumption made on $\{x_i, a_i, c_i\}_{i\in[n]}$ can be violated. Indeed, a decision system can interact with the same context x_i multiples times, trying different actions and logging the feedbacks as represented in Figure 4.3. Let m_i be the number of times the system interacted with context x_i. The logged dataset in this case can be represented by

$$\mathcal{D}_{n_c}^{\{m_i\}_{i\in[n_c]}} = \left\{x_i, \{a_i^j, c_i^j\}_{j\in[m_i]}\right\}_{i\in[n_c]}$$

with n_c representing the number of contexts and $n = \sum_i^{n_c} m_i$ the total number of datapoints. As soon as we have an $m_{i_0} > 1$, the i.i.d. assumption does not hold anymore as the samples $\{x_{i_0}, \{a_{i_0}^j, c_{i_0}^j\}_{j=1}^{m_{i_0}}\}$ share the same observation x_{i_0} and thus are dependent. In this case, the cvcIPS estimator will be written as:

$$\hat{R}_n^{\tau,\xi}(\pi_Q) = \xi + \sum_{i=1}^{n_c}\sum_{j=1}^{m_i} \frac{\omega_{\pi_Q}^{\tau}(a_i^j|x_i)(c_i^j - \xi)}{n_c m_i}$$

We recover the i.i.d. case by taking $m_i = 1\ \forall i$. Under this weaker assumption, Catoni (2007) or any classical PAC-Bayesian bound cannot be applied directly.

Proof of Proposition 3

In this section, we begin by stating Proposition 4.4.3 for the more general case of multiple interactions, where we have a logged dataset $\mathcal{D}_{n_c}^{\{m_i\}_{i\in[n_c]}}$.

Proposition 4.8.1. *Given a prior P on $\mathcal{F}_\Theta$, $\xi \in [-1, 0], \tau \in (0, 1]$, $\delta \in (0, 1]$ and a set of strictly positive scalars $\Lambda = \{\lambda_i\}_{i\in[n_\Lambda]}$. We have with probability at least $1-\delta$ over draws of $\mathcal{D}_{n_c}^{\{m_i\}_{i\in[n_c]}} \sim \prod_{i=1}^{n_c}(\nu, \pi_0^{m_i})$: For any Q that is P-continuous, any $\lambda \in \Lambda$:*

$$\begin{aligned} R(\pi_Q) \leq \hat{R}_n^{\tau,\xi}(\pi_Q) - \xi \mathcal{B}_{n_c}^{\tau}(\pi_Q) + \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{4\sqrt{n_c}}{\delta}}{2n_c}} \\ + \frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\lambda}{\delta}}{\lambda} + \frac{\lambda l_\xi}{n_c}\sum_{i=1}^{n_c}\frac{1}{m_i n_c} g\left(\frac{\lambda b_\xi}{m_i n_c}\right)\mathcal{V}^{\tau,i}(\pi_Q) \end{aligned}$$

with $g : u \rightarrow \frac{\exp(u)-1-u}{u^2}$, $l_\xi = \max\left(\xi^2, (1+\xi)^2\right)$, $b_\xi = \frac{1+\xi}{\tau} - \xi$,

$$\begin{aligned} \mathcal{V}^{\tau,i}(\pi) &= \mathbb{E}_{\pi(.|x_i)}\left[\frac{\pi_0(a|x_i)}{\max(\tau, \pi_0(a|x_i))^2}\right], \\ \mathcal{B}_{n_c}^{\tau}(\pi) &= \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbb{E}_{\pi(.|x_i)}\left[\mathbb{1}[\pi_0(a|x_i) < \tau]\left(1 - \frac{\pi_0(a|x_i)}{\tau}\right)\right]. \end{aligned}$$

We use a decomposition similar to Kuzborskij et al. (2021) and rewrite the difference $R(\pi_Q) - \hat{R}_n^{\tau,\xi}(\pi_Q) = D_1(\pi_Q) + D_2(\pi_Q) + D_3(\pi_Q)$ with:

$$\begin{aligned} D_1(\pi_Q) &= R(\pi_Q) - \frac{1}{n_c}\sum_{i=1}^{n_c} R(\pi_Q|x_i) \\ D_2(\pi_Q) &= \frac{1}{n_c}\sum_{i=1}^{n_c} R(\pi_Q|x_i) - \frac{1}{n_c}\sum_{i=1}^{n_c}\xi + \mathbb{E}_{a\sim\pi_0(\cdot|x_i)}\left[\omega_{\pi_Q}^{\tau}(a_i^j|x_i)(c(a,x_i) - \xi)\right] \\ D_3(\pi_Q) &= \frac{1}{n_c}\sum_{i=1}^{n_c}\xi + \mathbb{E}_{\pi_0(\cdot|x_i)}\left[\omega_{\pi_Q}^{\tau}(a|x_i)(c(a,x_i) - \xi)\right] - \hat{R}_n^{\tau,\xi}(\pi_Q). \end{aligned}$$

For the first difference D_1, we use McAllester (1998) PAC-Bayesian bound applied on the shifted $[0, 1]$-bounded loss of $0 \leq 1 + R(\pi_Q|x_i) \leq 1$. We get with probability at least $1 - \delta$, For any Q that is P-continuous:

$$D_1(\pi_Q) \leq \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{2\sqrt{n_c}}{\delta}}{2n_c}}. \tag{4.10}$$

The second difference quantifies the bias of our estimator given the contexts $\{x_i, ..., x_{n_c}\}$. Even if we cannot compute it, we can give an upper bound for D_2. We have:

$$\begin{aligned} D_2(\pi_Q) &= \frac{1}{n_c}\sum_{i=1}^{n_c} R(\pi_Q|x_i) - \frac{1}{n_c}\sum_{i=1}^{n_c} \xi + \mathbb{E}_{a\sim\pi_0(\cdot|x_i)}\left[\omega^{\tau}_{\pi_Q}(a|x_i)(c(a,x_i)-\xi)\right] \\ &= \frac{1}{n_c}\sum_{i=1}^{n_c} \mathbb{E}_{a\sim\pi_0(\cdot|x_i)}\left[(\omega^{0}_{\pi_Q}(a|x_i)-\omega^{\tau}_{\pi_Q}(a|x_i))(c(a,x_i)-\xi)\right] \\ &= \frac{1}{n_c}\sum_{i=1}^{n_c} \mathbb{E}_{a\sim\pi_0(\cdot|x_i)}\left[\mathbb{1}[\pi_0(a|x_i)<\tau](\frac{\pi_Q(a|x_i)}{\pi_0(a|x_i)} - \frac{\pi_Q(a|x_i)}{\tau})(c(a,x_i)-\xi)\right] \\ &= \frac{1}{n_c}\sum_{i=1}^{n_c} \mathbb{E}_{a\sim\pi_Q(\cdot|x_i)}\left[\mathbb{1}[\pi_0(a|x_i)<\tau](1-\frac{\pi_0(a|x_i)}{\tau})(c(a,x_i)-\xi)\right] \; (c\leq 0) \\ &\leq -\frac{\xi}{n_c}\sum_{i=1}^{n_c} \mathbb{E}_{a\sim\pi_Q(\cdot|x_i)}\left[\mathbb{1}[\pi_0(a|x_i)<\tau](1-\frac{\pi_0(a|x_i)}{\tau})\right] = -\xi\mathcal{B}^{\tau}_{n_c}(\pi_Q). \end{aligned}$$

We obtain $-\xi\mathcal{B}^{\tau}_{n_c}(\pi_Q)$, an empirical upper bound to $D_2(\pi_Q)$. The last step is to control the difference D_3. Before doing this, we need to state two lemmas that will help us control the difference D_3.

Lemma 4.8.1. *Change of measure: Let f be a function of the parameter θ and data S, for any distribution Q that is P continuous, for any $\delta \in (0,1]$, we have with probability $1-\delta$:*

$$\mathbb{E}_{\theta\sim Q}[f(\theta,S)] \leq \mathcal{KL}(Q||P) + \ln\frac{\Psi_f}{\delta} \tag{4.11}$$

with $\Psi_f = \mathbb{E}_S\mathbb{E}_{\theta\sim P}[e^{f(\theta,S)}]$.

Lemma 4.8.1 is the backbone of many PAC Bayes bounds. It is proven in many references, see for example Alquier (2021) or Lemma 1.1.3 in Catoni (2007). We will combine it with an inequality on the moment generating function to prove a Bernstein-like PAC-Bayes bound (Seldin et al., 2012).

Lemma 4.8.2. *Let W be a r.v with $\mathbb{E}[W^2] < \infty$, we suppose that $\mathbb{E}[W] - W \leq B$. Let $g : u \to \frac{\exp(u)-1-u}{u^2}$, we have for all $\eta \geq 0$:*

$$\mathbb{E}[\exp(\eta(\mathbb{E}[W]-W) - \eta^2 g(\eta B)\mathbb{V}[W])] \leq 1. \tag{4.12}$$

Lemma 4.8.2 is stated and proven in McDiarmid (1998). Combining both lemmas allows us to control the difference D_3 with a conditional Bernstein PAC-Bayesian bound:

Corollary 4.8.1. *Conditional Bernstein PAC-Bayesian Bound: Let's fix a $\lambda > 0$ and a prior P, for any distribution Q that is P continuous, for any $\delta \in (0,1]$, we have with probability at least $1-\delta$:*

$$D_3(\pi_Q) \leq \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda} + \frac{\lambda l_\xi}{n_c}\sum_{i=1}^{n_c}\frac{1}{m_i n_c} g\left(\frac{\lambda b_\xi}{m_i n_c}\right)\mathcal{V}^{\tau,i}(\pi_Q) \tag{4.13}$$

with $l_\xi = \max\left(\xi^2, (1+\xi)^2\right)$, $b_\xi = \frac{1+\xi}{\tau} - \xi$, $\mathcal{V}^{\tau,i}(\pi) = \mathbb{E}_{\pi(.|x_i)}\left[\frac{\pi_0(a|x_i)}{\max(\tau,\pi_0(a|x_i))^2}\right]$.

Proof. Let us fix a context x_i and an action a_i^j and let $\theta \sim P$. We have:

$$D_i^j(\theta) = \mathbb{E}_{\pi_0(\cdot|x_i)}\left[\omega^\tau_{d_\theta}(a|x_i)(c(a,x_i)-\xi)\right] - \omega^\tau_{d_\theta}(a_i^j|x_i)(c(a_i^j,x_i)-\xi) \le b_\xi = \frac{1+\xi}{\tau} - \xi.$$

We fix a $\lambda > 0$ and choose:

$$\begin{aligned} f(\theta,S) &= \sum_{i=1}^{n_c}\sum_{j=1}^{m_i}\left[\frac{\lambda}{m_i n_c} D_i^j(\theta) - (\frac{\lambda}{m_i n_c})^2 g\left(\frac{\lambda b_\xi}{m_i n_c}\right)\mathbb{E}_{\pi_0(\cdot|x_i)}[D_i(\theta)^2]\right] \\ &= \sum_{i=1}^{n_c}\sum_{j=1}^{m_i}\left[\Delta_i^j(\theta)\right]. \end{aligned}$$

From Lemma 2 and because the prior P does not depend on the data, we have:

$$\begin{aligned} \Psi_f = \mathbb{E}_{\prod_i \pi_0(\cdot|x_i)}\mathbb{E}_{\theta\sim P}[e^{f(\theta,S)}] &= \mathbb{E}_{\theta\sim P}\mathbb{E}_{\prod_i \pi_0(\cdot|x_i)}[e^{f(\theta,S)}] \\ &= \mathbb{E}_{\theta\sim P}\prod_i(\mathbb{E}_{\pi_0(\cdot|x_i)}[e^{\Delta_i^0(\theta)}])^{m_i} \le 1. \end{aligned}$$

It means that $\ln \Psi_f \le 0$. Using this in Lemma 1, we get:

$$\begin{aligned} D_3(\pi_Q) &\le \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda} + \sum_{i=1}^{n_c}\sum_{j=1}^{m_i}\frac{\lambda}{(m_i n_c)^2} g\left(\frac{\lambda b_\xi}{m_i n_c}\right)\mathbb{E}_{\pi_0(\cdot|x_i)}\left[\mathbb{E}_{\theta\sim Q}[D_i(\theta)^2]\right] \\ &= \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda} + \frac{\lambda}{n_c}\sum_{i=1}^{n_c}\frac{1}{m_i n_c} g\left(\frac{\lambda b_\xi}{m_i n_c}\right)\mathbb{E}_{\theta\sim Q}\left[\mathbb{E}_{\pi_0(\cdot|x_i)}\left[D_i(\theta)^2\right]\right]. \end{aligned}$$

we also use the following inequality to upper bound $\mathbb{E}_{\pi_0(\cdot|x_i)}[D_i(\theta)^2]$:

$$\begin{aligned} \mathbb{E}_{\pi_0(\cdot|x_i)}[D_i(\theta)^2] &\le \mathbb{E}_{a\sim\pi_0(\cdot|x_i)}\left[\frac{d_\theta(a|x_i)}{\max(\pi_0(a|x_i),\tau)^2}(c(a,x_i)-\xi)^2\right] \\ &\le \max(\xi^2,(1+\xi)^2)\mathbb{E}_{a\sim\pi_0(\cdot|x_i)}\left[\frac{d_\theta(a|x_i)}{\max(\pi_0(a|x_i),\tau)^2}\right] \quad \text{because both } c,\xi \in [-1,0] \\ &= l_\xi \mathcal{V}^{\tau,i}(d_\theta). \end{aligned}$$

As the quantity $\mathcal{V}^{\tau,i}$ is linear in d_θ, the result in Corollary 1 follows:

$$D_3(\pi_Q) \le \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda} + \frac{\lambda l_\xi}{n_c}\sum_{i=1}^{n_c}\frac{1}{m_i n_c} g\left(\frac{\lambda b_\xi}{m_i n_c}\right)\mathcal{V}^{\tau,i}(\pi_Q).$$

Finally, We take a union bound of Corollary 1 over Λ, a discrete set with cardinal n_Λ, and combine its result with the bound giving (4.10) through another union bound to obtain Proposition 4.8.1.

■

Choice of Λ when $m_i = m$

When the number of interactions m is constant across all contexts, the result in Corollary 1 becomes for a fixed λ:

$$D_3(\pi_Q) \le \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} + \lambda l_\xi g\left(\lambda b_\xi\right)\mathcal{V}^\tau_{n_c}(\pi_Q)$$

where $\mathcal{V}_{n_c}^{\tau}(\pi_Q)$ was defined in Proposition 4.4.3. We would like to choose a λ that minimizes the bound on D_3. Unfortunately, we cannot do it because the minimizer λ^* depends on Q. Instead, we build an interval in which λ^* can be found. The function $g : u \to \frac{\exp(u)-1-u}{u^2}$ behaves like $\frac{\exp(u)}{u^2}$ when u is big enough, meaning that we should control the values of g, and thus λ by an upper bound. Choosing $\lambda \leq b = \frac{2n}{b_\xi}$ allows us to control the function $g\left(\frac{\lambda b_\xi}{n}\right) \leq g(2) \leq 1.1$.

Now that an upper bound is found, we still need to find the lowest possible value for λ^*. Of course, choosing the interval $[0, b]$ can be enough but we want to do more than that. λ^* verifies the following equality:

$$\lambda^* = \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\frac{l_\xi}{n} g\left(\frac{\lambda^* b_\xi}{n}\right)\mathcal{V}_{n_c}^{\tau}(\pi_Q) + \frac{\lambda^* l_\xi b_\xi}{n^2} g'\left(\frac{\lambda^* b_\xi}{n}\right)\mathcal{V}_{n_c}^{\tau}(\pi_Q)}}.$$

Let's assume that $\lambda^\star \leq b$. (If not, we can still restrict to $\lambda \in [a, b]$, with the value of a found below.) We have that $\mathcal{KL}(Q||P) \geq 0$, and $\mathcal{V}_{n_c}^{\tau} \leq \frac{1}{\tau}$. As the function g is increasing and convex (g' increasing), we get the following inequality:

$$\lambda^* \geq \sqrt{\frac{n\tau \ln\frac{1}{\delta}}{l_\xi g(2) + 2 l_\xi g'(2)}}.$$

Using the fact that $g'(2) = 1/2$ and $g(2) + 1 \leq 5/2$, we get:

$$\lambda^* \geq \sqrt{\frac{n\tau \ln\frac{1}{\delta}}{l_\xi g(2) + l_\xi}} \geq \sqrt{\frac{2n\tau \ln\frac{1}{\delta}}{5 l_\xi}} = a.$$

We now have an interval $\lambda^* \in [a, b]$. One can observe that the optimal $\mathcal{O}(\sqrt{n}) \leq \lambda^* \leq \mathcal{O}(n)$. We choose the set Λ to be a linear discretization of $[a, b]$ giving $\Lambda = \{a + i(b-a)\}_{i \in [n_\Lambda]}$.

Dependencies of the bound

The bound for the i.i.d. case can be written as:

$$R(\pi_Q) \leq \hat{R}_n^{\tau,\xi}(\pi_Q) - \xi\mathcal{B}_n^{\tau}(\pi_Q) + \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{4\sqrt{n}}{\delta}}{2n}} + \min_{\lambda\in\Lambda}\left\{\frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}}{\lambda n} + \lambda l_\xi g\left(\lambda b_\xi\right)\mathcal{V}_n^{\tau}(\pi_Q)\right\}$$

with $\Lambda = \{\frac{a}{b} + i\frac{(b-a)}{n}\}_{i\in[n_\Lambda]}$. We know that the biggest value of $\lambda \in \Lambda$ is $b = \frac{2}{b_\xi}$ and that $g(2) \approx 1$. This gives:

$$\begin{aligned}\min_{\lambda\in\Lambda} A(\lambda) &= \min_{\lambda\in\Lambda} \frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}}{\lambda n} + \lambda l_\xi g\left(\lambda b_\xi\right)\mathcal{V}_n^{\tau}(\pi_Q)\\ &\leq \min_{\lambda\in\Lambda} \frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}}{\lambda n} + \lambda l_\xi g\left(2\right)\mathcal{V}_n^{\tau}(\pi_Q)\\ &\lessapprox \min_{\lambda\in\mathbb{R}^+} \frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}}{\lambda n} + \lambda l_\xi \mathcal{V}_n^{\tau}(\pi_Q) = 2\sqrt{\frac{l_\xi \mathcal{V}_n^{\tau}(\pi_Q)\left(\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}\right)}{n}}\end{aligned}$$

We make the hypothesis that our Λ is well built to have a value of λ close to the true minimizer most of the time. This gives the following limiting behavior:

$$\mathcal{CBB}_n^{\xi}(\pi_Q) = \hat{R}_n^{\tau,\xi}(\pi_Q) - \xi\mathcal{B}_n^{\tau}(\pi_Q) + \mathcal{O}\left(\left(\frac{1}{2\sqrt{2}} + \sqrt{l_\xi \mathcal{V}_n^{\tau}(\pi_Q)}\right)\sqrt{\frac{\mathcal{KL}(Q||P)}{n}}\right).$$

4.8.6 Linear Independent Gaussian Policies

To obtain these policies, we restrict f_θ to:

$$\forall x, a \quad f_\theta(x, a) = \phi(x)^T \theta_a \tag{4.14}$$

with ϕ a fixed transform over the contexts. This results in a parameter θ of dimension $d = p \times K$ with p the dimension of the features $\phi(x)$ and K the number of actions. We also restrict the family of distributions $\mathcal{Q}_{d+1} = \{Q_{\boldsymbol{\mu},\sigma} = \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_d), \boldsymbol{\mu} \in \mathbb{R}^d, \sigma > 0\}$ to independent Gaussians with shared scale.

Estimating the propensity of a given x reduces the computation to a one dimensional integral:

$$\pi_{\boldsymbol{\mu},\sigma}(a|x) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \left[\prod_{a' \neq a} \Phi \left(\epsilon + \frac{\phi(x)^T (\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma ||\phi(x)||} \right) \right]$$

with Φ the cumulative distribution function of the standard normal.

Proof. We rewrite the definition of $\pi_{\boldsymbol{\mu},\sigma}$ as a probability and exploit the stability of the Gaussian distribution.

$$\begin{aligned}
\pi_{\boldsymbol{\mu},\sigma}(a|x) &= \mathbb{E}_{\theta \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_d)} \left[\mathbb{1}[\text{argmax}_{a'} \, \phi(x)^T \theta_{a'} = a] \right] \\
&= \mathbb{E}_{S \sim \mathcal{N}(\phi(x)^T \boldsymbol{\mu}, \sigma^2 ||\phi(x)||^2 I_K)} \left[\mathbb{1}[\text{argmax}_{a'} \, S_{a'} = a] \right] \\
&= \mathbb{P}_{S \sim \mathcal{N}(\phi(x)^T \boldsymbol{\mu}, \sigma^2 ||\phi(x)||^2 I_K)} \left(\text{argmax}_{a'} \, S_{a'} = a \right) \\
&= \mathbb{P}_{S \sim \mathcal{N}(\phi(x)^T \boldsymbol{\mu}, \sigma^2 ||\phi(x)||^2 I_K)} \left(S_a \geq S_{a'}, \quad \forall a' \neq a \right) \\
&= \mathbb{P}_{Z \sim \mathcal{N}(0_K, I_K)} \left(Z_a + \frac{\phi(x)^T (\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma ||\phi(x)||} \geq Z_{a'}, \quad \forall a' \neq a \right).
\end{aligned}$$

We condition on Z_a to obtain independent events as for all a, the random variables Z_a are independent.

$$\begin{aligned}
\pi_{\boldsymbol{\mu},\sigma}(a|x) &= \mathbb{P}_{Z \sim \mathcal{N}(0_K, I_K)} \left(Z_a + \frac{\phi(x)^T (\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma ||\phi(x)||} \geq Z_{a'}, \quad \forall a' \neq a \right) \\
&= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \left[\mathbb{P}_{Z \sim \mathcal{N}(0_K, I_K)} \left(\epsilon + \frac{\phi(x)^T (\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma ||\phi(x)||} \geq Z_{a'}, \quad \forall a' \neq a | Z_a = \epsilon \right) \right] \\
&= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \left[\prod_{a' \neq a} \mathbb{P}_{z \sim \mathcal{N}(0,1)} \left(z \leq \epsilon + \frac{\phi(x)^T (\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma ||\phi(x)||} \right) \right] \\
&= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \left[\prod_{a' \neq a} \Phi \left(\epsilon + \frac{\phi(x)^T (\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma ||\phi(x)||} \right) \right].
\end{aligned}$$

■

4.8.7 Why not Mixed Logit Policies?

London and Sandler (2019) used in their analysis Mixed Logit Policies to derive a learning principle for softmax policies. Mixed Logit Policies can be written as:

$$\forall (a, x), \quad \pi^{ML}_{\boldsymbol{\mu},\sigma}(a|x) = \mathbb{E}_{\theta \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_d)} [\text{softmax}_K(\phi(x)^T \theta_a)].$$

Even if these policies can behave properly (reparametrization trick gradient for instance), they are not ideal for learning with guarantees in the context of Offline Contextual Bandits.

Indeed, we know that the solution of the contextual bandit problem is a deterministic decision function d^*, always choosing the action with the minimum cost. Let us suppose that there exists a parameter μ^* such that:

$$\forall (a,x), \quad d^*(a|x) = d_{\mu^*}(a|x) = \mathbb{1}[\text{argmax}_{a' \in \mathcal{A}}(\phi(x)^T \mu^*_{a'}) = a]$$

We also suppose that we have access to its parameter μ^*. To recover d_{μ^*} with **LIG** policies, we need to have the scale parameter small enough $\sigma \to 0$ as :

$$\pi_{\boldsymbol{\mu}^*,\sigma}(a|x) \xrightarrow[\sigma \to 0]{} d_{\mu^*}(a|x) \quad \forall x, a.$$

For **Mixed Logit** policies however, having $\sigma \to 0$ is not enough as:

$$\pi^{ML}_{\boldsymbol{\mu}^*,\sigma}(a|x) \xrightarrow[\sigma \to 0]{} \text{softmax}_K(\phi(x)^T \mu^*_a) \quad \forall x, a.$$

One should also increase the norm of μ^* enough ($||\mu^*|| \to \infty$) to obtain d_{μ^*}.

Let us suppose that we start with the same prior $P = \mathcal{N}(\boldsymbol{\mu}^*, I_d)$ in our bounds. The price to pay in terms of complexity $\mathcal{KL}(Q_{\boldsymbol{\mu},\sigma}||P)$ to obtain the solution; a deterministic policy, will be much higher for **Mixed Logit** policies (as we should decrease σ and increase the norm of $\boldsymbol{\mu}$) than **LIG** policies (only decrease σ and let $\boldsymbol{\mu} = \boldsymbol{\mu}^*$). This means that for a fixed number of samples n, we will always get better results with **LIG** policies than **Mixed Logit** policies.

4.8.8 The bounds stated for LIG policies

In this section, we want to state the previous Propositions 4.4.2 and 4.4.3 (valid for any policy) for the class of **LIG** policies. This class of policies uses Independent Gaussian distributions with shared scale so we will begin by stating the KL divergence between $P = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0 I_d)$ and $Q = \mathcal{N}(\boldsymbol{\mu}, \sigma I_d)$. We have:

$$\mathcal{KL}(Q||P) = D[\boldsymbol{\mu}, \sigma, \boldsymbol{\mu_0}, \sigma_0] = \frac{||\mu - \mu_0||^2}{2\sigma_0^2} + d\left(\frac{\sigma^2}{2\sigma_0^2} + \ln\frac{\sigma_0}{\sigma} - \frac{1}{2}\right).$$

We write the bounds slightly differently by taking the minimimum over the considered λ (if the bound is true for any λ, it is true for the minimum of the bound over λ). We state Catoni's bound for **LIG** policies:

Corollary 4.8.2. ***LIG*** *policies with Catoni's bound*
Given a Gaussian prior $P = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0 I_d)$, $\tau \in (0,1]$, $\delta \in (0,1]$. We have with probability $1-\delta$ over draws of $\mathcal{D}_n \sim (\nu, \pi_0)^n$:
$\forall \boldsymbol{\mu} \in \mathbb{R}^d, \sigma > 0$:

$$R(\pi_{\boldsymbol{\mu},\sigma}) \leq \min_{\lambda > 0} \frac{1}{\tau(e^\lambda - 1)}\left[1 - \exp\left(-\tau\lambda\hat{R}^\tau_n(\pi_{\boldsymbol{\mu},\sigma}) + \frac{D[\boldsymbol{\mu}, \sigma, \boldsymbol{\mu_0}, \sigma_0] + \ln\frac{2\sqrt{n}}{\delta}}{n}\right)\right]$$

We call $\mathcal{C}_n(\pi_{\boldsymbol{\mu},\sigma})$ the upper bound stated by this corollary. We get:

$$\mathcal{GR}^*_{\mathcal{C}} = \min_{\pi_{\boldsymbol{\mu},\sigma}} \mathcal{C}_n(\pi_{\boldsymbol{\mu},\sigma})$$

$$\pi^*_{\mathcal{C}} = \arg\min_{\pi_{\boldsymbol{\mu},\sigma}} \mathcal{C}_n(\pi_{\boldsymbol{\mu},\sigma})$$

$$\mathcal{GI}^*_{\mathcal{C}} = R(\pi_0) - \mathcal{GR}^*_{\mathcal{C}}.$$

Similarly, we state our variance sensitive bound for **LIG** policies:

Corollary 4.8.3. ***LIG*** *policies variance sensitive bound. Given a Gaussian prior* $P = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0 I_d)$, $\xi \in [-1, 0], \tau \in (0, 1]$, $\delta \in (0, 1]$ *and a set of strictly positive scalars* $\Lambda = \{\lambda_i\}_{i \in [n_\Lambda]}$. *We have with probability at least* $1 - \delta$ *over draws of* $\mathcal{D}^m_{n_c} \sim \prod_{i=1}^{n_c}(\nu, \pi_0^m)$*:* $\forall \boldsymbol{\mu} \in \mathbb{R}^d, \sigma > 0$*:*

$$R(\pi_{\boldsymbol{\mu},\sigma}) \leq \hat{R}_n^{\tau,\xi}(\pi_{\boldsymbol{\mu},\sigma}) - \xi \mathcal{B}^\tau_{n_c}(\pi_{\boldsymbol{\mu},\sigma}) + \sqrt{\frac{D[\boldsymbol{\mu}, \sigma, \boldsymbol{\mu}_\mathbf{0}, \sigma_0] + \ln \frac{4\sqrt{n_c}}{\delta}}{2n_c}}$$
$$+ \min_{\lambda \in \Lambda} \left\{ \frac{D[\boldsymbol{\mu}, \sigma, \boldsymbol{\mu}_\mathbf{0}, \sigma_0] + \ln \frac{2n_\lambda}{\delta}}{\lambda} + \frac{\lambda l_\xi}{n} g\left(\frac{\lambda b_\xi}{n}\right) \mathcal{V}^\tau_{n_c}(\pi_{\boldsymbol{\mu},\sigma}) \right\}$$

We call $\mathcal{CBB}_n(\pi_{\boldsymbol{\mu},\sigma}, \xi, m)$ the upper bound stated by this corollary. Similarly we get:

$$\begin{aligned} \mathcal{GR}^*_{\mathcal{CBB}(\xi,m)} &= \min_{\pi_{\boldsymbol{\mu},\sigma}} \mathcal{CBB}_n(\pi_{\boldsymbol{\mu},\sigma}, \xi, m) \\ \pi^*_{\mathcal{CBB}(\xi,m)} &= \arg\min_{\pi_{\boldsymbol{\mu},\sigma}} \mathcal{CBB}_n(\pi_{\boldsymbol{\mu},\sigma}, \xi, m) \\ \mathcal{GI}^*_{\mathcal{CBB}(\xi,m)} &= R(\pi_0) - \mathcal{GR}^*_{\mathcal{CBB}(\xi,m)}. \end{aligned}$$

4.9 Appendix: Additional Experiments

4.9.1 Detailed Statistics of the dataset splits used

As described in the experiments section, we use the supervised to bandit conversion to simulate logged data as previously adopted in the majority of the literature (Swaminathan and Joachims, 2015a,b; London and Sandler, 2019; Faury et al., 2020; Sakhi et al., 2020b). In this procedure, you need a split D_l (of size n_l) to train the logging policy π_0, another split D_c (of size n_c) to generate the logging feedback with π_0, and finally a test split D_{test} (of size n_{test}) to compute the true risk $R(\pi)$ of any policy π. In our experiments, we split the training split D_{train} (of size N) of the four datasets considered into D_l ($n_l = 0.05N$) and D_c ($n_c = 0.95N$) and use their test split D_{test}. The detailed statistics of the different splits can be found in Table 4.1.

Datasets	N	n_l	n_c	n_{test}	K	p
FashionMNIST	60 000	3000	57 000	10 000	10	784
EMNIST-b	112 800	5640	107 160	18 800	47	784
NUS-WIDE-128	161 789	8089	153 700	107 859	81	128
Mediamill	30 993	1549	29 444	12 914	101	120

Table 4.1: Detailed statistics of the splits used.

4.9.2 Detailed hyperparameters

Contrary to previous work, our method does not require tuning any loss function hyperparameter over a hold out set. We do however need to choose parameters to optimize the policies.

The logging policy. π_0 is trained on D_l (supervised manner) with the following parameters:

- We use L_2 regularization of 10^{-6}. This is used to prevent the logging policy π_0 from being close to deterministic, allowing efficient learning with importance sampling.
- We use Adam (Kingma and Ba, 2014) with a learning rate of 10^{-1} for 10 epochs.

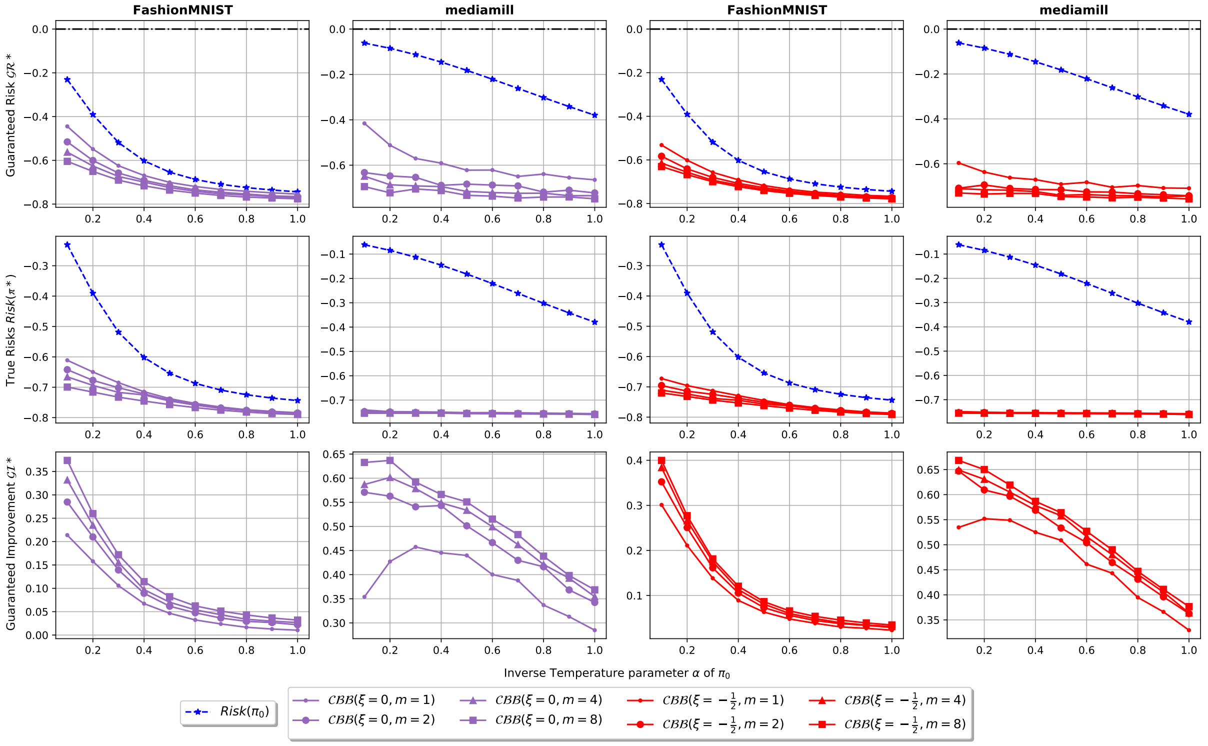


Figure 4.4: Behavior of the guaranteed risk $\mathcal{GR}^*$ (↓ is better), the risk of the minimizer $R(\pi^*)$ (↓ is better) and the guaranteed improvement $\mathcal{GI}^*$ (↑ is better) given by changing the number of interactions m and π_0.

Optimising the bounds. All the bounds are optimized with the following parameters:

- The clipping parameter τ is fixed to $1/K$ with K the action size of the dataset.
- We use Adam (Kingma and Ba, 2014) with a learning rate of 10^{-3} for 100 epochs.
- For the bounds optimized over **LIG** policies, the gradient is a one dimensional integral, and is approximated using $S = 32$ samples.

$$
\begin{aligned}
\pi_{\boldsymbol{\mu},\sigma}(a|x) &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}\left[\prod_{a'\neq a}\Phi\left(\epsilon + \frac{\phi(x)^T(\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma||\phi(x)||}\right)\right] \\
&\approx \frac{1}{S}\sum_{s=1}^{S}\prod_{a'\neq a}\Phi\left(\epsilon_s + \frac{\phi(x)^T(\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'})}{\sigma||\phi(x)||}\right) \qquad \epsilon_1, ..., \epsilon_S \sim \mathcal{N}(0,1).
\end{aligned}
$$

- For $\mathcal{C}_n$, we treat λ as a parameter and we look for the minimum of the bound with respect to μ, σ and λ.
- For $\mathcal{CBB}^{\xi}_n$, we choose the size of Λ to be $n_\Lambda = 100$ and for each iteration j of the optimization procedure, we take $\lambda_j \in \Lambda$ that minimizes the estimated bound and proceed to compute the gradient w.r.t μ and σ with λ_j.

4.9.3 Impact of changing the number of interactions m

The bound proposed in Proposition 4.4.3 can work beyond the i.i.d. setting and applies to the "multiple interactions" case. Intuitively, adding more interactions with the contexts x allows us to reduce the uncertainty on the cost and thus learn better policies. We want to explore this in Figure 4.4. We construct with π_0 a logged dataset with the number of interactions $m \in \{1, 2, 4, 8\}$ using both **FashionMNIST** and **Mediamill** datasets. Once $m > 1$, we can only use the $\mathcal{CBB}$ bound. We stick to the values of ξ previously used $\xi \in \{0, -1/2\}$.

We can observe that increasing the number of m consistently give better results, in terms of guarantees and also the quality of the policy π^* minimizing the bounds. We can also observe that even though m reduces the gap between the two estimators ($\xi = 0$ compared to $\xi = -1/2$), the cvcIPS estimator with $\xi = -1/2$ still gives the best results.

Chapter 5

A Better PAC-Bayesian Analysis of Offline Learning

In this chapter, we aim at refining the PAC-Bayesian analysis provided earlier. Instead of adapting known proof techniques to our estimators, we want to derive bounds that exploit the structure of the problem while reducing the requirements to use the bounds. Our previous analysis relied on risk estimators that are *bounded*, assumed the knowledge of the structure of π_0 to build a prior ($\pi_P = \pi_0$), and also assumed the full access to the logging policy π_0 to compute both the bias and the variance. We provide new, fully empirical bounds, based on a more accurate analysis of the risk, that combined with a novel procedure, gets completely rid of the assumption of accessing π_0 or knowing its structure.

Contents

5.1 Introduction

In the previous chapter, we advocated for a new, theoretically grounded strategy to confidently improve on the logging policy π_0 based on tight generalization bounds. We motivated PAC-Bayesian theory as a good candidate to answer this need. The whole paradigm behind off-policy learning resembles the PAC-Bayesian one; we start from a distribution and want to improve it with data-driven methods. In addition, PAC-Bayes is proven to give tight generalization bounds that can be used in our framework to give guarantees about the policy we want to deploy. Our previous analysis demonstrated the effectiveness of our approach in multiple scenarios; our bounds guarantee improvement over π_0, providing an optimization algorithm that scales to large datasets and requires no hyperparameter tuning. Our strongest results came from a variance sensitive bound that needs full access to the currently deployed policy π_0. We recall the bound in the following. Let g be the function:

$$g : u \to \frac{\exp(u) - 1 - u}{u^2}.$$

We recall the expressions of the control variate risk $\hat{R}_n^{\tau,\xi}$, the conditional bias $\mathcal{B}_n^\tau$ and the conditional second moment $\mathcal{V}_n^\tau$:

- $\hat{R}_n^{\tau,\xi}(\pi) = \xi + \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\max\{\pi_0(a_i|x_i), \tau\}}(c_i - \xi)$
- $\mathcal{B}_n^\tau(\pi) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\pi(.|x_i)}\left[\mathbb{1}[\pi_0(a|x_i) < \tau]\left(1 - \frac{\pi_0(a|x_i)}{\tau}\right)\right]$
- $\mathcal{V}_n^\tau(\pi) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\pi(.|x_i)}\left[\frac{\pi_0(a|x_i)}{\max\{\pi_0(a|x_i), \tau\}^2}\right].$

With these definitions, we can state the strongest result of the previous chapter:

Proposition 5.1.1. *Given a prior P on $\mathcal{F}_\Theta$, $\xi \in [-1, 0], \tau \in (0, 1]$, $\delta \in (0, 1]$ and a set of strictly positive scalars $\Lambda = \{\lambda_i\}_{i\in[n_\Lambda]}$. The following bound holds with probability at least $1 - \delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:*

$$\begin{aligned} R(\pi_Q) \leq \hat{R}_n^{\tau,\xi}(\pi_Q) - \xi\mathcal{B}_n^\tau(\pi_Q) + \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{4\sqrt{n}}{\delta}}{2n}} \\ + \min_{\lambda\in\Lambda}\left\{\frac{\mathcal{KL}(Q||P) + \ln\frac{2n_\Lambda}{\delta}}{\lambda n} + \lambda l_\xi g\left(\lambda b_\xi\right)\mathcal{V}_n^\tau(\pi_Q)\right\}\end{aligned}$$

with $l_\xi = \max\left[\xi^2, (1+\xi)^2\right]$, $b_\xi = (1+\xi)/\tau - \xi$.

To make the use of this bound viable, we need to define a good prior P and be able to compute both the conditional bias and the conditional second moment. This requires the access to π_0 as:

- The prior P was fixed to the distribution that induced π_0 ($\pi_P = \pi_0$). As we are using this prior, the procedure assumed access to P and thus π_0.
- The values of the probabilities of π_0 on all actions is needed to compute both the conditional bias and the conditional second moment. This requires full access to π_0.

Contributions. In some applications, π_0 has a simple structure satisfying these assumptions. In general, especially for complex decision systems, the policy deployed π_0 is a combination of different policies that are called upon depending on the context, making the reduction of this potentially complicated system to a simple structure (find a P that gives $\pi_P = \pi_0$) infeasible. In other applications, even computing the probabilities of π_0 is difficult, which makes the previous assumption hard to satisfy. In this chapter, we want to provide practitioners with a procedure that keeps the same or even improves the guarantees given by this bound while loosening the requirement of access to π_0. To achieve this:

- We derive a new family of fully empirical PAC-Bayesian bounds that is tighter than Proposition 5.1.1 with no quantity requiring access to π_0.
- We define a procedure that allows the use of data-dependent priors P, which alleviates the need to fix the prior to the logging policy π_0.
- We demonstrate empirically the superiority of this procedure in providing better policies while loosening the assumptions.

5.2 A Family of Estimators

In the offline learning paradigm, we start by defining the estimator of the risk that is suitable for our analysis. In this chapter, our purpose is to study a family of risk estimators that can cover various well-known estimators for our analysis to be as general as possible. Let us recall the assumption on the cost gathered;

$$\forall (x, a) \quad c(a, x) \in [-1, 0].$$

This is easily obtained as the cost can be rescaled within this interval. Given the logged interactions $\mathcal{D}_n$ of our logging policy π_0, and for any policy π, we define the following estimator of the risk $\hat{R}_n^p(\pi)$, with the help of a function $p : \mathbb{R} \to \mathbb{R}$ as:

$$\hat{R}_n^p(\pi) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i$$

with the only condition on p to be $\{\boldsymbol{C_1} : \forall x, p(x) \geq x\}$. This condition helps us control the impact of actions with low probabilities under π_0. This risk estimator encompasses well known risk estimators depending on the choice of p, we can recover:

$\forall x, p(x) = x \implies$ IPS estimator (Horvitz and Thompson, 1952)
$\forall x, p(x) = \max(\tau, x), \tau \in [0, 1] \implies$ cIPS estimator (Bottou et al., 2013)
$\forall x, p(x) = x^\alpha, \alpha \in [0, 1] \implies$ Exponential Smoothing IPS estimator (Aouali et al., 2023a)
$\forall x, p(x) = x + \gamma, \gamma \geq 0 \implies$ Linear Smoothing IPS estimator (Kuzborskij et al., 2021)

With this condition alone, our estimator is not necessarily bounded (from below) and needs careful treatment, as common PAC-Bayesian tools cannot be easily adapted. The condition $\boldsymbol{C_1}$ however results in risk estimators with a positive bias, which is a needed property for our analysis. For a policy π, let $R^p(\pi)$ be the expectation of our estimator:

$$R^p(\pi) = \mathbb{E}_{x \sim \nu, a \sim \pi_0(\cdot|x)} \left[\hat{R}_n^p(\pi) \right].$$

We have the following properties:

Proposition 5.2.1. *The bias of the estimator:*
The bias of this estimator writes:

$$\begin{aligned}\mathcal{B}^p(\pi) &= R^p(\pi) - R(\pi)\\ &= \mathbb{E}_{x\sim\nu, a\sim\pi_0(\cdot|x)}\left[\left(\frac{1}{p(\pi_0(a|x))} - \frac{1}{\pi_0(a|x)}\right)\pi(a|x)c(a,x)\right] \geq 0\end{aligned}$$

The bias is positive, thus we have:

$$R(\pi) \leq R^p(\pi), \tag{5.1}$$

recovering equality when using the `IPS` *estimator, i.e.* $p : x \rightarrow x$ *the identity function.*

Equation 5.1 is a needed inequality for our analysis in order to obtain upper bounds on the true risk. Indeed, the idea is that any upper bound on the expectation of our estimator is an upper bound on the true risk. This enables us to focus on the analysis of the expectation of our estimator. With the estimator chosen, we provide our first result in the next section.

5.3 A Refined PAC-Bayesian Analysis

Now that we defined the estimator covered by our study, we attack the problem of deriving generalization bounds. We begin by stating the important change of measure lemma:

Lemma 5.3.1. *Change of measure:*
Let f be a function of the parameter θ and data $\mathcal{D}_n$, for any distribution Q that is P continuous, for any $\delta \in (0,1]$, we have with probability $1-\delta$:

$$\mathbb{E}_{\theta\sim Q}[f(\theta,\mathcal{D}_n)] \leq \mathcal{KL}(Q||P) + \ln\frac{\Psi_f}{\delta} \tag{5.2}$$

with $\Psi_f = \mathbb{E}_{\mathcal{D}_n}\mathbb{E}_{\theta\sim P}[e^{f(\theta,\mathcal{D}_n)}]$.

Lemma 5.3.1 is the backbone of a multitude of PAC-Bayesian bounds. With this result, the recipe of constructing a generalisation bound reduces to choosing an adequate function f for which we can control Ψ_f. We want to construct fully empirical generalisation bounds that are tight enough to compete with our previous results. We derive a **novel**, empirical high order bound expressed in the following:

Proposition 5.3.1. *Empirical High Order PAC-Bayes bound:*
Let $K \geq 1$. Given a prior P on $\mathcal{F}_\Theta$, $\delta \in (0,1]$ and $\lambda > 0$, the following bound holds with probability at least $1-\delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:

$$R(\pi_Q) \leq \hat{R}^p_n(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} + \sum_{k=2}^{2K}\frac{\lambda^{k-1}}{k}\hat{\mathcal{M}}^{p,k}_n(\pi_Q) \tag{5.3}$$

with:

$$\hat{\mathcal{M}}^{p,k}_n(\pi_Q) = \frac{1}{n}\sum_{i=1}^{n}\frac{\pi_Q(a_i|x_i)}{p(\pi_0(a_i|x_i))^k}c_i^k.$$

Proof. Let $K \geq 1$ and f_K be the following function:

$$f_K(x) = \frac{\log(1+x) - \sum_{k=1}^{K} \frac{(-1)^{k-1}}{k} x^k}{(-1)^K x^{K+1}}$$

We begin by demonstrating that f_K is a decreasing function in $\mathbb{R}^+$. Let $x \in \mathbb{R}^+$, we have the following identity holding $\forall t > 0$ and $\forall n \geq 0$:

$$\frac{1 + (-1)^n t^{n+1}}{1+t} = \sum_{k=0}^{n} (-1)^k t^k \iff \frac{1}{1+t} = \sum_{k=0}^{n} (-1)^k t^k + \frac{(-1)^{n+1} t^{n+1}}{1+t}.$$

Combined with the integral form of the log:

$$\log(1+x) = \int_0^x \frac{1}{1+t} dt,$$

we show that the numerator is equal to:

$$\log(1+x) - \sum_{k=1}^{K} \frac{(-1)^{k-1}}{k} x^k = (-1)^K \int_0^x \frac{t^K}{1+t} dt$$

which rewrites the function f_K to:

$$f_K(x) = \frac{1}{x^{K+1}} \int_0^x \frac{t^K}{1+t} dt.$$

Using the change of variable $t = ux$, we obtain:

$$f_K(x) = \int_0^1 \frac{u^K}{1+xu} dt$$

which is clearly decreasing for $x \in \mathbb{R}^+$. Now for $K \geq 1$, we have for a positive random variable $X \geq 0$ and $\lambda > 0$:

$$f_{2K-1}(0) = \frac{1}{2K} \geq f_{2K-1}(\lambda X) = -\frac{\log(1+\lambda X) - \sum_{k=1}^{2K-1} \frac{(-1)^{k-1}}{k} (\lambda X)^k}{(\lambda X)^{2K}}$$

which is equivalent to:

$$\begin{aligned}
\sum_{k=1}^{2K} \frac{(-1)^{k-1}}{k} (\lambda X)^k \leq \log(1+\lambda X) &\iff \exp\left(\sum_{k=1}^{2K} \frac{(-1)^{k-1}}{k} (\lambda X)^k\right) \leq 1 + \lambda X \\
&\implies \mathbb{E}\left[\exp\left(\sum_{k=1}^{2K} \frac{(-1)^{k-1}}{k} (\lambda X)^k\right)\right] \leq 1 + \mathbb{E}\left[\lambda X\right] \\
&\implies \mathbb{E}\left[\exp\left(\sum_{k=1}^{2K} \frac{(-1)^{k-1}}{k} (\lambda X)^k\right)\right] \leq \exp\left(\mathbb{E}\left[\lambda X\right]\right) \\
&\implies \mathbb{E}\left[\exp\left(\lambda(X - \mathbb{E}\left[X\right]) + \sum_{k=2}^{2K} \frac{(-1)^{k-1}}{k} (\lambda X)^k\right)\right] \leq 1.
\end{aligned}$$

For any $X \leq 0$, we can inject $-X \geq 0$ to obtain:

$$\forall X \leq 0, \quad \mathbb{E}\left[\exp\left(\lambda(\mathbb{E}\left[X\right] - X) - \sum_{k=2}^{2K} \frac{1}{k} (\lambda X)^k\right)\right] \leq 1. \tag{5.4}$$

Let $\lambda > 0$. The adequate function f we are going to use in combination with Lemma 5.3.1 is:

$$\begin{aligned} f(\theta, \mathcal{D}_n) &= \sum_{i=1}^{n} \lambda \left(R^p(d_\theta) - \frac{d_\theta(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i \right) - \sum_{k=2}^{2K} \frac{1}{k} \left(\lambda \frac{d_\theta(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i \right)^k \\ &= \sum_{i=1}^{n} \lambda \left(R^p(d_\theta) - \frac{d_\theta(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i \right) - \sum_{k=2}^{2K} \frac{d_\theta(a_i|x_i)}{k} \left(\frac{\lambda}{p(\pi_0(a_i|x_i))} c_i \right)^k . \end{aligned}$$

By exploiting the i.i.d. nature of the data and exchanging the order of expectations (P is independent of $\mathcal{D}_n$), we can naturally prove using (5.4) that:

$$\psi_f = \mathbb{E}_P \left[\prod_{i=1}^{n} \mathbb{E} \left[\exp \left(\lambda \left(R^p(d_\theta) - \frac{d_\theta(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i \right) - \sum_{k=2}^{2K} \frac{1}{k} \left(\lambda \frac{d_\theta(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i \right)^k \right) \right] \right] \leq 1,$$

as we have :

$$\frac{d_\theta(a_i|x_i)}{p(\pi_0(a_i|x_i))} c_i \leq 0 \quad \forall i.$$

Injecting ψ_f in Lemma 5.3.1, rearranging terms and using (5.1) concludes the proof. ■

To the best of our knowledge, this is the first time an empirical, high order PAC-Bayes bound was derived. The particularity of this bound is that it does not necessarily suppose the existence of the moments to be valid, as all quantities are finite sums. The value of K determines which (even) moment we want to stop. At a first glance and for some particular values of λ, it seems that increasing K makes the bound tighter but increases the computation needed to evaluate the bound. We prove the following result linking the value of K to the tightness of the bound:

Proposition 5.3.2. *Impact of K on the tightness of the bound:*
Let P a prior, Q a distribution on $\mathcal{F}_\Theta$, $\delta \in (0, 1]$ and a $\lambda > 0$. For any $K \geq 1$, let $B_K(\pi_Q)$ be defined as:

$$B_K(\pi_Q) = \hat{R}^p_n(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln \frac{1}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_n(\pi_Q).$$

Then:

$$\lambda < \min_{i \in [n]} \left\{ \left(\frac{2K+2}{2K+1} \right) \frac{p(\pi_0(a_i|x_i))}{|c_i|} \right\} \implies B_{K+1}(\pi_Q) < B_K(\pi_Q) \tag{5.5}$$

which implies that:

$$\lambda < \min_{i \in [n]} \left\{ \frac{p(\pi_0(a_i|x_i))}{|c_i|} \right\} \implies B_K(\pi_Q) \textit{ is a strictly decreasing function w.r.t } K.$$

Proof. We want to prove the implication (5.5) from which the condition on the decreasing nature of our bound will follow. Indeed, Let us suppose that (5.5) is true, we have:

$$\begin{aligned} \lambda < \min_{i \in [n]} \left\{ \frac{p(\pi_0(a_i|x_i))}{|c_i|} \right\} &\implies \forall K \geq 1, \quad \lambda < \min_{i \in [n]} \left\{ \left(\frac{2K+2}{2K+1} \right) \frac{p(\pi_0(a_i|x_i))}{|c_i|} \right\} \\ &\implies \forall K \geq 1, \quad B_{K+1}(\pi_Q) < B_K(\pi_Q) \quad \text{(Using (5.5))} \\ &\implies B_K(\pi_Q) \text{ is a strictly decreasing function w.r.t } K. \end{aligned}$$

Now let us prove the implication in (5.5). Let $p_i = \pi_0(a_i|x_i)$, we have for any $K \geq 1$:

$$\begin{aligned} B_{K+1}(\pi_Q) < B_K(\pi_Q) &\iff \sum_{k=2K+1}^{2K+2} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}_n^{p,k}(\pi_Q) \leq 0 \\ &\iff \frac{\lambda^{2K}}{n} \sum_{i=1}^{n} \pi_Q(a_i|x_i) \left(\frac{c_i}{p_i}\right)^{2K+1} \left(\frac{1}{2K+1} + \frac{\lambda c_i}{(2K+2)p_i}\right) \leq 0 \end{aligned}$$

As $c_i \leq 0$, we can assure this inequality by choosing a λ that verifies:

$$\forall i \in [n], \quad \lambda < \left\{ \left(\frac{2K+2}{2K+1}\right) \frac{p_i}{|c_i|} \right\} \iff \lambda < \min_{i \in [n]} \left\{ \left(\frac{2K+2}{2K+1}\right) \frac{p_i}{|c_i|} \right\}$$

which concludes the proof of (5.5). ■

From this proposition, we obtain a *sufficient* condition for which our bound gets tighter with K. This condition implies a hard constraint on λ, being smaller than a data-dependent value. This result, even if it is weak, may give us some insight into the bound behaves. One can deduce that the bigger K, the harder the constraint on λ; for example, for any π_Q, it is easier to have $B_2(\pi_Q) < B_1(\pi_Q)$ than $B_{100}(\pi_Q) < B_{99}(\pi_Q)$. A bigger λ, in the other hand, means that we can move even further from the prior. This can guide us towards the following; If our prior is close to the optimal distribution Q, using a big K can be beneficial, in the other hand, if our prior is far from the optimal solution, a smaller K might be better. In practice, we do not know how close we are to the optimal policy, and we do not know if the best values for λ will verify the conditions in Proposition 5.3.2. This means that determining the tightest bound w.r.t K will depend on the application. To this end, let us study two particular cases of this bound; its value when $K = 1$ and when $K \to \infty$.

Empirical Second Moment Bound. With $K = 1$, we obtain the following:

Corollary 5.3.1. *Second Moment Upper bound:*
Given a prior P on $\mathcal{F}_\Theta$, $\delta \in (0, 1]$ and $\lambda > 0$. The following bound holds with probability at least $1 - \delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:

$$R(\pi_Q) \leq \hat{R}_n^p(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln \frac{1}{\delta}}{\lambda n} + \frac{\lambda}{2} \hat{\mathcal{S}}_n^p(\pi_Q) \tag{5.6}$$

with:

$$\hat{\mathcal{S}}_n^p(\pi_Q) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_Q(a_i|x_i)}{p(\pi_0(a_i|x_i))^2} c_i^2$$

With $K = 1$, our bound looks similar to the one derived in Mhammedi et al. (2019). However, our result is slightly tighter as it drops the multiplicative factor on the second moment. We can compare this result to previously derived PAC-Bayesian bounds for off-policy learning. We start by writing down the conditional Bernstein bound of Proposition 5.1.1 for the (linear) `cIPS` $(p : x \to \max(x, \tau))$. For a policy π_Q and a $\lambda > 0$, we have:

$$R(\pi_Q) \leq \hat{R}_n^\tau(\pi_Q) + \sqrt{\frac{\mathcal{KL}(Q||P) + \ln \frac{4\sqrt{n}}{\delta}}{2n}} + \frac{\mathcal{KL}(Q||P) + \ln \frac{2}{\delta}}{\lambda n} + \lambda g\left(\lambda/\tau\right) \mathcal{V}_n^\tau(\pi_Q). \tag{C-Bern}$$

$$R(\pi_Q) \leq \hat{R}_n^\tau(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln \frac{1}{\delta}}{\lambda n} + \frac{\lambda}{2} \hat{\mathcal{S}}_n^\tau(\pi_Q). \tag{K = 1}$$

We can observe that the previously derived conditional Bernstein bound has several terms that make it less tight:

- It has an additional, strictly positive square root KL divergence term.
- The multiplicative factor $g(\lambda/\tau)$ is always bigger than $1/2$, and diverges when $\tau \to 0$.
- With enough data ($n \gg 1$), we also have:

$$\hat{\mathcal{S}}_n^\tau(\pi_Q) \approx \mathbb{E}\left[\frac{\pi_Q(a|x)}{\max\{\pi_0(a|x),\tau\}^2}c(a,x)^2\right] \leq \mathbb{E}\left[\frac{\pi_Q(a|x)}{\max\{\pi_0(a|x),\tau\}^2}\right] \approx \mathcal{V}_n^\tau(\pi_Q).$$

These observations confirm that the new bound derived with $K = 1$ is tighter than what was previously proposed for cIPS, especially when $n \gg 1$. As our bound can work for other estimators, we also compare it to a recently proposed PAC-Bayes bound in Aouali et al. (2023a) for the exponentially-smoothed estimator ($p : x \to x^\alpha$) with $\alpha \in [0, 1]$:

$$R(\pi_Q) \leq \hat{R}_n^\alpha(\pi_Q) + \sqrt{\frac{\mathcal{KL}(Q||P) + \ln\frac{4\sqrt{n}}{\delta}}{2n}} + \frac{\mathcal{KL}(Q||P) + \ln\frac{2}{\delta}}{\lambda n} + \frac{\lambda}{2}\left(\mathcal{V}_n^\alpha(\pi_Q) + \hat{\mathcal{S}}_n^\alpha(\pi_Q)\right). \tag{α-Smooth}$$

$$R(\pi_Q) \leq \hat{R}_n^\alpha(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} + \frac{\lambda}{2}\hat{\mathcal{S}}_n^\alpha(\pi_Q). \tag{K = 1}$$

We can clearly see that the previously proposed bound for the exponentially smoothed estimator has two additional positive quantities that makes it less tight than our bound. In addition, computing our bound does not rely on expectations under π_0 (contrary to the previous bounds that have $\mathcal{V}_n$) which alleviates the need to access the logging policy and reduce the computations. Another particularity of our bound is that it works for a large family of estimators, parameterized by the function p. A natural question to ask is what is the form of p that gives the tightest results? The answer to this question depends on the form of the bound we consider. Let us keep focusing on the case of $K = 1$. We have the following result:

Proposition 5.3.3. *Optimal p for $K = 1$:*
For a fixed prior P on $\mathcal{F}_\Theta$, $\delta \in (0, 1]$ and a $\lambda > 0$. The function p that minimizes the bound for $K = 1$, giving the tightest result is:

$$\forall i, \quad p_i = p(\pi_0(a_i, x_i)) = \max\{\pi_0(a_i|x_i), \lambda|c_i|\}. \tag{5.7}$$

This means that when the costs are binary, we obtain the classical cIPS estimator, with $p_i = \max\{\pi_0(a_i|x_i), \lambda\}$.

Proof. We want to look for the value of p that minimize the bound. Formally, by fixing all variables of the bound, this problem reduces to:

$$\arg\min_{p\in C_1} \hat{R}_n^p(\pi_Q) + \frac{\lambda}{2}\hat{\mathcal{S}}_n^p(\pi_Q) = \arg\min_{p\in C_1} \frac{1}{n}\sum_{i=1}^n \pi_Q(a_i|x_i)\left(\frac{1}{p(\pi_0(a_i|x_i))}c_i + \frac{\lambda}{2p(\pi_0(a_i|x_i))^2}c_i^2\right).$$

The objective decomposes across data points. For any $i \in [n]$, we set:

$$l_i = \frac{1}{p_i} = \frac{1}{p(\pi_0(a_i|x_i))}.$$

Let us fix a $j \in [n]$, if $\pi_Q(a_j|x_j)c_j = 0$, then any l_j is a minimizer.

In the other case, the following problem:

$$\begin{aligned}&\underset{l_j\in\mathbb{R}}{\arg\min}\, \hat{R}^p_n(\pi_Q) + \frac{\lambda}{2}\hat{\mathcal{S}}^p_n(\pi_Q) = \underset{l_j\in\mathbb{R}}{\arg\min}\, \frac{1}{n}\sum_{i=1}^{n}\pi_Q(a_i|x_i)\left(l_i c_i + \frac{\lambda}{2}l_i^2 c_i^2\right)\\ &\text{subject to} \quad l_j \leq \frac{1}{\pi_0(a_j|x_j)}\end{aligned}$$

is strongly convex in l_j. We write the KKT conditions for l_j to be optimal; there exists α^* that verifies:

$$c_j + 2\lambda c_j^2 l_j + \alpha^* = 0 \tag{5.8}$$

$$\alpha^* \geq 0 \tag{5.9}$$

$$\alpha^*\left(l_j - \frac{1}{\pi_0(a_j|x_j)}\right) = 0 \tag{5.10}$$

$$l_j \leq \frac{1}{\pi_0(a_j|x_j)} \tag{5.11}$$

We study the two following two cases:

1. $$\boxed{l_j \geq \frac{1}{\lambda\left|c_j\right|} \iff p_j \leq \lambda\left|c_j\right| :}$$

   we have $\alpha^* = |c_j| - \lambda c_j^2 l_j \leq 0 \implies \alpha^* = 0$, meaning that:

   $$l_j = \frac{1}{\lambda\left|c_j\right|} \iff p_j = \lambda\left|c_j\right|$$

2. $$\boxed{l_j < \frac{1}{\lambda\left|c_j\right|} \iff p_j > \lambda\left|c_j\right| :}$$

   we have $\alpha^* = |c_j| - \lambda c_j^2 l_j > 0$, which combined to condition (4.9) gives:

   $$l_j = \frac{1}{\pi_0(a_j|x_j)} \iff p_j = \pi_0(a_j|x_j).$$

The two results combined mean that we always have:

$$p_j \geq \lambda\left|c_j\right|, \text{ with } p_j > \lambda\left|c_j\right| \implies p_j = \pi_0(a_j|x_j).$$

We deduce that p_j has the following form when $c_j \neq 0$:

$$p_j = p(\pi_0(a_j|x_j)) = \max\left\{\lambda\left|c_j\right|, \pi_0(a_j|x_j)\right\} \tag{5.12}$$

$$\alpha^* = |c_j| - \lambda\frac{c_j^2}{\max\left\{2\lambda\left|c_j\right|, \pi_0(a_j|x_j)\right\}} \tag{5.13}$$

These values verify the KKT conditions. As the problem is strongly convex, p_j has a unique possible value and must be equal to equation (4.11). The form of p_j resembles an adaptive version of `cIPS`. In the case where the cost function c is binary:

$$\forall i \quad c_i \in \{-1, 0\},$$

we recover the classical `cIPS` as an optimal solution for p:

$$p_j = \max\left\{\lambda, \pi_0(a_j|x_j)\right\}.$$

■

This result strengthens the use of the (linear) `cIPS` estimator for policy learning as it is found to be optimal in the sense of minimizing the empirical second moment PAC-Bayesian bound. We also want to explore the more extreme case when increasing K towards infinity.

Limit Upper Bound. We tend K towards infinity to obtain the following:

Corollary 5.3.2. *Limit Upper Bound:*
Given a prior P on $\mathcal{F}_\Theta$, $\delta \in (0,1]$ and a controlled value of $\lambda > 0$ so as:

$$\lambda < \min_{i \in [n]} \left\{ \frac{p(\pi_0(a_i|x_i))}{|c_i|} \right\}.$$

The following bound holds with probability at least $1-\delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:

$$R(\pi_Q) \leq -\frac{1}{n}\sum_{i=1}^{n} \frac{\pi_Q(a_i|x_i)}{\lambda} \log\left(1 - \frac{\lambda c_i}{p(\pi_0(a_i|x_i))}\right) + \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} \tag{5.14}$$

Proof. For any fixed K, we have the following bound holding with high probability:

$$\begin{aligned}
R(\pi_Q) &\leq \hat{R}_n^p(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_Q(a_i|x_i)}{p(\pi_0(a_i|x_i))^k} c_i^k \\
&\leq \hat{R}_n^p(\pi_Q) + \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} + \frac{1}{\lambda n} \sum_{i=1}^{n} \pi_Q(a_i|x_i) \sum_{k=2}^{2K} \frac{1}{k} \left(\frac{\lambda c_i}{p(\pi_0(a_i|x_i))}\right)^k \\
&\leq \frac{\mathcal{KL}(Q||P) + \ln\frac{1}{\delta}}{\lambda n} + \frac{1}{\lambda n} \sum_{i=1}^{n} \pi_Q(a_i|x_i) \sum_{k=1}^{2K} \frac{1}{k} \left(\frac{\lambda c_i}{p(\pi_0(a_i|x_i))}\right)^k.
\end{aligned}$$

Let S_i^K be defined:

$$S_i^K = \sum_{k=1}^{2K} \frac{1}{k} \left(\frac{\lambda c_i}{p(\pi_0(a_i|x_i))}\right)^k.$$

For all series S_i^K to converge, we need to choose a $\lambda < \min_{i \in [n]} \left\{ \frac{p(\pi_0(a_i|x_i))}{|c_i|} \right\}$. Having this condition insures that:

$$\forall i \in [n], \quad \lim_{K \to \infty} S_i^K = -\log\left(1 - \frac{\lambda c_i}{p(\pi_0(a_i|x_i))}\right)$$

Injecting this result in the bound ends the proof. ■

To the best of our knowledge, this is the first time a bound of this form was suggested in the context of off-policy learning. A bound of a similar nature was presented in (Alquier, 2021, Theorem 5.2) for unbounded losses, our proof technique however is different, and the connexion between the bounds will be explored in the future. The expression of the obtained bound is simple, it is easy to interpret, but puts a hard constraint on λ making it less flexible than the previously derived bound for $K \in \mathbb{N}$. We will however investigate its potential use and tightness in the experimental section. As in the case of $K = 1$, the question of finding the optimal p for this bound is interesting, and we provide the following proposition as an answer:

Proposition 5.3.4. *Optimal p for $K \to \infty$:*
For a fixed prior P on $\mathcal{F}_\Theta$, $\delta \in (0, 1]$ and a $\lambda > 0$. The function p that minimizes the bound for $K \to \infty$, giving the tightest result is:

$$\forall i, \quad p_i = p(\pi_0(a_i|x_i)) = \pi_0(a_i|x_i) \tag{5.15}$$

recovering the `IPS` estimator.

Proof. The proof of this proposition is quite simple. The functions:

$$f_i(x) = -\log\left(1 - \frac{\lambda c_i}{x}\right)$$

are increasing. As our function p respects the condition $\forall x, \quad p(x) \geq x$, $p : x \to x$ gives the tightest result, recovering the `IPS` estimator as an optimal choice for this bound. ■

In this chapter, we wanted to loosen the requirements used for our theoretically grounded strategy, which previously needed access to π_0 to compute important statistics in our tightest bound and to construct the prior P. Now that we have suggested new generalization bounds that do not use π_0, and are potentially tighter than what was previously suggested as shown with the example of $K = 1$, our aim is to propose a procedure to obtain bounds that are tight empirically without constructing P with the help of the logging policy.

5.4 Unknown Structure of the Logging Policy

We use the procedure of Mhammedi et al. (2019) to construct priors that are data-dependent, alleviating the requirement of having access to π_0, thus, to a good prior P. The underlying idea resembles cross-validation (Arlot and Celisse, 2010). We split the logged data $\mathcal{D}_n$ into two disjoint subsets S_1 and S_2 with $\mathcal{D}_n = S_1 \cup S_2$. These two subsets are used to learn two priors P_1 from S_1 and P_2 from S_2. As P_1 (respectively P_2) is learned using S_1 (respectively S_2), with the i.i.d. assumption on how data is collected, P_1 is independent of S_2 and the same is true for P_2 and S_1. This independence help us define two bounds, one on S_1 with the prior P_2, the second one on S_2 with the prior P_1, these two bounds are combined using a union argument to construct a bound on the whole logged dataset D_n, with two data-dependent priors P_1 and P_2, making access to π_0 unnecessary. Using this idea with Proposition 5.3.1, we give the following result:

Proposition 5.4.1. *CV-type Empirical High Order PAC-Bayes bound:*
Let $K \geq 1$. Given a split $\mathcal{D}_n = S_1 \cup S_2$, and learned priors P_1 on S_1 and P_2 on S_2, a $\delta \in (0, 1]$ and $\lambda > 0$. The following bound holds with probability at least $1 - \delta$ uniformly for all distribution Q over $\mathcal{F}_\Theta$:

$$R(\pi_Q) \leq \hat{R}^p_n(\pi_Q) + \frac{\mathcal{KL}(Q||P_1) + \mathcal{KL}(Q||P_2) + \ln\frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_n(\pi_Q). \tag{5.16}$$

Proof. Let $\mathcal{D}_n = S_1 \cup S_2$, a disjoint partition with $Card(S_1) = n_1$ and $Card(S_2) = n_2$. Let P_1 (respectively P_2) a prior trained on S_1 (respectively S_2). As the priors are constructed using S_1 and S_2. We state Proposition 5.3.1 for S_1 with P_2 and for S_2 with P_1. We obtain two bounds holding simultaneously with probability $1 - \delta/2$ for all Q:

$$n_2 R(\pi_Q) \leq n_2 \hat{R}^p_{n_2}(\pi_Q) + \frac{\mathcal{KL}(Q||P_1) + \ln \frac{2}{\delta}}{\lambda} + n_2 \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_{n_2}(\pi_Q)$$

$$n_1 R(\pi_Q) \leq n_1 \hat{R}^p_{n_1}(\pi_Q) + \frac{\mathcal{KL}(Q||P_2) + \ln \frac{2}{\delta}}{\lambda} + n_1 \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_{n_1}(\pi_Q).$$

We take a union of the bounds and sum the two inequalities to obtain a new bound holding with probability $1 - \delta$ for all distributions Q:

$$R(\pi_Q) \leq \hat{R}^p_n(\pi_Q) + \frac{\mathcal{KL}(Q||P_1) + \mathcal{KL}(Q||P_2) + \ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_n(\pi_Q)$$

ending the proof. ■

In the previous form of the bound, we had the term $\mathcal{KL}(Q||P)$ penalizing how far our new distribution Q is from the prior P. By fixing P to the distribution inducing the logging policy, this term quantified how the new policy π_Q is far from π_0. For **LIG** policies and given an (PAC-Bayesian) upper bound $\mathcal{UB}_n$, the old procedure of learning the new candidate policy $\pi_{\texttt{new}}$ looked like this:

```
Algorithm 1: Learning with access to $\pi_0$
    Inputs: Logged dataset $\mathcal{D}_n$, Logging policy $\pi_0$.
    Initialise: $P$ such as $\pi_P = \pi_0$, $\alpha = \alpha_0$ (all other parameters of the bound, $\tau$ and $\delta$ for
      example).
    Compute:
        $\pi_{\texttt{new}} = \arg\min_{\pi_Q \in \Pi} \mathcal{UB}_n(\pi_Q, P, \alpha_0)$
      return $\pi_{\texttt{new}}$.
```

With the new form of the bound in Proposition 5.4.1, the term $\mathcal{KL}(Q||P_1) + \mathcal{KL}(Q||P_2)$ has a different interpretation. It quantifies how stable our learning algorithm of Q, P_1 and P_2 is; for this term to be small, we need to have $Q \approx P_1 \approx P_2$. This means that the learned P_1 on S_1 should not be different from the learned P_2 on S_2. In addition, as our purpose is to obtain the tightest/smallest upper bound, the priors P_1 and P_2 should verify the following:

1. The bound needs to be small when $Q = P_1$:

$$\hat{R}^p_n(\pi_{P_1}) + \frac{\mathcal{KL}(P_1||P_2) + \ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_n(\pi_{P_1}) \quad \text{should be small.}$$

2. The bound needs to be small when $Q = P_2$:

$$\hat{R}^p_n(\pi_{P_2}) + \frac{\mathcal{KL}(P_2||P_1) + \ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_n(\pi_{P_2}) \quad \text{should be small.}$$

From these two conditions, we want both P_1 and P_2 to induce a small risk (and empirical moments), while both P_1 and P_2 being close (in the Jensen-Shannon divergence sense). As P_1 (respectively P_2) can only depend on S_1 (respectively S_2), these observations give us an idea on how to construct both P_1 and P_2. We want to:

- Construct a prior P_1 that depends on S_1 ($Card(S_1) = n_1$) for which:

$$\hat{R}^p_{n_1}(\pi_{P_1}) + \frac{\ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_{n_1}(\pi_{P_1}) \quad \text{is small.}$$

- Construct a prior P_2 that depends on S_2 ($Card(S_2) = n_2$) for which:

$$\hat{R}^p_{n_2}(\pi_{P_2}) + \frac{\ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{p,k}_{n_2}(\pi_{P_2}) \quad \text{is small.}$$

- Both P_1 and P_2 should be close, which can be achieved by regularizing them towards a carefully chosen distribution P_0.

For **LIG** policies (Equation (4.6)) with distributions $\left\{\mathcal{N}(\mu, I_d), \mu \in \mathbb{R}^d\right\}$, previous observations motivate the following learning strategy:

```
Algorithm 2: Learning with CV-type bound for LIG policies
  Inputs: Logged dataset $\mathcal{D}_n$.
  Initialise:
    • Strategy parameters: a reference mean $\mu_0$, a regularization parameter $\beta$.
    • Bound parameters: the parameter $K$, Lambda set $\Lambda$, clipping parameter $\tau$ and
      tolerance $\delta$.
  Split: $\mathcal{D}_n = S_1 \cup S_2$ with $n_1 = n_2 = n/2$.
  Optimize:
    • Train on $S_1$:
        $\mu_1 = \arg\min_{\mu \in \mathbb{R}^d} \min_{\lambda > 0} \left\{ \hat{R}^\tau_{n_1}(\pi_\mu) + \frac{\ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{\tau,k}_{n_1}(\pi_\mu) + \beta ||\mu - \mu_0||_2^2 \right\}.$
    • Train on $S_2$:
        $\mu_2 = \arg\min_{\mu \in \mathbb{R}^d} \min_{\lambda > 0} \left\{ \hat{R}^\tau_{n_2}(\pi_\mu) + \frac{\ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{\tau,k}_{n_2}(\pi_\mu) + \beta ||\mu - \mu_0||_2^2 \right\}.$
    • Minimize the bound on $\mathcal{D}_n$:
        $\hat{\mu} = \arg\min_{\mu \in \mathbb{R}^d} \min_{\lambda \in \Lambda} \left\{ \hat{R}^\tau_n(\pi_\mu) + \frac{\ln \frac{2n_\Lambda}{\delta}}{\lambda n} + \frac{||\mu - \mu_1||_2^2 + ||\mu - \mu_2||_2^2}{2n\lambda} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{\tau,k}_n(\pi_\mu) \right\}.$
  return $\pi_{\hat{\mu}}$.
```

The new strategy developed in Algorithm 2 bypasses the use of a prior specified to match π_0. It does require however a reference mean μ_0 and a hyperparameter β that controls how close the learned priors (learned means) should be. This reference can be set to be the null vector $\mu_0 = \mathbf{0}$. Ideally, μ_0 should be informative and penalize our policies towards a good region of the space. Previously, we had access to π_0 and its parameter helped us learn new policies. An idea that can be explored in the absence of π_0 is to use S_1 and S_2 to imitate π_0 (London and Sandler, 2019). Imitating π_0 with S_1 results in μ_{01} and imitating π_0 with S_2 results in μ_{02}. As imitation learning do not suffer from high variance, we will have $\mu_{01} \approx \mu_{02}$. This observation motivates another strategy, described below:

```
Algorithm 3: Learning with CV-type bound with Imitation for LIG policies
  Inputs: Logged dataset $\mathcal{D}_n$.
  Initialise:
    • Strategy parameters: a regularization parameter $\beta$.
    • Bound parameters: the parameter $K$, Lambda set $\Lambda$, clipping parameter $\tau$ and tolerance $\delta$.
  Split: $\mathcal{D}_n = S_1 \cup S_2$ with $n_1 = n_2 = n/2$.
  Optimize:
    • Train on $S_1$:
        $\mu_{01} = \arg\min_{\mu \in \mathbb{R}^d} \left\{ -\frac{1}{n_1} \sum_{i \in S_1} \log \pi_\mu(a_i|x_i) + \frac{1}{n_1} ||\mu||_2^2 \right\}$   Imitate $\pi_0$ with $S_1$.
        $\mu_1 = \arg\min_{\mu \in \mathbb{R}^d} \min_{\lambda > 0} \left\{ \hat{R}^\tau_{n_1}(\pi_\mu) + \frac{\ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{\tau,k}_{n_1}(\pi_\mu) + \beta ||\mu - \mu_{01}||_2^2 \right\}.$
    • Train on $S_2$:
        $\mu_{02} = \arg\min_{\mu \in \mathbb{R}^d} \left\{ -\frac{1}{n_2} \sum_{i \in S_2} \log \pi_\mu(a_i|x_i) + \frac{1}{n_2} ||\mu||_2^2 \right\}$   Imitate $\pi_0$ with $S_2$.
        $\mu_2 = \arg\min_{\mu \in \mathbb{R}^d} \min_{\lambda > 0} \left\{ \hat{R}^\tau_{n_2}(\pi_\mu) + \frac{\ln \frac{2}{\delta}}{\lambda n} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{\tau,k}_{n_1}(\pi_\mu) + \beta ||\mu - \mu_{02}||_2^2 \right\}.$
    • Minimize the bound on $\mathcal{D}_n$:
        $\hat{\mu} = \arg\min_{\mu \in \mathbb{R}^d} \min_{\lambda \in \Lambda} \left\{ \hat{R}^\tau_n(\pi_\mu) + \frac{\ln \frac{2n_\Lambda}{\delta}}{\lambda n} + \frac{||\mu - \mu_1||_2^2 + ||\mu - \mu_2||_2^2}{2n\lambda} + \sum_{k=2}^{2K} \frac{\lambda^{k-1}}{k} \hat{\mathcal{M}}^{\tau,k}_n(\pi_\mu) \right\}.$
  return $\pi_{\hat{\mu}}$.
```

This new strategy does not require setting a reference mean μ_0 but is computationally more expensive. We will investigate the the effectiveness of all the strategies presented in the experiments section.

5.5 Experiments

Experimental Setup. We follow the same experimental setup than the previous chapter. For ease of exposition, we use a softmax logging policy of parameter $\mu_0 \in \mathbb{R}^{p \times K}$:

$$\pi^{\mathcal{S}}_{\boldsymbol{\mu_0}}(a|x) \propto \exp(\phi(x)^T \boldsymbol{\mu_0}_a).$$

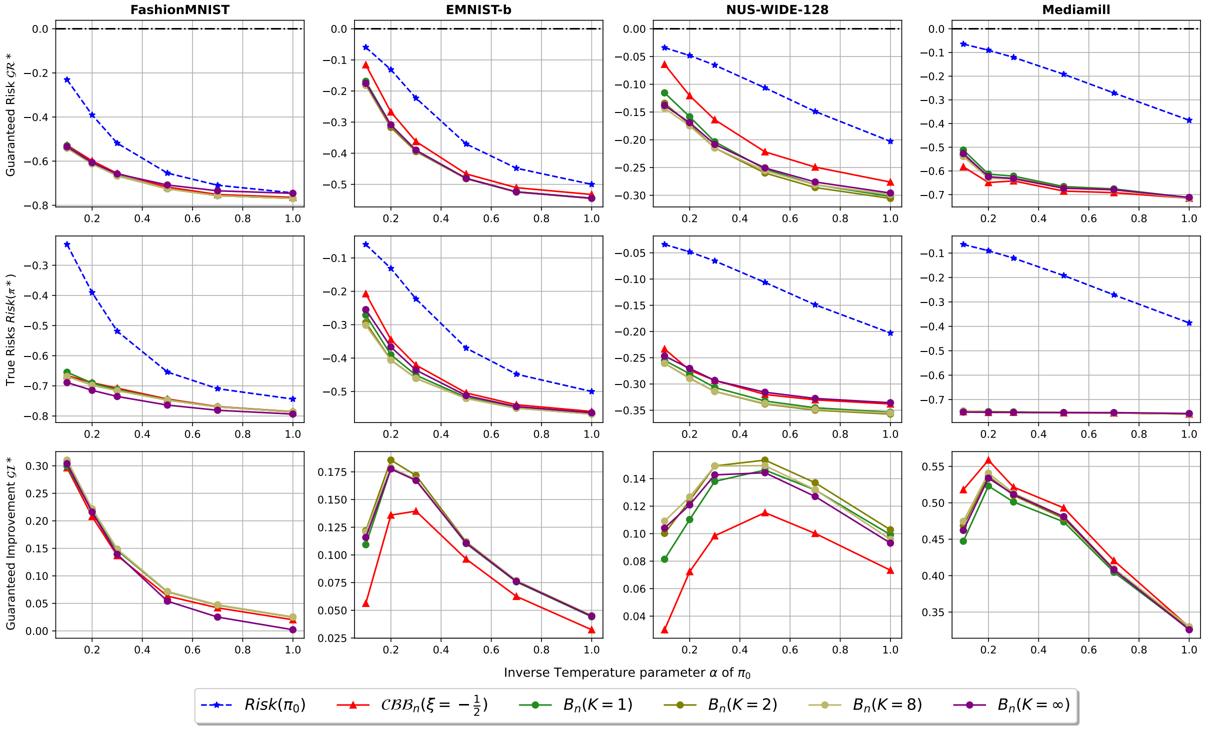


Figure 5.1: Behaviour of the guaranteed risk $\mathcal{GR}^*$ (↓ is better), the risk of the minimizer $R(\pi^*)$ (↓ is better) and the guaranteed improvement $\mathcal{GI}^*$ (↑ is better) given by the bounds (optimized with **LIG** policies) while changing π_0. We can observe that the new family of empirical moments bounds performs better than $\mathcal{CBB}^{\xi}_n(\xi = -\frac{1}{2})$ in most scenarios. K between 2 and 8 seems to gives the best results.

$\mu_{0_a} \in \mathbb{R}^p$ is the parameter associated with action a. The policy $\pi^{\mathcal{S}}_{\boldsymbol{\mu_0}}$ is used to generate the logged interactions' data $\mathcal{D}_n$. We adopt the standard supervised-to-bandit conversion to generate logged data in all of our experiments (Swaminathan and Joachims, 2015a). We use two mutliclass datasets: **FashionMNIST** (Xiao et al., 2017) and **EMNIST-b** (Cohen et al., 2017), alongside two multilabel datasets: **NUS-WIDE-128** (Chua et al., 2009) with 128-VLAD features (Spyromitros-Xioufis et al., 2014) and **Mediamill** (Snoek et al., 2006) to empirically validate our findings. The statistics of the datasets are described in Table 4.1 in Appendix 4.9.1; N the size of the training split, K the number of actions and p the dimension of the features $\phi(x)$. We take a small fraction (5%) of the training data that will only be used to learn $\boldsymbol{\mu_0}$ in a supervised manner. We introduce an inverse temperature parameter α to our softmax logging policy $\pi^{\mathcal{S}}_{\alpha\boldsymbol{\mu_0}}$. Changing α allows us to cover logging policies with different entropies ($\alpha \approx 0$ gives a uniform π_0 and $\alpha \approx 1$ gives a peaked π_0). We run $\pi_{\alpha\boldsymbol{\mu_0}}$ on the rest ($n_c = 0.95N$) of the training data to generate $\mathcal{D}_n$. For a context x and an action a, we define in our setting the cost as $c = -\mathbb{1}[a \in y]$ with y the set of true labels for x.

5.5.1 Tightness of the new bounds

As we proposed a new family of bounds, we want to measure how well they compare to the best performing bound so far, the $\mathcal{CBB}_n$ bound (with $\xi = -0.5$) defined in Proposition 5.1.1. To this end, we follow our old strategy to setting the prior P such as $P = \pi_0$ (Strategy described in Algorithm 1) and use **LIG** policies to optimize the bounds. We compare $\mathcal{CBB}_n(\xi = -0.5)$ to our empirical moment bound $B_n(K)$ for different values of $K \in \{1, 2, 8, \infty\}$.

We set the allowed uncertainty to $\delta = 0.05$. For all datasets, τ will be set to $\tau = 1/K$ to get no bias when the logging policy is close to uniform. We use Adam (Kingma and Ba, 2014) with a learning rate of 10^{-3} for 100 epochs to optimize the bounds w.r.t their parameters. The results of these experiments are shown in Figure 5.1.

We can observe that the new family of bounds achieves strong performance. As expected, these bounds give guaranteed risks that are lower than the risk of π_0, which means that they are capable of guaranteeing the improvement of the logging policy π_0. In addition, we can see that their performance is on par with the state-of-the-art $\mathcal{CBB}_n(\xi = -0.5)$ bound on **FashionMNIST** and **Mediamill**, while outperforming it on more difficult datasets (**EMNIST** and **NUS-WIDE-128**). These empirical results confirm the practicality of the newly proposed bounds. The new family achieves superior performance, while being fully empirical, contrary to $\mathcal{CBB}_n$ that requires access to π_0 to compute certain statistics. Another observation from these experiments is that there is, unfortunately, no value of K that always gives the best results. The condition described in Proposition 5.3.2 to have a decreasing bound with K is not respected as $K = \infty$ does not result in the tightest bound. In our experiments, it seems that values between $K = 2$ (stop at the 4th moment) and $K = 8$ (stop at the 16th moment) give the best results in terms of tightness of the bound and performance of the derived policies.

5.5.2 Effectiveness of the new strategies

After demonstrating the tightness of the newly proposed family of bounds, we want to test new strategies to guarantee improvement with PAC-Bayesian bounds in the case where access to π_0 is difficult. To this end, we choose our empirical moment bound $B_n(K = 2)$, as it is not computationally expensive and does not require access π_0. We optimize this bound over **LIG** policies, in all strategies adopted. We also fix the allowed uncertainty to $\delta = 0.05$ and set $\tau = 1/K$ in all scenarios. We compare the following approaches:

- **Strategy** 1 consisting of minimizing the bound while fixing the parameter μ_0 (the prior) such that $\pi_0 = \pi_{\mu_0}$. This approach assumes access to π_0.
- **Strategy** 2 with $\beta = 0.1$ and we fix the reference parameter μ_0 (the prior) such that $\pi_0 = \pi_{\mu_0}$. This approach assumes access to π_0.
- **Strategy** 2 with $\beta = 0.1$ and we fix the reference parameter $\mu_0 = \mathbf{0}$. This approach **does not** assume access to π_0.
- **Strategy** 3 with $\beta = 0.1$. We use Imitation learning and do not need a reference parameter μ_0. This approach **does not** assume access to π_0.

We plot all results in Figure 5.2. Our first observation is that adopting Strategy 1, which consists of minimizing the bound after fixing the prior to π_0 is not optimal. This strategy has the advantage of being computationally straightforward, but requires access to π_0 and can be largely improved. CV-type bounds (Proposition 5.4.1) help us define new strategies that result in better performing policies.

Strategy 2. This strategy learns two data-dependent priors (μ_1 and μ_2) while regularizing both of them towards a fixed, reference distribution μ_0. We test this strategy with μ_0 set such that $\pi_0 = \pi_{\mu_0}$ (assuming access to π_0) and with $\mu_0 = \mathbf{0}$ (no access to π_0). We first observe that this strategy, even without access to π_0, gives good results. We also observe that, as predicted, setting the reference parameter μ_0 to a good value helps this strategy improve the obtained policy. We can see that this strategy results in better performing policies than Strategy 1. These results suggest that even if we have access to π_0, we should prefer Strategy 2 (CV-type bound) over Strategy 1 (Data-free Prior) as the former dominates the latter in all scenarios.

Strategy 3. This strategy works with CV-type bounds and also learns two data-dependent priors (μ_1 and μ_2). For some additional computations, it gets rid of the need for specifying a reference parameter μ_0. Instead, it imitates the logging policy and learns a reference parameter using S_1 and S_2. This strategy gives the best guarantees, results in the best performing policies, and is on par with Strategy 2 (with access to π_0). If our application does not mind additional computations, this strategy should be adopted, as it gives the best results without assuming access to the policy π_0 or its structure.

CV-type strategies. All newly proposed strategies have the particularity to not only give superior policies, but their guaranteed risk is tight and is close to the real performance of the policies. For example, in the **EMNIST-b** dataset and for $\alpha = 0.1$, Strategy 1 gives a guaranteed risk that differs from the true risk by more than 0.1, which is not the case for the CV-type strategies which guarantee risks close to the true risk. This is true for all scenarios and is worth investigating in the future.

5.6 Conclusion

In this chapter, we derive a new family of bounds, based on a finer analysis of the moment generating function of commonly used estimators. Our analysis results in bounds with high order empirical moments, that are tractable and can be computed without access to π_0. We study the newly proposed bounds and show that they are tighter than the best bounds derived so far. In our pursuit to lessen the assumptions of our previous learning with performance guarantees approach, we suggest new learning strategies, that do not require access to the structure of π_0. We demonstrate empirically the superiority of the new strategies, as they require fewer assumptions and result in policies with greater performance guarantees. In the future, we want to theoretically investigate the strategies proposed. We are also interested in studying the tightness of the bounds to predict more accurately the performance of trained policies before deploying them. We believe that these research avenues will drive us closer to understanding the problem of learning policies offline.

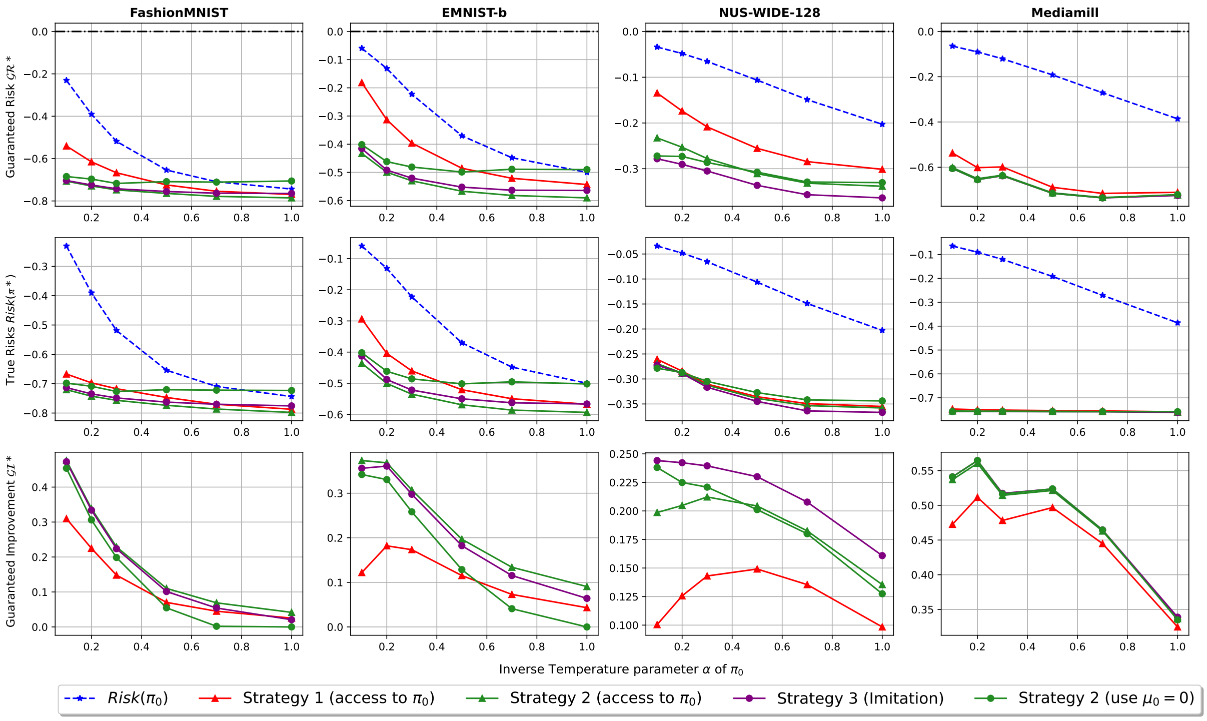


Figure 5.2: The effect of adopting different strategies to optimize the PAC-Bayesian bound ($B_n(K = 2)$).We plot the guaranteed risk $\mathcal{GR}^*$ (↓ is better), the risk of the minimizer $R(\pi^*)$ (↓ is better) and the guaranteed improvement $\mathcal{GI}^*$ (↑ is better) given by the different strategies for different logging policies. We can observe that adopting Strategy 1 (fixing the prior to π_0) is not the best. Strategies building on CV-type bounds give better guarantees and result in superior policies, even if we do not have access to π_0 (Strategy 3).

Part II

Offline Learning of Large Scale Decision Systems

Chapter 6

Scalable Bayesian Reward Modelling

Abstract

A common task for recommender systems is to build a profile of the interests of a user from items in their browsing history, and later to recommend items to the user from the same catalogue. The users' behaviour consists of two parts: the sequence of items that they viewed without intervention (the organic part) and the sequences of items recommended to them and their outcome (the bandit part). In this chapter, we propose *Bayesian Latent Organic Bandit model (BLOB)*, a probabilistic approach to combine the 'organic' and 'bandit' signals in order to improve the quality of recommendation. The bandit signal is valuable as it gives direct feedback of recommendation performance, but the signal quality is very uneven, as it is highly concentrated on the recommendations deemed optimal by the past version of the recommender system. In contrast, the organic signal is typically strong and covers most items, but is not always relevant to the recommendation task. In order to leverage the organic signal to efficiently learn the bandit signal in a Bayesian model, we identify three fundamental types of distances, namely action-history, action-action and history-history distances. We implement a scalable approximation of the full model using variational auto-encoders and the local reparametrization trick. We show using extensive simulation studies that our method out-performs or matches the value of both state-of-the-art organic-based recommendation algorithms, and of bandit-based methods (both value and policy-based) both in organic and bandit-rich environments.

Contents

6.1 Introduction

The recommender systems literature is somewhat bifurcated into two distinct branches. One branch concerns analysing logs of organic user sessions where similar items co-occur (Adomavicius and Tuzhilin, 2005; Koren and Bell, 2015; Hidasi and Karatzoglou, 2018; Liang et al., 2018). A distinguishing feature of this research is that it focuses on logs of organic user sessions, where users view variable numbers of (usually) related items in a shopping session.

A second branch of research explicitly (and entirely) focuses on the logs of the recommender system, using the history of successful and unsuccessful recommendations in order to discover a good recommender system policy. This branch uses off policy learning in order to discover new policies with good actions (Beygelzimer and Langford, 2009; Bottou et al., 2013; Swaminathan and Joachims, 2015a). This work is distinguished by its use of recommender system logs for training and its anonymous feature vector (usually called the context).

The purpose of this work is twofold. Firstly, we pose a simple yet powerful model, that combines these two distinct data sources in order to efficiently learn good recommendation policies. Secondly, we develop a fully probabilistic approach to recommendation and outline its benefits and consequences. The probabilistic formulation gives insights into user embedding creation and the alternative frameworks of value (direct method) and policy learning (importance weighting methods).

The remainder of the chapter is structured as follows: In Section 2 we introduce our probabilistic model of organic and bandit behaviour and discuss its properties. In Section 3 we describe the training of the model. In section 4 we apply our model to the RecoGym simulator (Rohde et al., 2018) and present results. Concluding remarks are made in Section 5.

6.2 Probabilistic Model of Organic and Bandit Sessions

We develop a simple probabilistic model that allows us to build a representation of a user from a variable length organic sequence of items, and then predict accurately how probable the user is to respond positively to each recommendation in the catalogue.

Throughout this chapter, we will make use of the notation introduced in Table 6.1. We use u to denote a user or a session, we use t time to denote sequential time and v to denote which product they viewed from 1 to P where P is the number of products ($P = |\mathcal{A}|$). User u will also be given some recommendations (or actions) $a_{u,1}, ..., a_{u,n}$ again which can take values from 1 to P and we will observe a reward (or a click) for each of these recommendations $c_{u,1}, ..., c_{u,n}$. The organic part of the session are the items the user views without any encouragement from the recommender system i.e. $v_{u,1}, ...v_{u,T_u}$, the bandit part of the session refers to the recommender system log: $a_{u,1}, ..., a_{u,n_u}; c_{u,1}, ..., c_{u,n_u}$. Thus, the size of the organic dataset is U, the number of users, and the bandit dataset size is $\sum_u n_u = N$. We drop the u subscript and treat the bandit dataset as records with $n \in [1, ..., N]$.

In our model, the user's interest is described by a K dimensional variable $\boldsymbol{\omega}_u$, which can be interpreted as the user's interest in K topics. We then assume the following generative process for the organic views in each session:

$$\boldsymbol{\omega}_u \sim \mathcal{N}(\mathbf{0}_K, \boldsymbol{I}_K), \quad v_{u,1}, .., v_{u,T_u} \sim \text{categorical}(\text{softmax}(\boldsymbol{\Psi}\boldsymbol{\omega}_u + \boldsymbol{\rho}))$$

Symbol	Dimension	Description
u	Scalar	A given user's id.
t	Scalar	sequential time.
P	Scalar	Total number of products.
K	Scalar	The size of the embedding.
$v_{u,t}$	Scalar	Product id for user u at time t.
$\boldsymbol{\omega}_u$	$K \times 1$	A given user's state.
$\boldsymbol{\Psi}$	$P \times K$	Organic embedding matrix.
$\boldsymbol{\Psi}_v$	$1 \times K$	Organic embedding for v.
$\boldsymbol{\beta}$	$P \times K$	Bandit embedding matrix.
$\boldsymbol{\beta}_v$	$1 \times K$	Bandit embedding for v.
$\boldsymbol{\rho}$	$P \times 1$	Item popularity intercept.
$\boldsymbol{\kappa}$	$P \times 1$	Item recommendability intercept.
T_u	Scalar	Session length for u.
N	Scalar	The size of the Bandit dataset.
U	Scalar	The number of user sessions.

Table 6.1: Notations and Definitions

The organic embedding matrix $\boldsymbol{\Psi}$ is $P \times K$ and represents information about how items correlate in a user's session organically (i.e. without any intervention from the recommender system). The P dimensional vector $\boldsymbol{\rho}$ is related to the organic popularity of each of the items. Once this session is generated, a recommendation or actions is made to user u denoted a_u and a reward or click will be observed c_u.

$$c_u | a_u, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\kappa} \sim \text{Bernoulli}\{\text{sigmoid}(\boldsymbol{\beta}_{a_u} \boldsymbol{\omega}_u + \boldsymbol{\kappa}_{a_u})\}$$

The bandit embedding matrix $\boldsymbol{\beta}$ is $P \times K$ and represents information about how to personalise recommendations to a user u with a latent user representation $\boldsymbol{\omega}_u$. The organic behaviour is parameterized by $\boldsymbol{\Psi}, \boldsymbol{\rho}$ and the bandit behaviour is parameterized $\boldsymbol{\beta}, \boldsymbol{\kappa}$ in order to relate the two we use the following matrix variate prior distribution of $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} | \boldsymbol{\Psi} \sim \mathcal{MN}(s^+(w_a)\boldsymbol{\Psi}, s^+(w_b)\boldsymbol{\Psi}\boldsymbol{\Psi}^T, s^+(w_b)\frac{1}{P}\boldsymbol{\Psi}^T\boldsymbol{\Psi}).$$

Where $\mathcal{MN}(\cdot)$ is the matrix variate normal distribution[1]. We will show how each of the three terms in the matrix variate normal allow us to include in our model one of the three fundamental differences of recommendation. The softplus function is defined:

$$s^+(w) = \log\{1 + \exp(w)\}.$$

We also put a prior on $\boldsymbol{\kappa}$, which is $P \times 1$:

$$\boldsymbol{\kappa} \sim \mathcal{N}(w_c, \boldsymbol{I}_P \sigma_\kappa^2).$$

The hyperparameters w_a, w_b, w_c are also given normal priors:

$$w_a \sim \mathcal{N}(\mu_{w_{a_0}}, \sigma^2_{w_{a_0}}), \;\; w_b \sim \mathcal{N}(\mu_{w_{b_0}}, \sigma^2_{w_{b_0}}), \;\; w_c \sim \mathcal{N}(\mu_{w_{c_0}}, \sigma^2_{w_{c_0}}).$$

[1] The matrix normal distribution can be defined by its connection to the multivariate normal. If $\boldsymbol{\beta} \sim \mathcal{MN}(\boldsymbol{M}, \boldsymbol{R}, \boldsymbol{S})$, where mean matrix $\boldsymbol{M}$ is $M \times N$, and $\boldsymbol{R}$ is $M \times M$ and $\boldsymbol{S}$ is $N \times N$ - then: $\text{vec}(\boldsymbol{\beta}) \sim \mathcal{N}(\text{vec}(\boldsymbol{M}), \boldsymbol{R} \otimes \boldsymbol{S})$. In this way the matrix variate normal has a more compact and restricted representation of the co-variance than the matrix variate normal. Here ⊗ denotes the Kronecker product.

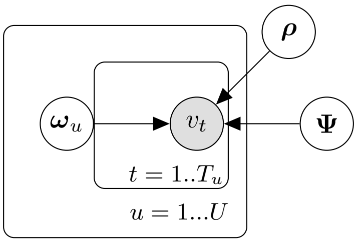


Figure 6.1: A graphical model of the organic behaviour.

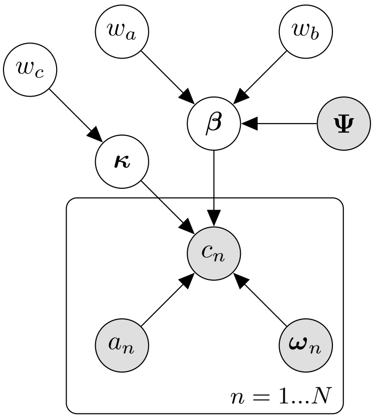


Figure 6.2: A graphical model of the bandit behaviour.

In this chapter, we will mostly consider the organic and bandit behaviour as separate but related processes. A graphical model defining the organic portion of the model is given in Figure 6.1. This graphical model has a similar structure to the latent Dirichlet Allocation model (LDA, (Blei et al., 2003)) , the difference being that where we model $v \sim \text{categorical}\{\text{softmax}(\boldsymbol{\Psi}\boldsymbol{\omega}+\boldsymbol{\rho})\}$, LDA uses $v \sim \text{categorical}(\boldsymbol{\Psi}\boldsymbol{\omega})$ putting simplex constraints on $\boldsymbol{\Psi}$ and $\boldsymbol{\omega}$, similarly correlated topic models (Lafferty and Blei, 2006) use $v \sim \text{categorical}\{\boldsymbol{\Psi}\text{softmax}(\boldsymbol{\omega})\}$ where the simplex constraint is only on $\boldsymbol{\Psi}$. This model can also be viewed as a linear version of the Multi-VAE (Liang et al., 2018). We will show that using variational autoencoders with the re-parameterization trick is an effective way to train the organic model.

The approach developed in this chapter takes the organic model and estimates $\boldsymbol{\Psi}$ by maximum likelihood and $\boldsymbol{\omega}$ by the posterior mean (denoted $\hat{\boldsymbol{\omega}}$) and then treats $\boldsymbol{\Psi}$ and $\hat{\omega}$ as observed in the bandit model. The graphical model is shown in Figure 6.2. In this probabilistic model we will develop full Bayesian inference of the $\boldsymbol{\beta}$, $\boldsymbol{\kappa}$, w_a, w_b and w_c this is important because the bandit signal is very uneven. Lots of information is available on past actions that the previous recommender system favoured and little information or no information is available on many other actions, this means the posterior is tight in some regions but broad and highly influenced by the prior in others. We use variational approximations and the local re-parameterization trick in order to capture this complex structure.

We refer to the organic only component of the model as **BLO** (Bayesian Latent Organic) model (we apply maximum likelihood to $\boldsymbol{\Psi}, \boldsymbol{\rho}$ and integrate $\boldsymbol{\omega}$). The full model is referred to as **BLOB** (Bayesian Latent Organic Bandit Model).

6.2.1 Intuition for the model

The model presented embodies a fundamental implicit assumption in the traditional recommendation system, the assumption that auto-completion of a session results in good recommendations being made. This is one of the three fundamental distances of recommendation, the action-history distance.

The implicit assumption in traditional recommendation: good recommendations are (usually) similar to the items in the user's history

Algorithms in the recommendation literature look at items in a user's history and attempt to predict the final element in this session. The fraction of times that the predicted item is within the top K items in a held out data set is a key metric called precision@K that measures a models' ability to "auto-complete" a users' behaviour. The organic performance is therefore computed:

$$P(v_{u,T_u}|v_{u,1},..,v_{u,T_u-1}).$$

Metrics such as NDCG, recall@K or log likelihood are computed on this auto-completion task.

However auto-completion is not the same as recommendation. In fact, to reduce recommendation to auto-completion removes the opportunity for a recommender system to help a user discover new things which arguably is the primary objective of recommendation. That said, organic data is usually plentiful and this implicit assumption that recommendation as auto-completion certainly has some merit. We can state this assumption as, if:

$$P(V_{u,T_u}=v_a|v_{u,1},..,v_{u,T_u-1}) > P(V_{u,T_u}=v_b|v_{u,1},..,v_{u,T_u-1})$$

Then item v_a is probably better than item v_b as a recommendation, i.e. the following holds often:

$$P(c=1|A=v_a,v_{u,1},..,v_{u,T_u-1}) > P(c=1|A=v_b,v_{u,1},..,v_{u,T_u-1}).$$

Although this relationship often holds, it need not hold in every single instance. Maybe the user already knows about item v_a, maybe the recommendation for v_a is unattractive or maybe the reason the user never visited item v_b is lack of knowledge, and it is actually a very valuable recommendation. We want our recommender system to make use of the organic relationship, but we also want to learn from the logs of the recommender system itself which records if the recommendations that we chose to deliver were successful or not. This "bandit feedback" is in some sense the true arbiter of if a recommendation is good or not, but the bandit signal is usually highly concentrated around what the previous version of the recommendation system judged to be a good recommendation, so it cannot reliably be used over the entire recommendation space. For example, the organic session might contain information that two products (say) rice and a phone are rarely viewed together in the same organic session. However, it probably will not contain many events where a phone is recommended to a user with rice in their history. If the recommender system is to infer that this is likely a poor recommendation, it must do so through a prior linking the bandit behaviour to the organic behaviour.

When deployed in a production recommender system, the model operates in the following way. First, a posterior over a user embedding is approximately calculated:

$$P(\boldsymbol{\omega}_u|v_{u,1},...,v_{u,T_u},\boldsymbol{\Psi},\boldsymbol{\rho})$$

A fast variational approximation can be made of $\boldsymbol{\omega}_u \sim \mathcal{N}(\boldsymbol{\mu}_{\omega_q},\boldsymbol{\Sigma}_{\omega_q})$ which gives both a mean and a variance (this can be done using either a variational EM algorithm or a VAE).

For our purposes we make the pragmatic compromise that we can summarise the user history with a posterior mean point estimate $\hat{\boldsymbol{\omega}} = \boldsymbol{\mu}_{\omega_q}$, this prevents numerical integration of $\boldsymbol{\omega}_u$ at recommendation time. Once this compromise is made, it also makes sense to train the organic and bandit components separately. The probability of a click is given by:

$$P(c|\hat{\boldsymbol{\omega}}, \boldsymbol{\beta}, \boldsymbol{\kappa}, a) = \text{sigmoid}(\boldsymbol{\beta}_a \hat{\boldsymbol{\omega}} + \boldsymbol{\kappa}_a)$$

The recommender system will then choose a recommendation that will optimise this reward (or a combination of reward and exploration - but the explore-exploit dilemma (Lattimore and Szepesvári, 2020) is beyond the scope of this chapter).

The organic parameters $\boldsymbol{\Psi}$ and $\boldsymbol{\rho}$ are not required in order to deliver a recommendation. They are used only to put a prior on the bandit embeddings. We note parenthetically that due to the fact that once the user embedding $\hat{\boldsymbol{\omega}}$ is created, the model is linear, and we can exploit fast algorithms to quickly find the optimal recommendation over large catalogues (Gionis et al., 1999; Malkov and Yashunin, 2018).

The organic user session

The organic user session model we propose can be understood in a number of ways. It can be viewed as a user item matrix factorization where the user has a latent interest in K topics - a discussion of this interpretation is given in the supplementary material.

It can also be viewed as an i.i.d. categorical process with a (usually) low-rank multivariate normal prior. The prior causes similar items to co-occur in a session with high probability. Because of this assumption, seeing an item will always make it more likely to be viewed again. If we had a full rank model, the user session would imply the law of large numbers where the next item prediction will converge to the empirical frequency. In practice, the session history is short, and the embedding size is much lower than the number of products, but the assumption remains that viewing an item makes the conditional probability for that same item increase (also, the conditional probability that similar items will be viewed also increases).

This is a relatively strong assumption compared to powerful sequential models such as recurrent neural networks (Hidasi and Karatzoglou, 2018), which can model complex sequences. Recurrent neural networks excel at text processing tasks where there is need to do things like close brackets in text. The simpler and stronger assumption made by BLO is reasonable in many settings and greatly simplifies learning.

The bandit session and the three distances in recommendation

The auto-complete assumptions as embodied in the recommendation research measures the similarity between the recommendation and the items in history. This is the first similarity or distance, the distance between the history and the action. The mean of the matrix normal $\boldsymbol{\Psi}$ embodies this assumption.

The second similarity in recommendation is the similarity between actions. That is if action a_1 and a_2 are similar then we expect that the responses to these actions to the same (or similar) users be correlated. This distance is encoded with the first (low rank) co-variance $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$ in the matrix normal prior on $\boldsymbol{\beta}$.

The third similarity in recommendation is the similarity between users. If user u_1 and u_2 are similar, then we expect the response to the same (or similar) action on these users to be correlated. This distance is encoded with the second co-variance $\mathbf{\Psi}^T\mathbf{\Psi}$ in the matrix normal prior on $\boldsymbol{\beta}$.

The effect of the first distance is to seed the recommendation using the organic similarities, the effect of the second and third is to borrow strength, allowing the bandit signal to be used more effectively. Finally, the parameters w_a and w_b control the strength of the influence of the first and second distance. The relative strength of the first distance and the second is an extremely important hyperparameter.

6.2.2 Value vs policy learning

The method proposed here is a value based (direct) method, as it learns the value for every action and then can determine a decision rule using unconstrained optimisation. In this way it differs from alternative methods for learning from bandit feedback that have been recently proposed Beygelzimer and Langford (2009); Bottou et al. (2013); Swaminathan and Joachims (2015a) which learns a policy directly using importance weighting objectives.

Bayesian methods are inherently value based and bring the benefit of being able to synthesis data sources such as organic and bandit, they also produce uncertainty that is useful for explore-exploit strategies such as upper confidence bound and Thompson sampling (Lattimore and Szepesvári, 2018). From a purely statistical point of view, principles such as the conditionality and the likelihood principle actually forbid the use of the propensity score (Berger et al., 1988; Hernan and Robins, 2010). Given that training on bandit feedback is sometimes considered to be synonymous with using the inverse propensity score (IPS), it is worth reviewing some advantages of Bayesian value based methods.

It has been shown in Ritov et al. (2014), that under regularity conditions that apply in the recommendation case, the Bernstein-von Mises theorem applies, and that the Bayesian estimator is efficient, $\sqrt{n}$ consistent and necessarily better than the IPS (or Horvitz-Thompson) estimator[2]. However, note that a real recommender system log will be of sufficient dimensionality that even with terabytes of logs, asymptotic theory is usually not relevant (i.e. priors will have real impacts).

It is also sometimes argued that the IPS score is necessary to apply in counterfactual settings due to the domain shift which occurs in causal settings (Johansson et al., 2016). However, this argument does not apply when the model has enough capacity to accurately predict the value everywhere (Storkey, 2009) and there is no need to constrain capacity to reduce estimator variance when applying Bayesian methods (Neal, 2012). It seems that some of the positive aspects of value based methods have been overlooked due to criticisms that apply only in the non-Bayesian case.

Policy learning also suffers from some drawbacks. Policy learning extends the principle of Statistical Learning Theory (SLT) to the counterfactual setting. The idea of SLT is that a decision rule is fit to the historical data from a constrained set. If a decision rule from a restricted set has good performance (low risk) then it is likely to also have low risk on out

[2] They additionally show that IPS based methods can have better frequentist properties than Bayesian estimators when these regulatory conditions break down.

of sample data (Vapnik, 2013). These analyses are based upon treating empirical risk or counterfactual risk as a statistic, but these are highly non-sufficient statistics and there is no ability to order decision rules that have the same empirical risk even when away from the data they are very different. The theory is heavily based on having a restricted set of decision rules, but restricting the set might exclude good decisions. Value based methods make no such restriction.

Extending SLT to the counterfactual setting requires some additional ideas because the consequences of decisions the new policy will make are not available. IPS-based methods have been a recent research focus that extend the empirical risk minimisation to the counterfactual setting. Technical challenges are being addressed, such as the fact that the variance of the decision rule can vary depending on how much it differs from the historical logging policy Swaminathan and Joachims (2015a). As well as the problem of propensity overfitting i.e. decision rules can achieve an estimated reward of 0 by avoiding past decisions (0 might be good or bad depending on how the reward is defined) causing decision rules either to cling to the old policy or to be driven away from it[3]. It is usually considered a better heuristic for the new policy to cling to the old one.

One simple method to control variance is to cap large weights (Bottou et al., 2013) (necessarily associated with actions that are different to the logging policy). This method controls the bias-variance trade-off. Another method that more explicitly discourages deviation from the logging policy is to apply variance penalization (Swaminathan and Joachims, 2015a) here rather than optimizing the counterfactual risk directly a penalized term is instead optimized, this penalization naturally goes up if the recommendations are rare under the logging policy (and hence have a high IPS weight).

Many of the standard policy learning settings [4] have the property that the learnt policy will only deviate from the preferred decision of the logging policy in the face of considerable evidence. This is a good heuristic in cases where the logging policy is good, but can be a problem in other situations. The potential strength of policy based approaches is due to the fact they do not use a model, and they focus directly on the decision rule, focusing optimisation and capacity on the parts of the problem that matters most. Bayesian value based methods cannot do this because the modelling step is made before and separately to the decision-making step.

6.3 Model Training

6.3.1 Organic session training: learning the organic embeddings

The log likelihood of the organic model has the form:

$$\log p(v_1, .., v_T, \boldsymbol{\omega}_u | \boldsymbol{\Psi}) = \left(\sum_t^T \boldsymbol{\Psi}_{v_t} \boldsymbol{\omega}_u + \boldsymbol{\rho}_{v_t} \right) - T \log \left\{ \sum_p^P \exp(\boldsymbol{\Psi}_p \boldsymbol{\omega}_u + \boldsymbol{\rho}_p) \right\} + \log p(\boldsymbol{\omega}_u)$$

As the posterior on ω is intractable, we use a normal distribution $\boldsymbol{\omega}_u \sim \mathcal{N}(\boldsymbol{\mu}_{q_\omega}, \boldsymbol{\Sigma}_{q_\omega})$ to approximate it, we get a variational lower bound (ELBO) of the form:

[3] The self normalized importance sampling variant of IPS is one proposal to remove this sensitivity to the definition of the reward (Schnabel et al., 2016)

[4] This includes having reward positive and no-reward zero, capping and variance penalization

$$\begin{aligned}\mathcal{L} &= \mathop{\mathbb{E}}_{q(\boldsymbol{\omega}_u)}\left[\log p(v_1, .., v_T, \boldsymbol{\omega}_u | \boldsymbol{\Psi}) - \log q(\boldsymbol{\omega}_u)\right] \\ &= \left(\sum_t^T \boldsymbol{\Psi}_{v_t}\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{v_t}\right) - T \mathop{\mathbb{E}}_{q(\boldsymbol{\omega}_u)}\left[\log\left\{\sum_p^P \exp(\boldsymbol{\Psi}_p\boldsymbol{\omega}_u + \boldsymbol{\rho}_p)\right\}\right] - \mathcal{KL}(\mathrm{q}(\boldsymbol{\omega}_\mathrm{u})||\mathrm{p}(\boldsymbol{\omega}_\mathrm{u})).\end{aligned}$$

Where $\mathcal{KL}$ is a closed form KL divergence between the variational posterior and the prior (a multivariate standard normal distribution). We see that there is a problematic term associated with the denominator of the softmax. We use the re-parameterization trick (Kingma and Welling, 2014) to approximate the gradient of this term. It is also possible to use the Bouchard bound (which also enables an EM algorithm) and the log concave bound, both bounds can alleviate computational issues associated with the softmax sum (Bouchard, 2007), details of these lower bounds and the EM and simulated EM algorithm are given in the supplementary material.

Re-parameterization Trick

An effective approach to computing expectations with respect to the denominator of the softmax is to use the re-parameterization trick (Kingma and Welling, 2014), which allows us to take a sample of $\boldsymbol{\omega}$ from the variational distribution and compute a noisy derivative of the lower bound. Within each iteration, we proceed by simulating:

$$\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}(\mathbf{0}_K, \boldsymbol{I}_K),$$

and then computing:

$$\boldsymbol{\omega}^{(s)} = L_{\boldsymbol{\Sigma}_{q_\omega}}\boldsymbol{\epsilon}^{(s)} + \boldsymbol{\mu}_{q_\omega}.$$

Where $\boldsymbol{L}_{\boldsymbol{\Sigma}_{q_\omega}}\boldsymbol{L}^T_{\boldsymbol{\Sigma}_{q_\omega}} = \boldsymbol{\Sigma}_{q_\omega}$, we can then optimize an approximation of the lower bound:

$$\begin{aligned}\mathcal{L}_{MC} = &\left(\sum_t^T \boldsymbol{\Psi}_{v_t}\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{v_t}\right) - \mathcal{KL}(q(\boldsymbol{\omega}_u)|p(\boldsymbol{\omega}_u)) \\ &- T\log\left\{\sum_p^P \exp\left(\boldsymbol{\Psi}_p(L_{\boldsymbol{\Sigma}_{q_\omega}}\boldsymbol{\epsilon}^{(s)} + \boldsymbol{\mu}_{q_\omega}) + \boldsymbol{\rho}_p\right)\right\}\end{aligned}$$

Often $\boldsymbol{\Sigma}_{q_\omega}$ is taken to be diagonal, which makes computing $\boldsymbol{L}_{\boldsymbol{\Sigma}_{q_\omega}}$ simply an element-wise square root. A naive application of the algorithm discussed so far would have the number of variational parameters $\boldsymbol{\mu}_{q_\omega}, \boldsymbol{\Sigma}_{q_\omega}$ growing with the number of user sessions. We propose instead to limit the number of parameters by the use of a variational auto-encoder (Kingma and Welling, 2014). This involves using a flexible function and optimizing it to do the job of the EM algorithm, i.e.

$$\boldsymbol{\mu}_{q_\omega},\ \boldsymbol{\Sigma}_{q_\omega} = f_\Xi(v_1, ...v_T),$$

Where any function (e.g. a deep net) can be used for $f_\Xi(\cdot)$.

6.3.2 Bandit session training: learning the bandit embeddings

For every user we compute: $\hat{\boldsymbol{\omega}}_u = f(\boldsymbol{v}_u)$ (uncertainty over $\boldsymbol{\omega}_u$ is ignored and a point estimate taken). The hierarchical model has the form:

$$w_a \sim \mathcal{N}(\mu_{0_{w_a}}, \sigma^2_{0_{w_a}}), \quad w_b \sim \mathcal{N}(\mu_{0_{w_b}}, \sigma^2_{0_{w_b}}), \quad w_c \sim \mathcal{N}(\mu_{0_{w_c}}, \sigma^2_{0_{w_c}})$$

$$\boldsymbol{\kappa}' \sim \mathcal{N}(\mathbf{0}_P, \sigma^2_{\kappa 0}\boldsymbol{I}_P), \qquad \boldsymbol{\kappa} = \boldsymbol{\kappa}' + w_c$$

$$\beta|\boldsymbol{\Psi}, w_a, w_b \sim \mathcal{MN}(s^+(w_a)\boldsymbol{\Psi}, s^+(w_b)\boldsymbol{\Psi}\boldsymbol{\Psi}^T, s^+(w_b)\frac{1}{P}\boldsymbol{\Psi}^T\boldsymbol{\Psi})$$

$$c_n|a_n, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\kappa} \sim \text{Bernoulli}\{\text{sigmoid}(\boldsymbol{\beta}_{a_n}\boldsymbol{\omega}_n + \boldsymbol{\kappa}_{a_n})\}.$$

While β is a $[P \times K]$ random variable, we can leverage its low rank covariance matrix to transform the problem to inferring a posterior on a $[K \times K]$ random variable. This reduces dramatically the training time as P, the size of the catalogue items is usually very large compared with K. The low rank alternative parameterization of this distribution can be set as follows. Let:

$$\boldsymbol{\zeta} \sim \mathcal{MN}(\mathbf{0}_{K,K}, \boldsymbol{I}_K, \boldsymbol{I}_K).$$

If we let $\boldsymbol{L}$ be the result of a Cholesky decomposition of $\frac{1}{P}\boldsymbol{\Psi}^T\boldsymbol{\Psi}$, i.e. $\boldsymbol{L}\boldsymbol{L}^T = \frac{1}{P}\boldsymbol{\Psi}^T\boldsymbol{\Psi}$. A valid way to sample from a matrix variate normal gives:

$$\boldsymbol{\beta} = s^+(w_a)\boldsymbol{\Psi} + s^+(w_b)\boldsymbol{\Psi}\boldsymbol{\zeta}\boldsymbol{L}^T$$

As mentioned before, we treat the problem in a Bayesian way and approximate the posterior over all the parameters. We use variational inference to transform the problem into an optimization problem. We use a univariate normal variational approximation on w_a, w_b, w_c with means $\mu_{q_{w_a}}, \mu_{q_{w_b}}, \mu_{q_{w_c}}$ and variance $\sigma^2_{q_{w_a}}, \sigma^2_{q_{w_b}}, \sigma^2_{q_{w_c}}$. The variational approximation on κ is a diagonal covariance multivariate normal, with mean given by $\boldsymbol{\mu}_{q_\kappa}$ and covariance given by $\text{diag}(\boldsymbol{\sigma}^2_{q_\kappa})$. Similarly, we put a univariate normal variational approximation over each element of $\boldsymbol{\zeta}$ parameterized so that $\boldsymbol{\zeta}_{i,j}$ has mean $\mu_{q_{\zeta_{i,j}}}$ and variance $\sigma^2_{q_{\zeta_{i,j}}}$. This gives us $2(P + K^2 + 3)$ parameters to estimate. We denote Q as the Gaussian variational posterior over all the parameters, and P the prior and maximize :

$$\mathcal{L} = \mathop{\mathbb{E}}_{Q}\left[c_n \log\left\{\texttt{sigmoid}(\lambda_n)\right\} + (1 - c_n)\log\left\{1 - \texttt{sigmoid}(\lambda_n)\right\}\right] - \frac{1}{N}\mathcal{KL}(Q||P), \tag{6.1}$$

where:

$$\begin{aligned}\lambda_n &= \boldsymbol{\beta}_{a_n}\hat{\boldsymbol{\omega}}_n + \boldsymbol{\kappa}_{a_n} \\ &= s^+(w_a)\boldsymbol{\Psi}_{a_n}\hat{\boldsymbol{\omega}}_n + s^+(w_b)\{(\boldsymbol{L}\hat{\boldsymbol{\omega}}_n)^T \otimes \boldsymbol{\Psi}_{a_n}\}\text{vec}(\boldsymbol{\zeta}) + \boldsymbol{\kappa}_{a_n}.\end{aligned}$$

We use the local re-parameterization trick (Kingma et al., 2015) which uses the Affine transform properties of the multivariate Gaussian distribution to allow the re-parameterization trick to be employed on lower dimensions. This results in sampling at lower dimensions and more importantly makes the derivatives of the loss less noisy. To implement the local re-parameterization trick, we draw random samples:

$$\boldsymbol{\epsilon}_{w_a} \sim \mathcal{N}(0,1), \quad \boldsymbol{\epsilon}_{w_b} \sim \mathcal{N}(0,1), \quad \boldsymbol{\epsilon}_{\text{lrt}} \sim \mathcal{N}(0,1), \quad \boldsymbol{\epsilon}_{\kappa} \sim \mathcal{N}(0,1).$$

with $\boldsymbol{R}_n = (\boldsymbol{L}\hat{\boldsymbol{\omega}}_n)^T \otimes \boldsymbol{\Psi}_{a_n}$, we can get a one dimensional noisy estimate of λ_n :

$$\begin{aligned}\hat{\lambda}_n =& s^+(\mu_{q_{w_a}} + \epsilon_{w_a}\sigma_{q_{w_a}})\boldsymbol{\Psi}_{a_n}\hat{\boldsymbol{\omega}}_n \\ &+ s^+(\mu_{q_{w_b}} + \epsilon_{w_b}\sigma_{q_{w_b}})(\boldsymbol{R}_n \ \text{vec}(\boldsymbol{\mu}_{q_\zeta}) + \|\boldsymbol{R}_n^T \odot \text{vec}(\boldsymbol{\sigma}_{q_\zeta})\|_2 \boldsymbol{\epsilon}_{\text{lrt}}) \\ &+ \boldsymbol{\mu}_{q_{\kappa_a}} + \mu_{q_{w_c}} + \epsilon_\kappa\sqrt{\boldsymbol{\sigma}^2_{q_{\kappa_a}} + \sigma^2_{q_{w_c}}}.\end{aligned}$$

where $\|\cdot\|_2$ denotes the L_2 norm and $\odot$ element wise multiplication. We can optimize a noisy version of our objective :

$$\hat{\mathcal{L}}_n = c_n \log\left\{\texttt{sigmoid}(\hat{\lambda}_n)\right\} + (1-c_n)\log\left\{1-\texttt{sigmoid}(\hat{\lambda}_n)\right\} - \frac{1}{N}\mathcal{KL}(Q||P) \tag{6.2}$$

We call the solution of this optimization problem **BLOB-NQ** as we considered a Normal approximation for the posterior on ζ.

An alternative approach is to use a matrix variate normal distribution as the variational approximation of $\boldsymbol{\zeta}$ with mean matrix $\boldsymbol{\mu}_{q_\zeta}$ and the two covariance matrices given by: $\text{diag}(\boldsymbol{\sigma}^2_{q_{\zeta_1}})$ and $\text{diag}(\boldsymbol{\sigma}^2_{q_{\zeta_2}})$. This reduces the number of variational parameters used for representing the variance of the variational posterior. We thus need to estimate $2(P+3)+K^2+2K$ parameters, which is less than the previous approximation for $K \geq 2$. To apply the local re-parameterization trick, let:

$$\begin{aligned} std_n =& \sqrt{(\boldsymbol{\sigma}^2_{q_{\zeta_1}} \cdot \boldsymbol{\Psi}^2_{a_n})(\boldsymbol{\sigma}^2_{q_{\zeta_2}} \cdot (\boldsymbol{L}^T\hat{\boldsymbol{\omega}}_n)^2)} \\ \hat{\phi}_n =& s^+(\mu_{q_{w_a}} + \epsilon_{w_a}\sigma_{q_{w_a}})\boldsymbol{\Psi}_{a_n}\hat{\boldsymbol{\omega}}_n \\ &+ s^+(\mu_{q_{w_b}} + \epsilon_{w_b}\sigma_{q_{w_b}})\{\boldsymbol{\Psi}_{a_n}\boldsymbol{\mu}_{q_\zeta}\boldsymbol{L}^T\hat{\boldsymbol{\omega}}_n + std_n\boldsymbol{\epsilon}_{\text{lrt}}\} \\ &+ \boldsymbol{\mu}_{q_{\kappa_a}} + \mu_{q_{w_c}} + \epsilon_\kappa\sqrt{\boldsymbol{\sigma}^2_{q_{\kappa_a}} + \sigma^2_{q_{w_c}}}. \end{aligned}$$

A noisy estimate of the lower bound can then be computed by substituting $\hat{\phi}_n$ into Equation (6.2). We call its solution of (6.2) with $\hat{\phi}_n$ substituted in **BLOB-MNQ** as we use a Matrix Normal variational posterior.

In both approximations and when the objective is at its maximum, we can take a point estimate of the bandit embeddings:

$$\hat{\boldsymbol{\beta}} = s^+(\mu_{q_{w_a}})\boldsymbol{\Psi} + s^+(\mu_{q_{w_b}})\boldsymbol{\Psi}\boldsymbol{\mu}_{q_\zeta}\boldsymbol{L}^T.$$

The bandit embedding can be interpreted as a weighted sum of the organic embedding and the organic embedding multiplied by a $K \times K$ matrix that can adjust the bandit embeddings based on the bandit signal.

6.4 Results

6.4.1 Organic Evaluation

We demonstrate that our method produces useful user representations on next item prediction using the RecoGym simulation environment (Rohde et al., 2018). RecoGym is a framework for simulating a recommender system and enables the simulation of A/B tests, although here we simply use it to create organic sequences of item views and test the organic model's ability to do next item prediction. We split both the datasets into train and test so that sessions reside entirely in one of the two groups. We fit the model to the training set, we then evaluate by providing the model $v_1, ..v_{T_u-1}$ events and testing the model's ability to predict v_{T_u}.

The organic model was implemented using the PyTorch automatic differentiation package in Python (Paszke et al., 2017) and trained using Stochastic Gradient Descent (SGD), specifically the RMSProp variant. We set the learning rate to 10^{-3} and tune the other hyperparameters, including L2 regularization, for each dataset based upon a validation set[5].

The various models are evaluated using recall at K (`RC@K`) and truncated discounted cumulative gain at K (`DCG@K`), which are defined below. Let r_k be the kth highest value of $p(\boldsymbol{\omega}_{v_{T_u}}|v_1, ..v_{T_u - 1})$. For all results presented in this chapter, we set K to 5.

$$\texttt{RC@K} = \begin{cases} 1, & \text{if } v_{T_u} \in \{r_1, ..., r_K\}. \\ 0, & \text{otherwise.} \end{cases}$$

$$\texttt{DCG@K} = \sum_i \frac{2^{r_i \mathbf{1}\{v_{T_u} \in \{r_1,...,r_K\}\}} - 1}{\log i + 1}.$$

We compute the average of these quantities over all sessions in the test set. We consider two alternative methods for training the model:

- **Bouch/AE** - A linear variational auto-encoder using the Bouchard bound (see the supplementary material).
- **RT/AE** - A deep auto-encoder again using the re-parameterization trick. The deep auto-encoder consists of mapping an input of size P to three linear rectifier layers of K units each.

When we update the posterior over a user's latent variable representation at test time, we assess both using the auto-encoder denoted AE and using the 100 iterations of the EM algorithm denoted EM in the results. When we compute next item predictions we consider both using a 100 sample Monte Carlo approximation denoted MC and just taking the mean as a point estimate denoted mean. To demonstrate the effectiveness of our approach, we present results from the following baseline approaches:

- **Popularity**: Item popularity provides no personalization, but is nonetheless a strong baseline for certain recommendation tasks.
- **Item KNN**: Item K Nearest Neighbors (KNN) involves computing the correlation matrix of the sample data adding the identity to prevent division by zero and then using these correlations as recommendations based on a user's most recent historical item. The limitations of this technique is that it ignores item popularity and multiple items in the user's history, but despite these limitations it is often a strong baseline.
- **Recurrent Neural Network**: For this baseline, we make use of a recurrent neural network to learn a user representation by predicting the next item in the session. The model architecture we employ is similar to that of Hidasi and Karatzoglou (2018), in that we feed the output from an embedding layer into a Gated Recurrent Unit (GRU) (Cho et al., 2014) with 64 hidden units to learn the temporal dynamics of the user's session. The output from the GRU is then passed through a final softmax layer, which gives the probability of the next item in the sequence. The network is trained to minimize the categorical cross-entropy over the training sessions via RMSProp.

For our organic experiment we use the RecoGym simulator with 2000 products and $\sigma_\omega = 0$, i.e. a static user state, we generate a training set of 100 sessions and a test set of 100 sessions,

[5] All code will be released upon acceptance. The RecoGym simulator allows reproducible results for all recommendation algorithms and policies.

Algorithm	Latent	Next Item	RC@5	DCG@5
Pop			0.020	0.016
ItemKNN			0.020	0.024
RNN			0.035	0.033
Bouch/AE	AE	MC	0.082	0.128
Bouch/AE	AE	mean	0.082	0.079
Bouch/AE	EM	MC	**0.117**	0.128
Bouch/AE	EM	mean	**0.117**	**0.130**
RT/AE	AE	MC	0.090	0.105
RT/AE	AE	mean	0.080	0.068
RT/AE	EM	MC	0.090	0.105
RT/AE	EM	mean	0.090	0.106

Table 6.2: Results on the testset of the RecoGym dataset with 2000 products. For both metrics, a higher value is better.

this results in 21852 and 19533 events for train and test respectively. The BLO models were all trained using 15000 epochs using the RMSProp algorithm, the embedding size was set to 10. The RNN was trained with $K = 200$ for 5000 epochs (it performed slightly worse with a training run of 25000). The results are shown in Table 6.2. BLO is much better than the baselines at standard organic recommender systems metrics. However, if being able to build an adequate model of organic behaviour is sufficient for building a recommender system depends on if the organic behaviour is aligned with bandit behaviour. This requires using RecoGym for its intended purpose, simulating A/B tests and varying the agreement between the organic behaviour and bandit behaviour using the provided flips parameter.

6.4.2 The Complete Model - Organic and Bandit

Experimental Setup

Unfortunately, no real world dataset exhibits the required properties (both organic and bandit behaviour). Moreover, no real world dataset including counterfactual datasets allow us to evaluate the quality of a recommender systems recommendations reliably. For this reason, for the complete dataset, we do our evaluations completely in the RecoGym simulator. A strong advantage of the simulation environment is that not only can we compute offline organic metrics but we can also simulate A/B tests.

Another advantage of the RecoGym simulator that simulates both organic and bandit behaviour is that algorithms from the traditional organic part of recommender systems research and bandit algorithms can be compared side by side. We consider traditional organic algorithms like ItemKNN (Davidson et al., 2010) along side our organic Bayesian Latent Organic model (BLO) and sophisticated deep learning approaches such as the MultiVAE (Liang et al., 2018). In the case of bandit algorithms, we can test value based logistic regression as well as the policy based contextual bandit. In order to apply any bandit algorithm, we need to perform feature engineering in order to transform the history consisting of item views into a vector of history. For the logistic regression, we elect to make a P dimensional feature vector crossed with the action also of size P giving P^2 features. Similarly, the contextual bandit is a linear model that maps the P dimensional vector of historical counts to a P dimensional action space.

We are interested to see how the recommender system responds to different logging policies, we therefore test it using a good logging policy based on the session popularity. That is the probability $1-\epsilon$ is shared proportionally to the items in a users history we use considerable exploration ($\epsilon = 0.3$). We are interested in the (common) case where we have plentiful organic data, so we set RecoGym to have 20000 organic sessions. Finally, we are interested in situations where the next item prediction is an optimal recommendation and cases where the organic signal alone is misleading to recommendation quality. This connection between the organic and the bandit signal is controlled with the *flips* parameter in RecoGym. The *flips* parameter permutes the behaviour of two actions.

A unique feature of RecoGym is that we are able to simulate both organic and bandit feedback, this means we are able to compare algorithms that operate on the bandit signal (both policy and value based) with algorithms that operate on the organic signal. We consider the following baselines:

- **Logistic regression (bandit, value)**: Perhaps the simplest way to process a bandit signal. We regress the reward on features derived from the users history and the recommended action. In order to deliver the recommendation, we predict the reward for every action and select the highest.
- **Contextual bandit (bandit, policy)**: The contextual bandit is a policy based method that maps a context to a recommendation in one-of-n coding a vector of length P. The algorithm is trained using the IPS score logged by RecoGym without any clipping or variance penalty.
- **Session ItemKNN (organic)**: This organic algorithm operates by determining for each session if an item was present or absent, from this dataset a correlation matrix is computed. At recommendation is delivered by computing the average correlations for each item in history as a single vector and then taking the maximum. We take the whole session into account rather than the most recent item (unlike most recent ItemKNN used above).
- **Multi-VAE (organic)**: A state-of-the-art deep learning recommendation algorithm similar to the organic portion of the model presented here, except the model is non-linear and uses some non-standard heuristics such as 'beta-annealing'.
- **BLO (organic)**: The organic portion of the model developed here. We set the embedding size to be K=20 and use a linear variational auto-encoder. This is implemented in PyTorch. A learning rate of 0.0001 is used with 1000 epochs and an embedding size of $K = 20$.
- **BLOB (organic and bandit combined)**: The complete model developed here. We use priors: $w_a \sim \mathcal{N}(-1, 1^2)$, $w_b \sim \mathcal{N}(-6, 1^2)$, $w_c \sim \mathcal{N}(-4.5, 10^2)$, $\kappa \sim \mathcal{N}(w_c, 0.01^2\boldsymbol{I})$. We consider both the normal variational approximation NQ and the matrix normal variational approximation MNQ. The bandit layer is implemented using TensorFlow with a learning rate of 0.001 and 800 epochs for the $P = 100$ and 1200 epochs for the $P = 1000$, with a batch size of 1024 and using the RMSprop training algorithm.
- **Random**: The actions are recommended randomly. A weak baseline, but useful to calibrate performance.

Experimental Results

The first experiment considers the catalogue size to be $P = 100$, the number of user sessions to be 1000, the simulated A/B test is done over 4000 users and the logging policy being session popularity with epsilon greedy exploration (epsilon=0.3). This means that the bandit signal will resemble that found in real systems with a strong signal around some actions favoured by the previous version of the recommender system (session popularity policy - a decent baseline) and a weak signal over much of the remaining action space. Results are shown in Table 6.3.

In the *flips*=0 scenario, RecoGym is configured so that next item prediction based on organic data is a perfect proxy for delivering good recommendations. As a consequence, all the organic based methods do well including the BLO (organic), both our methods that combine organic and bandit **BLOB-NQ** and **BLOB-MNQ** and the Multi-VAE baseline. the Session ItemKNN baseline while organic does not perform well.

When the *flips*=50 scenario, RecoGym internally permutes 50 actions behavior. This means that next item prediction is now a poor proxy of recommendation performance. We see this as all purely organic based agents now perform poorly. Indeed, the connection between organic and bandit is reduced to the point that Session ItemKNN, the Multi-VAE and BLO all perform worse than random. It is in this case that the value of our BLOB model is demonstrated, as both **BLOB-NQ** and **BLOB-MNQ** perform strongly.

For the pure bandit algorithms, the value based Log Reg and the policy based CB perform similarly to each other in both scenarios (*flips*=0 and *flips*=50). They perform a little better than random (except for CB *flips*=50) demonstrating that there is some usable signal in the bandit feedback but are far from state of the art, especially in the *flips*=0 case where ignoring the organic signal profoundly limits recommendation quality. In the *flips*=50 case, the pure bandit approaches outperform the purely organic algorithms but the combined approach performs significantly better, giving a click-through rate of 1.57% for the **BLOB-NQ** compared to 1.21% for the logistic regression.

Importantly, the **BLOB-NQ** and **BLOB-MNQ** are on par or outperform the other methods in the *flips*=0 setting, and significantly outperform the other methods in the *flips*=50 setting.

Table 6.3: Simulated A/B test results on the RecoGym simulator using: $P = 100$, $U = 1000$, organic only sessions=20 000.

Agent	Type	CTR (%)	CTR (%)
		flips=0	*flips*=50
Log Reg	(bandit)	1.37	1.21
CB	(bandit)	1.37	1.09
ItemKNN	(organic)	1.39	0.92
MultiVAE	(organic)	**2.43**	0.76
BLO	(organic)	**2.42**	0.76
BLOB-NQ	(combined)	**2.42**	**1.57**
BLOB-MNQ	(combined)	**2.40**	**1.56**
Random		1.09	1.11

The second experiment considers the same setup but with $P = 1000$, we also increase the number of epochs on the bandit component of the model to 1200. Results are shown in Table 6.3.

Again we see that the methods that use the organic data either the purely organic or the combined BLOB methods we propose perform work well when *flips*=0, but when *flips*=500 the purely organic methods fall in performance to little above random yet the combined methods **BLOB-MNQ** and **BLOB-NQ** continue to perform well beating all other baselines.

The policy based contextual bandit shows a small improvement over the value based logistic regression in the *flips*=0 case, although this advantage vanishes when *flips*=500, this is may be due to the fact that the contextual bandit "clings" to the logging policy and the session popularity logging policy is better in the case where *flips*=0.

Table 6.4: Simulated A/B test results on the RecoGym simulator using: P=1000, U=1000, organic only sessions=20 000.

Agent	Type	CTR (%)	CTR (%)
		flips=0	*flips*=500
Log Reg	(bandit)	1.26	1.30
CB	(bandit)	1.38	1.29
ItemKNN	(organic)	1.39	0.87
MultiVAE	(organic)	**2.43**	1.15
BLO	(organic)	**2.42**	1.13
BLOB-NQ	(combined)	**2.40**	1.51
BLOB-MNQ	(combined)	**2.39**	**1.62**
Random		1.13	1.12

6.5 Conclusion

We focus on a particular recommendation task, one where a user profile is defined by a history of items in a catalogue and the recommendation task is to recommend items from the same catalogue. Our model is able to learn both from the organic signal and the bandit signal jointly beating baselines in a range of settings by exploiting the three fundamental distances of recommendation action-history, action-action and history-history.

We use computational techniques which allow large scale Bayesian inference suitable for Recommendation with large catalogues. The local re-parameterization trick was particularly valuable in reducing the variance in our optimisation problem.

BLOB is able to perform well both in situations where next item prediction is a good proxy for recommendations and situations where it is poor. Meeting the performance of pure organic algorithms in settings where the organic signal is sufficient and exceeding all baselines in more realistic scenarios. This strongly validates the value of Bayesian methods to infer in the cases of a signal of varying strength and their practical value thanks to modern developments in Bayesian deep learning.

There are many possible extensions to this work, one is to produce end to end training, i.e. training both the organic and bandit component simultaneously. To apply this approach would require a more complicated training procedure. We also expect there are other useful ways to combine organic and bandit signal, perhaps based on models that avoid the softmax and sigmoid transform such as LDA for the organic and using the approach out lined in (Lumbreras et al., 2018) for the Bandit. Avoiding softmax and sigmoid transforms has both computational advantages and can increase interpretability.

6.6 Appendix

6.6.1 Approximating expectations under the log softmax

The variational lower bound of BLO (and BLOB) contains a log softmax term. An alternative to using the re-parameterization trick is to use The Bouchard bound, which removes the need for Monte Carlo methods. The Bouchard bound introduces a further approximation and additional variational parameters a, ξ but produces an analytical bound:

$$\begin{aligned}
&\mathcal{L} \geq \mathcal{L}_{\text{Bouch}} = \left(\sum_{t}^{T} \boldsymbol{\Psi}_{vt} \boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{vt} \right) \\
&- T[a + \sum_{p}^{P} \frac{\boldsymbol{\Psi}_p \boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_p - a - \xi_p}{2} \\
&+ \lambda_{\text{JJ}}(\xi_p)\{(\boldsymbol{\Psi}_p \boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_p - a)^2 + \boldsymbol{\Psi}_p \boldsymbol{\Sigma}_{q_\omega} \boldsymbol{\Psi}_p^T - \xi_p^2\} + \log(1 + e^{\xi_p})] \\
&- \frac{K}{2} \log(2\pi) - \frac{1}{2}\{\boldsymbol{\mu}_{q_\omega}^T \boldsymbol{\mu}_{q_\omega} + \text{trace}(\boldsymbol{\Sigma}_{q_\omega})\} + \frac{1}{2} \log |2\pi e \boldsymbol{\Sigma}_{q_\omega}|.
\end{aligned}$$

Because the Bouchard bound causes the softmax to decompose into a sum, we can avoid the expensive normalization by subsampling some of the terms in the softmax.

$$\begin{aligned}
&\hat{\mathcal{L}}_{\text{Bouch}}(v_1, ..., v_T, n_1, ...n_S, \Xi, \boldsymbol{\Psi}) = \left(\sum_{t}^{T} \boldsymbol{\Psi}_{vt} \boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{vt} \right) \\
&- T[a + \frac{P}{S} \sum_{s'=1}^{S} \frac{\boldsymbol{\Psi}_{n_{s'}} \boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{n_{s'}} - a - \xi_{n_{s'}}}{2} \\
&+ \lambda_{\text{JJ}}(\xi_{n_{s'}})\{(\boldsymbol{\Psi}_{n_{s'}} \boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{n_{s'}} - a)^2 + \boldsymbol{\Psi}_{n_{s'}} \boldsymbol{\Sigma}_{q_\omega} \boldsymbol{\Psi}_{n_{s'}}^T - \xi_{n_{s'}}^2\} + \log(1 + e^{\xi_{n_{s'}}})] \\
&- \frac{K}{2} \log(2\pi) - \frac{1}{2}\{\boldsymbol{\mu}_{q_\omega}^T \boldsymbol{\mu}_{q_\omega} + \text{trace}(\boldsymbol{\Sigma}_{q_\omega})\} + \frac{1}{2} \log |2\pi e \boldsymbol{\Sigma}_{q_\omega}|.
\end{aligned}$$

where $v_1, ..., v_T$ are the items associated with the session and $n_1, ...n_S$ are $S < P$ negative items randomly sampled, and $\lambda_{\text{JJ}}(\cdot)$ is the Jaakola and Jordan function (Jaakkola and Jordan, 1997):

$$\lambda_{\text{JJ}}(\xi) = \frac{1}{2\xi} \left(\frac{1}{1 + e^{-\xi}} - \frac{1}{2} \right).$$

This algorithm is similar to the word2vec algorithm (Mikolov et al., 2013), but without any non-probabilistic heuristics.

6.6.2 Log concavity bound

The log concave bound (Ruiz et al., 2018; Blei and Lafferty, 2005; Bouchard, 2007) also breaks the log softmax into a sum

$$\begin{aligned}
\log p(v_1, .., v_T, \boldsymbol{\omega}_u | \boldsymbol{\Psi}) &= \left(\sum_{t}^{T} \boldsymbol{\Psi}_{v_t} \boldsymbol{\omega}_u + \boldsymbol{\rho}_{v_t} \right) - T \log \left\{ \sum_{p}^{P} \exp(\boldsymbol{\Psi}_p \boldsymbol{\omega}_u + \boldsymbol{\rho}_p) \right\} - \frac{K}{2} \log(2\pi) - \frac{1}{2} \boldsymbol{\omega}_u^T \boldsymbol{\omega}_u \\
&\geq \left(\sum_{t}^{T} \boldsymbol{\Psi}_{v_t} \boldsymbol{\omega}_u + \boldsymbol{\rho}_{v_t} \right) - T\phi \left\{ \sum_{p}^{P} \exp(\boldsymbol{\Psi}_p \boldsymbol{\omega}_u + \boldsymbol{\rho}_p) \right\} + T \log \phi + T \\
&- \frac{K}{2} \log(2\pi) - \frac{1}{2} \boldsymbol{\omega}_u^T \boldsymbol{\omega}_u \\
&= L_{\log}.
\end{aligned}$$

Taking an expectation under q of this lower bound gives:

$$E_{q(\omega)}[L_{\text{log}}] = \mathcal{L}_{log} = \left(\sum_t^T \boldsymbol{\Psi}_{vt}\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{vt}\right) - T\phi\{\sum_p^P \exp(\boldsymbol{\Psi}_p\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_p + \frac{1}{2}\boldsymbol{\Psi}_p\boldsymbol{\Sigma}_{q_\omega}\boldsymbol{\Psi}_p^T)\} + \log\phi + 1$$
$$- \mathcal{KL}(Q||P).$$

A fast approximation of the bound can be retrieved by subsampling the items in the catalogue:

$$\hat{\mathcal{L}}_{log}(v_1, .., v_T, n_1, n_{S_{\text{neg}}}) = \left(\sum_t^T \boldsymbol{\Psi}_{vt}\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{vt}\right) - \mathcal{KL}(Q||P)$$
$$- T\frac{P}{S_{\text{neg}}}\phi\{\sum_{s'}^{S_{\text{neg}}} \exp(\boldsymbol{\Psi}_{n_{s'}}\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_{n_{s'}} + \frac{1}{2}\boldsymbol{\Psi}_{n_{s'}}\boldsymbol{\Sigma}_{q_\omega}\boldsymbol{\Psi}_{n_{s'}}^T)\} + T\log\phi + T.$$

Finally the one vs each bound (Titsias, 2016) also breaks the log softmax into a sum without introducing any variational parameter whatsoever.

We can also use a variational auoto-encoders for a, ξ in the case of the Bouchard bound and ϕ in the case of the log concave bound to prevent variational parameters growing with the size of the dataset. This is similar to the augment and reduce approach (Ruiz et al., 2018) but has no requirement to be in complete data exponential family form.

The computational impact of turning the log softmax into a sum computationally is driven by P and GPU size. If P is small compared to the GPU it may be preferable to avoid using any additional approximations and compute the full softmax using the re-parameterization trick.

6.6.3 The EM Algorithm - an alternative to the VAE

Standard EM algorithm

If the parameters $\boldsymbol{\Psi}, \boldsymbol{\rho}$ are already known then the posterior over the user embedding $\boldsymbol{\omega}$ may be calculated by optimizing the lower bound using the following variational EM algorithm. The EM algorithm exploits the fact that the Bouchard bound is quadratic and conjugate to the Gaussian distribution. The algorithm here is the *dual* of the one presented in Bouchard (2007) as we assume the embedding $\boldsymbol{\Psi}$ is fixed and $\boldsymbol{\omega}$ is updated where the algorithm they present does the opposite. The EM algorithm consists of cycling the following update equations:

$$\boldsymbol{\Sigma}_{q_\omega}^{-1} = I_k + 2T\sum_p \lambda_{\text{JJ}}(\xi_p)\boldsymbol{\Psi}_p^T\boldsymbol{\Psi}_p,$$

$$\boldsymbol{\mu}_{q_\omega} = \boldsymbol{\Sigma}_{q_\omega}\left((\sum_t^T \boldsymbol{\Psi}_{vt}^T) - T\left[\sum_p^P\{\frac{1}{2} + 2(\boldsymbol{\rho}_p - a)\lambda_{\text{JJ}}(\xi_p)\}\boldsymbol{\Psi}_p^T\right]\right),$$

$$a = \frac{-1 + \frac{P}{2} + \sum_p 2\lambda_{\text{JJ}}(\xi_p)(\boldsymbol{\Psi}_p\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_p)}{2\sum_p \lambda_{\text{JJ}}(\xi_p)},$$

$$\xi_p = h(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, a, \boldsymbol{\Sigma}_{q_\omega}, \boldsymbol{\rho}_q) = \sqrt{\boldsymbol{\Psi}_p\boldsymbol{\Sigma}_{q_\omega}\boldsymbol{\Psi}_p^T + (\boldsymbol{\Psi}_p\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_p - a)^2}.$$

Fast online EM algorithm

We further note that the EM algorithm is (with the exception of the a variational parameter) a fixed point update (of the natural parameters) that decomposes into a sum. The terms in the sum come from the softmax in the denominator. After substituting a co-ordinate descent update of a with a gradient descent step update, then the entire fixed point update becomes a sum:

$$(\boldsymbol{\Sigma}_{q_\omega}^{-1})^{\text{new}} = I_k + 2\sum_{p} \lambda_{\text{JJ}}(h(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, a, \boldsymbol{\Sigma}_{q_\omega}, \boldsymbol{\rho}_q))\boldsymbol{\Psi}_p^T\boldsymbol{\Psi}_p,$$

$$(\boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega})^{\text{new}} = (\sum_{t}^{T} \boldsymbol{\Psi}_{v_t}^T) - T\left[\sum_{p}^{P}\{\frac{1}{2} + 2(\boldsymbol{\rho}_p - a)\lambda_{\text{JJ}}\{h(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, a, \boldsymbol{\Sigma}_{q_\omega}, \boldsymbol{\rho}_q)\}\}\boldsymbol{\Psi}_p^T\right]$$

$$a^{\text{new}} = a + \frac{-1 + \frac{P}{2}}{2} + \sum_{p} \lambda_{\text{JJ}}\{h(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, a, \boldsymbol{\Sigma}_{q_\omega}, \boldsymbol{\rho}_q)\}(\boldsymbol{\Psi}_p\boldsymbol{\mu}_{q_\omega} + \boldsymbol{\rho}_p) - a\lambda_{\text{JJ}}\{h(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, a, \boldsymbol{\Sigma}_{q_\omega}, \boldsymbol{\rho}_q)\}$$

That is the EM algorithm can be written:

$$\left((\boldsymbol{\Sigma}_{q_\omega}^{-1})^{\text{new}}, (\boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega})^{\text{new}}, a^{\text{new}}\right) = \sum_{p}^{P} g(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, \boldsymbol{\Sigma}_{q_\omega}^{-1}, \boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega}, a).$$

As noted in Cappé and Moulines (2009) when an EM algorithm can be written as a fixed point update over a sum, then the Robbins-Monro algorithm can be applied. Allowing updates of the form (p is chosen randomly):

$$\begin{aligned}(\boldsymbol{\Sigma}_{q_\omega}^{-1})^{(\text{s})},&(\boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega})^{(\text{s})}, a^{(\text{s})} \\ &= (1 - \Delta_s)\left((\boldsymbol{\Sigma}_{q_\omega}^{-1})^{(\text{s}-1)}, (\boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega})^{(\text{s}-1)}, a^{(\text{s}-1)}\right) \\ &+ \Delta_s g(\boldsymbol{\Psi}_p, \boldsymbol{\rho}_p, (\boldsymbol{\Sigma}_{q_\omega}^{-1})^{(\text{s}-1)}, (\boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega})^{(\text{s}-1)}, a^{(\text{s}-1)}).\end{aligned}$$

where Δ is a slowly decaying Robbins Monro sequence (Robbins and Monro (1951b)) with $\Delta_1 = 1$ (meaning no initial value of $(\boldsymbol{\Sigma}_{q_\omega}^{-1})^{(0)}, (\boldsymbol{\Sigma}_{q_\omega}^{-1}\boldsymbol{\mu}_{q_\omega})^{(0)}, a^{(0)}$) is needed. For large P this algorithm is many times faster than the generic EM algorithm. Note that (unusually) the Robbins Monro algorithm is applied to the softmax of a large categorical variable and not to individual records under a conditionally independent assumption.

There are other variational bounds that may be considered for this problem most notably the tilted bound (Knowles and Minka, 2011). For the tilted bound the known fixed point algorithms are not guaranteed to be stable and are not always stable in practice (Nolan and Wand, 2017; Rohde and Wand, 2016) so extra methods such as line searches need to be considered. The tilted bound also does not decompose into a sum. We do not further consider alternative bounds.

The computational cost of this algorithm depends on the number of products P linearly and the embedding size K cubicly, if P and K are modest it can take less than a second making it potentially deployable at prediction time. In practice we found the cost of large P might be prohibitive due to the sums over all P embeddings, in these cases a variational auto-encode described in the next section, is to be preferred.

6.6.4 Next Item Prediction

The predictive distribution required to do next item prediction is also not trivial in this case, i.e. approximating:

$$p(v_{u,T+1}|v_{u,1}, .., v_{u,T}) = \int p(v_{u,T+1}|\boldsymbol{\omega}, \boldsymbol{\Psi}, \boldsymbol{\rho})p(\boldsymbol{\omega}|v_{u,1}, .v_{u,T})d\boldsymbol{\omega}_u$$

is not trivial even if $p(\boldsymbol{\omega}|v_{u,1},..v_{u,T_u})$ is approximated with a Gaussian distribution $\boldsymbol{\omega}_u|v_1,..v_T \sim \mathcal{N}(\boldsymbol{\mu}_{q_\omega}, \boldsymbol{\Sigma}_{q_\omega})$. We are interested in computing:

$$p(v_{n+1}|v_1,...v_n) \approx \mathop{\mathbb{E}}_{q(\boldsymbol{\omega})}\left[\frac{\exp(\boldsymbol{\Psi}_v\boldsymbol{\omega}+\boldsymbol{\rho})}{\sum_{v'}\exp(\boldsymbol{\Psi}_{v'}\boldsymbol{\omega}+\boldsymbol{\rho})}\right].$$

We considered using a Monte Carlo based approximation, first by drawing S samples:

$$\boldsymbol{\omega}^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}_{q_\omega}, \boldsymbol{\Sigma}_{q_\omega}),$$

$$p(v_{n+1}|v_1,...v_n) \approx \frac{1}{S}\sum_{s}^{S}\frac{\exp(\boldsymbol{\Psi}_v\boldsymbol{\omega}^{(s)}+\boldsymbol{\rho})}{\sum_{v'}\exp(\boldsymbol{\Psi}_{v'}\boldsymbol{\omega}^{(s)}+\boldsymbol{\rho})},$$

as well as using a number of fast approximations such as:

$$p(v_{n+1}|v_1,...v_n) \approx \frac{\exp(\boldsymbol{\Psi}_v\boldsymbol{\mu}_{q_\omega}+\boldsymbol{\rho})}{\sum_{v'}\exp(\boldsymbol{\Psi}_{v'}\boldsymbol{\mu}_{q_\omega}+\boldsymbol{\rho})}.$$

while we investigated more complex approximations (such as normalizing the exponential of the lower bound) we did not find they helped in practice.

Chapter 7

Fast Offline Learning for One-Item Recommendation

Abstract

Personalised interactive systems such as recommender systems require selecting relevant items from massive catalogs dependent on context. Reward-driven offline optimisation of these systems can be achieved by a relaxation of the discrete problem resulting in policy learning or REINFORCE style learning algorithms. Unfortunately, this relaxation step requires computing a sum over the entire catalogue making the complexity of the evaluation of the gradient (and hence each stochastic gradient descent iterations) linear in the catalogue size. This calculation is untenable in many real world examples such as large catalogue recommender systems, severely limiting the usefulness of this method in practice. In this chapter, we derive an approximation of these policy learning algorithms that scale logarithmically with the catalogue size. Our contribution is based upon combining three novel ideas: a new Monte Carlo estimate of the gradient of a policy, the self normalised importance sampling estimator and the use of fast maximum inner product search at training time. Extensive experiments show that our algorithm is an order of magnitude faster than naive approaches yet produces equally good policies.

Contents

7.1 Introduction

Large Scale Recommender systems are helping users navigate the enormous amount of content present on the internet, allowing them to identify relevant items. From movie recommendation, basket completion to ad placement, all of these systems need to make decisions in an accurate and fast manner. In this work, we cast the problem of recommendation in the offline contextual bandit framework Swaminathan and Joachims (2015a); Dudík et al. (2014). Given a context x, the decision system performs an action a, the context then interacts with the action recommended and we receive a reward $r(a, x)$. We represent the recommender system as a stochastic parametric policy $\pi_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{A})$, which given a context $x \in \mathcal{X}$, defines a probability distribution over the discrete action space $\mathcal{A}$ of size P. We suppose that the contexts x are stochastic and coming from an unknown distribution ν on $\mathcal{X}$. Our objective is to maximize w.r.t to our parameter θ the average reward over contexts and actions performed by π_θ. It can be written as:

$$\begin{aligned} R(\pi_\theta) &= \mathbb{E}_{x\sim\nu(\mathcal{X}), a\sim\pi_\theta(.|x)}[r(a,x)] && (7.1)\\ &= \mathbb{E}_{x\sim\nu(\mathcal{X})}[\sum_{a\in\mathcal{A}} \pi_\theta(a|x) r(a,x)] && (7.2)\end{aligned}$$

In real world applications, we usually have access to a finite number of context observations $\{x_i\}_{i=1}^N$ and a reward estimator $\hat{r}(a, x)$ built depending on the application and the task our system is trying to solve. We define an empirical estimator aligned with Equation (7.1) by:

$$\begin{aligned} \hat{R}(\pi_\theta) &= \frac{1}{N}\sum_{i=1}^N \mathbb{E}_{a\sim\pi_\theta(.|x_i)}[\hat{r}(a,x_i)] && (7.3)\\ &= \frac{1}{N}\sum_{i=1}^N \sum_{a\in\mathcal{A}} \pi_\theta(a|x_i)\hat{r}(a,x_i). && (7.4)\end{aligned}$$

This simple equation actually encompasses the majority of the objectives used in the offline bandit literature depending on the reward estimator chosen. For example, if we have access to a bandit dataset with the actions done, their propensities and the reward obtained $\{a_i, p_i, r_i\}_{i=1}^N$, the IPS/Horwitz-Thompson estimator and its clipped variant Swaminathan and Joachims (2015a); Bottou et al. (2013) can be obtained by choosing a reward estimator of the following form :

$$\hat{r}^{\tau}_{IPS}(a, x_i) = \begin{cases} \frac{r_i}{\max(\tau, p_i)} & \text{if } \; a = a_i \\ 0 & \text{otherwise} \end{cases}$$

with τ the clipping factor between 0 and 1, with the original IPS retreived when $\tau = 0$. With the additional assumption that we have access to a reward model $r_{\mathcal{M}}$, we can similarly define the Doubly Robust estimator Dudík et al. (2014) and its clipped variant Su et al. (2020) by choosing a reward estimator of the form :

$$\hat{r}^{\tau}_{DR}(a, x_i) = \begin{cases} \frac{r_i - r_{\mathcal{M}}(a,x_i)}{\max(\tau, p_i)} + r_{\mathcal{M}}(a, x_i) & \text{if } \; a = a_i \\ r_{\mathcal{M}}(a, x_i) & \text{otherwise .} \end{cases}$$

One can also cast other methods Wang et al. (2017) into this framework or just optimize any offline metric by designing an adequate reward estimator $\hat{r}$ and plugging it in the simple objective defined in Equation (7.3). Our method is versatile and can deal with different applications as long as we provide the right reward estimator. In the rest of the chapter, we will not make any assumptions on the reward estimator used unless stated explicitly.

7.2 Parametrizing the Policy

In the problem of recommendation, our objective is to find the best product that matches the current interest of the user. As we deal with distinct enumerable products, we consider our action space discrete and we opt naturally for a policy, that is, conditioned on the context x, of the softmax form (Swaminathan and Joachims, 2015a):

$$\pi_\theta(a|x) = \frac{\exp\{f_\theta(a,x)\}}{\sum_b \exp\{f_\theta(b,x)\}} = \frac{\exp\{f_\theta(a,x)\}}{Z_\theta(x)}$$

with f_θ a parametric transformation of the context and the action that encodes the relevance of the action a for the context x. After training the policy and given a context x, the best recommendation online is retrieved by computing the action that maximizes the scores:

$$a_x^* = \text{argmax}_a \ f_\theta(a,x). \tag{7.5}$$

This recommendation needs to be done in milliseconds over large catalog sizes making the form of f_θ crucial to performing Equation (7.5) rapidly. By restricting the policy to the following form:

$$f_\theta(a,x) = h_\Xi(x)^T\beta_a$$

with $\theta = [\Xi, \beta]$, h_Ξ a transform that creates a user embedding and β_a the item embeddings. Equation (7.5) becomes:

$$a_x^* = \text{argmax}_a \ h_\Xi(x)^T\beta_a$$

which is precisely the problem that approximate MIPS: Maximum Inner Product Search algorithms Malkov and Yashunin (2020); Johnson et al. (2019) solves quickly (if approximately). This is achieved by first building a fixed index with a particular structure Malkov and Yashunin (2020) allowing fast identification of the item with the largest inner product with the query $h_\Xi(x)$. With this index, solving Equation (7.5) is done in a time complexity logarithmic in the the size of the action space P making it possible to deliver recommendations quickly from a large catalog.

7.3 Optimizing the Objective

In large scale recommendation problems, we usually deal with a considerable amount of observations making stochastic gradient descent and its variants Ruder (2016) suitable for such application. As our objective is decomposable, we are interested in the gradient of $\hat{R}_i(\pi_\theta) = \mathbb{E}_{a\sim\pi_\theta(.|x_i)}[\hat{r}(a,x_i)]$ for a single observation x_i. A gradient can be derived using the log trick Williams (1992):

$$\nabla_\theta \hat{R}_i(\pi_\theta) = \mathbb{E}_{a\sim\pi_\theta(.|x_i)}[\hat{r}(a,x_i)\nabla_\theta \log \pi_\theta(a|x_i)]. \tag{7.6}$$

When the action space is of small size Swaminathan and Joachims (2015a); Dudík et al. (2014); Su et al. (2020), this gradient can be computed exactly as an expectation over the discrete distribution π_θ. Once the size of the catalog P is in the order of millions, an exact gradient update becomes a bottleneck for the optimization process because of the complexity of the following computations:

1 - Computing $\nabla_\theta \log \pi_\theta(.|x_i)$: We need to deal with the normalizing constant $Z_\theta(x_i)$ present in the computation of $\nabla_\theta \log \pi_\theta(.|x_i)$. Indeed, $Z_\theta(x_i)$ is a sum over all the action space and its computation needs to be avoided if we hope to reduce the complexity of the gradient update.

2 - Computing the expectation: The expectation is a sum over all the action space and is obviously computed in $\mathcal{O}(P)$. To avoid this expensive sum, we can resort to sampling from π_θ to approximate the gradient. This allows us to obtain the REINFORCE estimator Williams (1992), an unbiased estimator of the expectation but does not change the complexity of the method which stays linear in the catalog size. Indeed, sampling needs the computation of $Z_\theta(x_i)$ or can be done with the gumbel trick Huijben et al. (2021) which both scale in $\mathcal{O}(P)$. To lower the time complexity, we need to avoid sampling directly from π_θ and use Monte Carlo techniques instead such as importance sampling Owen (2013) with carefully chosen proposals to achieve fast sampling and accurate gradient approximation.

The proposed approach will try to reduce the complexity of the gradient computation by separately dealing with the issues mentioned above in a principled manner. This will achieve a faster offline training, and will hopefully not suffer a loss in the quality of the policy learned.

Remark. The problem we are interested in should not be confused with the maximum log-likelihood problem which is to maximize:

$$\mathcal{L} = \sum_n h_\Xi(x_n)^T \beta_{a_n} - \log(\sum_i \exp(h_\Xi(x_n)^T \beta_i)). \tag{7.7}$$

In the context of policy learning, we are seeking a decision rule that maps x to the highest reward action according to $r(a,x)$. This is different from maximising Equation (7.7) which enables us to find the model $P(a|x,\Xi)$ that fits the data the most and totally ignores $r(a,x)$. While both approaches slow down when the catalogue size P is very large due to the sum, they are not the same problem. Many existing methods in the literature have been proposed to optimize Equation (7.7) when dealing with large action spaces. These include Bengio and Senecal (2003); Blanc and Rendle (2018); Rawat et al. (2019); Mikolov et al. (2013). There are some overlaps between the two problems above, as both require calculating expectations under a large categorical distribution, but the difference in the loss functions makes the solutions of these problems different as well. For instance, the solution of the policy learning problem is a deterministic policy, putting, conditionally on x, all the mass on the action that maximizes $r(a,x)$. If any of the methods suggested to solve the maximum likelihood problem can be adapted to the policy learning case is beyond the scope of this chapter.

7.4 The Proposed Method

As pointed out in the previous section, we need a workaround to deal with the presence of the normalizing constant in the gradient. For a fixed observations x_i and similar to the derivations found in Masrani et al. (2019), we can push further the computation of $\nabla_\theta \log \pi_\theta(a|x_i)$ to obtain a quantity that does not involve $Z_\theta(x_i)$. Indeed, we have for a fixed action a:

$$\begin{aligned}
\nabla_\theta \log \pi_\theta(a|x_i) &= \nabla_\theta f_\theta(a,x_i) - \nabla_\theta \log Z_\theta(x_i) \\
&= \nabla_\theta f_\theta(a,x_i) - \frac{\nabla_\theta Z_\theta(x_i)}{Z_\theta(x_i)} \\
&= \nabla_\theta f_\theta(a,x_i) - \sum_b \pi_\theta(b|x_i) \nabla_\theta f_\theta(b,x_i) \\
&= \nabla_\theta f_\theta(a,x_i) - \mathbb{E}_{b\sim\pi_\theta(.|x_i)}[\nabla_\theta f_\theta(b,x_i)]
\end{aligned}$$

Injecting the above expression of $\nabla_\theta \log \pi_\theta(a|x_i)$ in Equation (7.6) leads us to the following covariance gradient Masrani et al. (2019):

$$\nabla_\theta \hat{R}_i(\pi_\theta) = \mathbf{Cov}_{a\sim\pi_\theta(.|x_i)}[\hat{r}(a,x_i), \nabla_\theta f_\theta(a,x_i)] \tag{7.8}$$

with $\mathbf{Cov}[A, \boldsymbol{B}] = \mathbb{E}[(A - \mathbb{E}[A]).(\boldsymbol{B} - \mathbb{E}[\boldsymbol{B}])]$, which is a covariance between A a scalar function, and $\boldsymbol{B}$ a vector. To estimate this, we first estimate the two inner expectations which are then used in estimating the outer expectation. Note that the covariance in Equation (7.8) is between two deterministic functions of one single random variable a that follows the distribution of π_θ. The gradient expression in Equation (7.8) has an intuitive interpretation, as the covariance quantifies how two quantities evolve together, gradient descent using (7.8) will move in directions where the reward and the gradient of the relevance function have the same evolution w.r.t to the action drawn from the policy π_θ enabling our algorithm to favor reward maximization.

The new gradient formula helps us get rid of the normalizing constant present in $\nabla_\theta \log \pi_\theta(a|x_i)$ but transforms the expectation we had in Equation (7.6) into a double expectation; a covariance between the reward estimator and the gradient of our relevance function f_θ. By exploiting this identity, we remove the requirement to compute $Z_\theta(x_i)$ which scales linearly with the catalogue size if we have available to us two (or more) samples from the policy $\pi_\theta(.|x_i)$ (computing covariances requires multiple samples). Unfortunately, sampling from $\pi_\theta(.|x_i)$ also scales in $\mathcal{O}(P)$ so it seems that no progress has been made.

Wanting to avoid sampling from the current policy π_θ, we use self normalized importance sampling Owen (2013) to approximate the expectations without relying on the computation of the normalizing constant Z_θ. Indeed, for a fixed x_i, if we have access to a discrete proposal q over the action space, one can build an estimator of the expectation of a general function g under $\pi_\theta(.|x_i)$ by:

$$\mathbb{E}_{\pi_\theta(.|x_i)}[g(a)] \approx \sum_{s=1}^{S} \bar{\omega}_s g(a_s)$$

with $a_s \sim q \quad \forall s$, $\omega_s = \frac{\exp\{f(a_s, x_i)\}}{q(a_s)}$ and $\bar{\omega}_s = \frac{\omega_s}{\sum_{s'=1}^{S} \omega_{s'}}$.

This algorithm removes the dependency on the catalogue size by avoiding the computation of $Z_\theta(x_i)$. The cost for this is that the estimator is now biased with a bias decreasing with how close the proposal q is to the policy π_θ Agapiou et al. (2017). This means that if we want to exploit self normalized importance sampling efficiently, we need to have access to a proposal q that is **fast** to sample from, easy to **evaluate** as its needed to compute the weights $\bar{\omega}_s$ and **close** to the actual policy π_θ to reduce the bias and the variance of the method. To build such proposal q that respects the three conditions, we need an additional assumption on the parameters learned of our policy π_θ.

Assumption 1: The item embedding matrix β is fixed and we are only interested in learning the user embedding transform h_Ξ, which means $\Xi = \theta$.

Making this assumption in the context of recommendation is reasonable Koch et al. (2021). We can learn different representations of the items from collaborative filtering data, text data or even images making learning meaningful embeddings possible without relying on the downstream task we are trying to solve.

This assumption allows us to fully exploit approximate MIPS algorithms in the training phase by building an index over the item embeddings β and fixing it before beginning the optimization of our policy. Indeed, while we can modify existing embeddings in the index for a logarithmic complexity Malkov and Yashunin (2020) in the training phase, this procedure will further slow down the method as we need to update the items in the index for each training iteration.

Making **Assumption 1** simplifies drastically the procedure as β are considered fixed and we only need to compute the index once before the start of the training of our policy. With the help of the approximate MIPS index, and for any x_i, we define the proposal q as a mixture between a distribution over the K most probable actions under $\pi_\theta(.|x_i)$ retrieved by the approximate

MIPS algorithm i.e $\alpha_K(x_i) = \text{argsort}(h_\theta(x_i)^T\beta)_{1:K}$ and a uniform distribution over actions. It can be expressed as:

$$q_{K,\epsilon}(a|x_i) = \begin{cases} \frac{\epsilon}{P} + (1-\epsilon)\kappa(a|x_i), & \text{if } \; a \in \alpha_K(x_i) \\ \frac{\epsilon}{P}, & \text{otherwise .} \end{cases}$$

Where ϵ is a parameter that controls the mixture and

$$\kappa(a|x_i) = \frac{\exp(h_\theta(x_i)^T\beta_a)}{\sum_{a' \in \alpha_K(x_i)} \exp(h_\theta(x_i)^T\beta_{a'})} \mathbb{1}[a \in \alpha_K(x_i)].$$

This proposal answers the necessary conditions to make self normalised importance sampling works efficiently:

- It is **fast** to sample from as a mixture of a uniform distribution and a distribution constructed with approximate MIPS making the time complexity $\mathcal{O}(\log P)$ Malkov and Yashunin (2020). Indeed, solving the argsort, thus constructing α_K can be done logarithmically in the catalog size with the help of approximate MIPS.
- Easy to **evaluate** as once we have the set α_K, the computation will require at maximum a sum over the top K retrieved actions with $K \ll P$.
- **Close** to π_θ as it exploits information about the top actions under π_θ and covers well the early stage (when π_θ close to uniform) and late stage (when π_θ is degenerate on the top actions) of the optimization process.

```
Algorithm 4: Fast Offline Policy Learning
Inputs: D = $\{x_i\}_{i=1}^N$, reward estimator $\hat{r}$, the item embeddings $\beta$
Parameters: $T \geq 1, \alpha, K, S \geq 2, \epsilon \in [0,1]$
Initialise: $\theta = \theta_0$, approximate MIPS index for $q_{\epsilon,K}$
for $t = 0$ to $T$ do
    $x \sim D$
    query $h_\theta(x)$ to get the approx. top $K$ actions set $\alpha_K$
    build the proposal $q_{\epsilon,K}(.|x)$
    sample $a_1, ..., a_S \sim q_{\epsilon,K}(.|x)$
    Estimate the covariance gradient:
    $\hat{r}_s = \hat{r}(a_s, x), \nabla f_s = \nabla_\theta f_\theta(a_s, x) \quad \forall s$
    $\omega_s \leftarrow \frac{\exp\{f(a_s,x)\}}{q_{\epsilon,K}(a_s|x)}, \bar{\omega}_s \leftarrow \frac{\omega_s}{\sum_s \omega_s} \quad \forall s$
    $\bar{r} \leftarrow \sum_{s=1}^S \bar{\omega}_s \hat{r}_s, \; \bar{\nabla f} \leftarrow \sum_{s=1}^S \bar{\omega}_s \nabla f_s$
    $grad_\theta \leftarrow \sum_{s=1}^S \bar{\omega}_s [\hat{r}_s - \bar{r}][\nabla f_s - \bar{\nabla f}]$
    Update the policy parameter $\theta$:
    $\theta \leftarrow \theta - \alpha grad_\theta$
end
return $\theta$
```

The performance of the self normalized importance sampling algorithm using our proposal is controlled by the number of Monte Carlo samples S, the mixture parameter ϵ and the number of items returned by the maximum inner product search K. This performance can also be impacted by the parameters of the approximate MIPS algorithms that trade-off speed of retrieval for the accuracy of the argsort, changing the parameters seemed to have little to no impact on the study so we decided to fix them for the whole experiments.

	Catalog size	Number of users
Twitch	790K	500K
GoodReads	2.33M	300K

Table 7.1: The statistics of the datasets after processing

By combining the self normalized importance sampling algorithm with our mixture proposal, a stochastic gradient descent Ruder (2016) version of the algorithm we suggest can be described in Algorithm 4.

This procedure is compatible with any stochastic first order optimization algorithm Ruder (2016); Kingma and Ba (2014). In the next section, we will validate the approach on real world datasets and study the impact of our method on the training speed and the quality of the learned policy.

7.5 Experimental Results

We test our approach on a session completion task using a collaborative filtering dataset. In the training phase, we split randomly the user-item interaction session into two parts, X and Y. X is used as observed and we will condition on it to predict items that are found in Y, or in other words, predict items that complements the vector X. Our goal is to build an algorithm that given a user-item interaction vector X_{new}, predict or recommend items that may interest the user. This can be cast into an offline bandit framework where our policy π_θ takes the observed part X as a context, recommends an item a and receives the binary reward $\hat{r}(a, X) = \mathbb{1}[a \in Y]$. The goal then is to learn a policy π_θ that will maximize the reward $\hat{r}$ allowing it to solve the session completion task. Note that our method is very versatile and can deal with different problems as long as they can be cast into an offline policy learning problem as described in the introduction. We chose a session completion task for simplicity as user-item interaction datasets are public making the experiments easily reproducible.

To prepare our experiment, we begin by splitting the user-item sessions into two independent sessions with the same number of interactions: the observed items X and the complementary items Y. Once this split performed, we also split the whole dataset into a train $D_{train} = [X_{train}, Y_{train}]$ and test split $D_{test} = [X_{test}, Y_{test}]$. Given the train split D_{train}, we use the observed contexts X_{train} to first compute the item embeddings β using SVD matrix decomposition Klema and Laub (1980) of dimension $[P, L]$ with $L \ll P$ that will be fixed to create the approximate MIPS index. This index is then used by our proposal q in the learning phase, but also used to get the best recommendation rapidly in the testing phase as described in Equation (7.5).

Once we have the item embedding matrix β, we compute the user contexts x as the mean embeddings Koch et al. (2021) of the items that the user interacted with. For X_i the observed item interactions of user i and n_i the number of items the same user interacted with, we define $x_i^{emb} = \frac{1}{n_i} \sum_{a \in X_i} \beta_a$. The obtained vector x_i^{emb} is of dimension L and will be the user context in our experiment, meaning that we will use $D^{emb}_{train} = [x^{emb}_{train}, Y_{train}]$ and $D^{emb}_{test} = [x^{emb}_{test}, Y_{test}]$ instead of D_{train} and D_{test}.

The next step is to parametrize the policy π_θ that we will train using our algorithm. With the item embeddings β fixed, we only need to define the user transform h_θ. We take h_θ to be a linear function of the user context x^{emb} defined above, ie. $h_\theta(x^{emb}) = \theta^T x^{emb}$ making the dimension of the parameter learned θ equals to $[L, L]$. After training the policy π_θ, we validate its performance on the test split D^{emb}_{test} by computing the average reward collected after querying

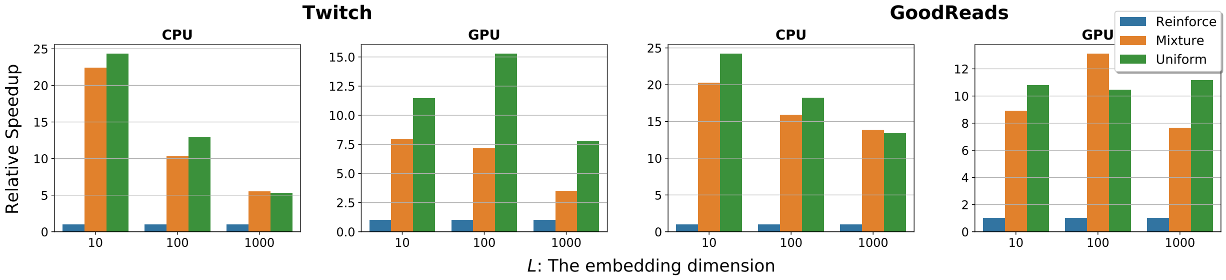


Figure 7.1: The speedups of the proposed algorithms on the Twitch and GoodReads datasets, with and without GPU acceleration.

Dataset	L	rP	rS
Twitch	10	*0.54*	**1.01**
	100	*0.40*	**1.01**
GoodReads	10	*0.83*	**1.00**
	100	*0.71*	**1.01**

Table 7.2: Impact of fixing the product embeddings.

the argmax of our policy using approximate MIPS, ie.

$$R_{test} = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \mathbb{1}[\text{argmax}_a f_\theta(a, x_j^{emb}) \in Y_j]$$

We choose two collaborative filtering datasets with large catalogue sizes to validate our approach. the Twitch dataset (Rappaz et al., 2021) and the GoodReads user-books interaction dataset (Wan and McAuley, 2018; Wan et al., 2019) with both representing a good test bed for the scalability of the proposed methods. We transform the datasets into a user-item interaction matrix with statistics represented in Table 7.1. To build the approximate MIPS index, we use the HNSW algorithm (Malkov and Yashunin, 2020) bundled in the FAISS library (Johnson et al., 2019). The optimization routine is implemented using Pytorch (Paszke et al., 2019), and we opt for the Adam optimizer Kingma and Ba (2014) with a batch size of 32 and a learning rate of 10^{-4} for all the experiments with the twitch dataset and 5.10^{-5} for the goodreads dataset. The source code[1] to reproduce the results has all implementation details.

With the experimental protocol described above, we want to study the speed improvement brought by our method, and the effect of the parameters controlling our proposal $q_{\epsilon,K}$, namely the mixture parameter ϵ and the number of top K retrieved items on the speed and the quality of the policy learned. The rest of the experiments section will be decomposed into different research questions that we would like to answer for a better understanding of the approach proposed.

RQ0 - What is the cost of fixing the embeddings? To verify the impact of making **Assumption 1**, we compare the performance and speed of REINFORCE with the item embeddings β fixed to REINFORCE with β initialised with the SVD decomposition and trained. We report in Table 7.2, for both datasets and two different values of the embedding dimension $L \in \{10, 100\}$, the relative speed-up $rS = T_{trained}/T_{fixed}$ and the relative performance $rP = R_{trained}/R_{fixed}$ with T_{method} and R_{method} are respectively the run time and reward of a method after training.

[1] https://github.com/criteo-research/fopo

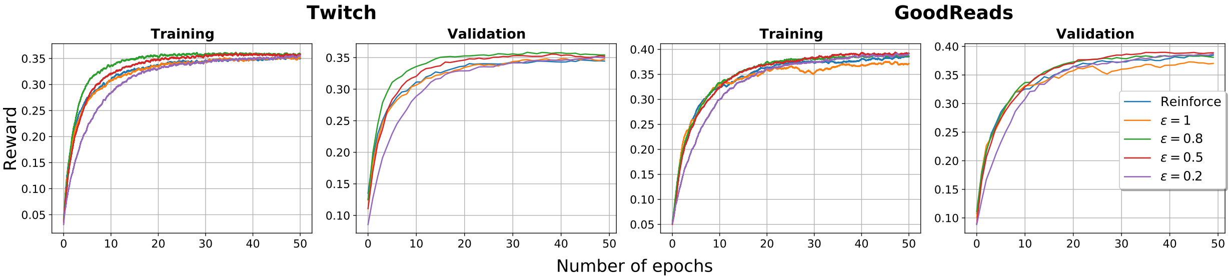


Figure 7.2: The performance of the algorithms while changing the mixture parameter ϵ on both Twitch and GoodReads datasets.

We observe that training β have no to little impact on the training time, but hurts the performance of REINFORCE mainly due to the variance introduced by having more parameters to optimize. Indeed, we observe that increasing the dimension L worsens the relative performance rP in Table 7.2. We conclude that in our experiments, **Assumption 1** can be made as it does not negatively affect the training.

RQ1 - What is the speed gain over REINFORCE? This work addresses the computational inefficiency of vanilla REINFORCE as we previously described. To show that our proposed method is a good solution to the problem presented, we conduct extensive experiments to study the acceleration brought by our methods relatively to our baseline. In this section, we quantify the gains in form of a relative speed-up of the proposed methods compared to REINFORCE; for the same experimental setup, if T_{method} is the wall time of a method, we define the relative speed-up to our baseline as $RS_{\text{method}} = T_{\text{REINFORCE}}/T_{\text{method}}$.

Before we dive into the experiments, we first identify the different parameters that can have a big impact on the running time of the learning algorithm. As Matrix-vector multiplication is naively done in $\mathcal{O}(L^2)$ with L being the embedding dimension, we conduct multiple experiments on the datasets we have at our disposal, on **CPU** and **GPU** devices while changing the embedding dimension L to have a good understanding of the speed gains in different settings. In Figure 7.1, we compare REINFORCE, the algorithm that approximates the gradient expression in Equation (7.6) by samples from the true policy, with the proposed approach for $\epsilon = 1$ for which our proposal becomes uniform, and the mixture algorithm with $\epsilon \neq 1$, represented in these experiments by a run with $\epsilon = 0.8$. Note that the value of ϵ does not affect the running time as long as it is different from 1. In these experiments, we fix $K = 256$ and $S = 1000$.

We can see from Figure 7.1 that we gain significant speed-ups in the different settings considered ranging from 5 to 25 faster offline policy training compared to REINFORCE, with the largest speed-ups observed on CPU machines (no massive parallelization contrary to GPUs) and with small embedding dimension L as the complexity of matrix-vector multiplication $\mathcal{O}(L^2)$ does not dominate the gradient update complexity. We can see that even if we have access to a GPU, and we make L big enough to learn better user embeddings, the speed-up is still interesting as we still obtain 5-10 times speed-up compared to our baseline. Note that all the run times were averaged over 5 epochs, and that these time comparisons can differ depending on the computational resources at one's disposal, but we argue that the differences should be more important when training on **CPU** machines, especially when the user embeddings are easy to compute, making offline policy training on less powerful machines possible. We also expect bigger gains when dealing with larger action spaces, given that our method scales better with growing catalogues.

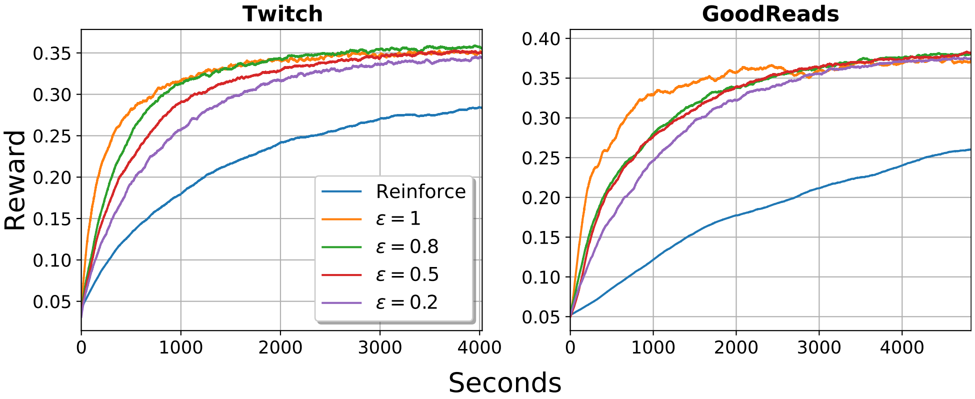


Figure 7.3: Training performance given a fixed time budget.

RQ2 - What is the impact of changing ϵ? We have seen that changing ϵ, especially setting it to 1, can have a significant impact on the iteration cost of the optimization procedure. To get a better understanding of our method, we need to study the effect of the mixture parameter on the quality and thus the performance of the trained policies. In this section, we investigate the impact of changing ϵ on the average reward, on the same problem with an embedding dimension set to $L = 1000$ to make the learning problem difficult, while also fixing the other parameters to $K = 256$ and $S = 1000$. The algorithms were run on the datasets considered for 50 epochs, and **GPU** training was used to be fair to the baseline, as the speed-up gains in this particular setting are the lowest. We plot the results of these runs on Figure 7.2. We observe that, even if REINFORCE has a much bigger time complexity per iteration (scales linearly on the catalogue size), it does not outperform the optimization routines suggested by our approach. Indeed, we can achieve the same level of performance, sometimes performances beyond what REINFORCE can reach with much faster training, so not only our method is faster than REINFORCE, it can also lead to a better optima as we suspect that using the index on the training phase helps the policies be better aligned with how the recommendation is done after deployment, as the same index is used online.

From the same plots, we can also conclude that fixing $\epsilon = 1$, even if it provides the fastest approximation as it boils down to using a uniform proposal, is far from being optimal if our main goal is to obtain the policy with the best average reward. Indeed, the optimal policy is obtained with values of $\epsilon \neq 1$ ($\epsilon = 0.8$ for the Twitch Dataset and $\epsilon = 0.5$ for the Goodreads dataset). Even if this means that we need to try out different values of ϵ to get the best out of our approach, it confirms that our first intuition of building a mixture proposal between the uniform distribution and a TOP-K distribution brings value, in terms of iteration speed and quality of the policy learned as our proposal is expected to behave well in all training phases; $\epsilon \approx 1$ is expected to work well in the beginning of the optimization while ϵ close to 0 is expected to work when the policy is close to convergence.

Figure 7.2 plot the evolution per epoch of the performance of the policies trained with the different algorithms. As the cost of iteration significantly changes depending on the algorithm used, we might be interested in a comparison given a fixed time budget, allowing us to train the policy with different methods for the same amount of time.

We provide Figure 7.3 to shed more light on how the different algorithms perform on the training phase with a fixed time budget. As the uniform proposal provides the fastest training, we consider its running time after 50 epochs the allowed time budget and compare the performance of all the other methods to it given that fixed running time. We can observe that REINFORCE is the worst behaving algorithm as its iteration cost scales linearly with P and that a well-chosen

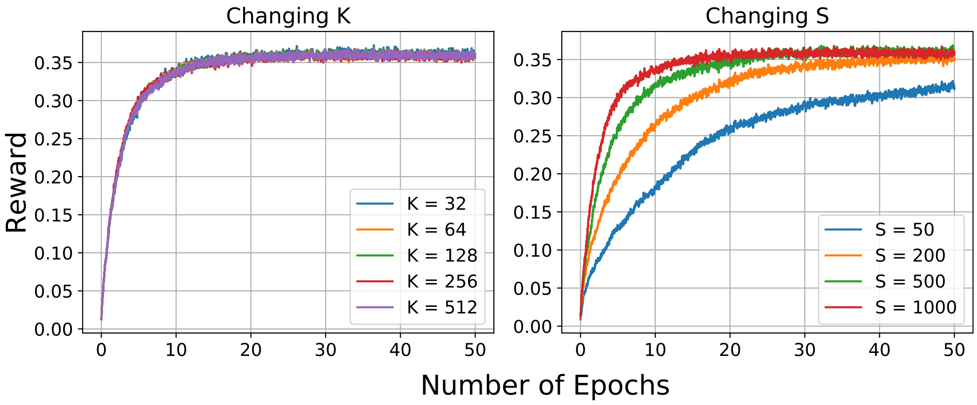


Figure 7.4: The effect of changing K and S on the training.

mixture have always a slight edge over the uniform proposal on both training datasets.

RQ3 - What is the impact of changing the number of the top retrieved items K? As our approach have different hyperparameters, we want to understand how changing them affect the behaviour of the algorithms proposed. To quantify the impact of K, we focus on the Twitch dataset and fix ϵ to 0.8, $L = 1000$ and $S = 1000$. We then try different values for $K \in \{32, 64, 128, 256, 512\}$, the number of top items returned by approximate MIPS. We run the optimization for 50 epochs and plot the reward of the policies on the training phase in Figure 7.4. We observe that the performance is robust to the choice of K as long as it is selected big enough to cover most of the top candidates. Even if it is not apparent on the plot, we can also confirm that the iteration cost is not greatly influenced by the choice of K, as long as it is orders of magnitude smaller than the catalogue size.

RQ4 - What is the impact of changing the number of Monte Carlo samples S? The number of Monte Carlo samples S controls the approximation accuracy of the gradient, as increasing it will reduce the bias (when using the covariance gradient with our proposal $q_{\epsilon,K}$) and the variance of our gradient estimate for an additional computation cost. To understand the impact of S on the training, we restrict ourselves to the Twitch dataset and use $q_{\epsilon=0.8,K}$ as a proposal. We fix $L = 1000$, $K = 256$ and plot the results of changing $S \in \{50, 200, 500, 1000\}$ in Figure 7.4. We run the optimization for 50 epochs. We observe that our policies converge to a better optima when we increase S, and that is expected as less noise is present in the gradient approximation Owen (2013). We also observed from our experiments that the average run time only increased slightly from $S = 50$ to $S = 1000$, suggesting that increasing S to a higher value ($S \ll P$) is beneficial for training policies offline, as the additional computational cost on **GPU** is minimal compared to the convergence speed gains achieved.

7.6 Related Work

There has been a surge of interest in off-policy learning in the last decade. Most of these contributions have focused upon improving the estimate of the reward function, including the development of estimators for the reward representing sophisticated modifications of the estimators $\hat{r}_{IPS}$ and $\hat{r}_{DR}$ introduced in the first section. Correctly estimating the reward is an extremely difficult problem and naive estimators suffer from very high variance, especially when the new policies we want to evaluate are far from the logging policy Agapiou et al. (2017). Many new methods improved the bias-variance trade-off of the reward estimators (Bottou et al., 2013;

Dudík et al., 2014; Su et al., 2020; Wang et al., 2017; Swaminathan and Joachims, 2015b) and tackled the learning problem by using these estimators within the framework of Empirical Risk Minimization Bottou et al. (2013); Dudík et al. (2014); Su et al. (2020), or by leveraging refined statistical learning techniques, giving birth to Sample Variance Penalization (Swaminathan and Joachims, 2015a), Distributionally Robust Counterfactual Risk Minimization (Faury et al., 2020; Sakhi et al., 2020b), PAC-Bayesian Counterfactual Risk Minimization (London and Sandler, 2019) and Imitation Offline Learning (Ma et al., 2019). Alternatively, instead of using an estimator, an explicit reward model can be used (Sakhi et al., 2020a; Jeunen and Goethals, 2021). Our contribution is orthogonal to these developments, as we consider the problem of how to efficiently find a good policy given a reward function. Most past work assumes the action space is small, and hence the optimization problem is tractable. Our algorithm reduces the optimization cost associated with large action spaces, but the estimation challenges of the reward function remain.

Policy learning emerges as the optimisation problem of a reward driven recommender system. Recommender system training is also sometimes formulated as a maximum likelihood approach when training a model (Liang et al., 2018; Steck, 2020) to predict missing entries (e.g. predicting missing elements of the MovieLens dataset (Harper and Konstan, 2015)). Maximum likelihood estimation also suffers from a computational cost that scales in $\mathcal{O}(P)$ but the problem has a different mathematical form to policy learning and methods developed in the maximum likelihood context (Tanielian and Vasile, 2019; Gutmann and Hyvärinen, 2010; Rendle et al., 2009) cannot be applied to policy learning.

There has been limited attention in the literature regarding scaling offline policy learning methods to the problems of recommendations with large discrete action spaces. For example, Dulac-Arnold et al. (2015) considered the acceleration of reinforcement learning Sutton and Barto (2018) which is an online learning framework by definition. Their proposed approach uses Fast K-NN Malkov and Yashunin (2020) algorithms to generate action candidates for the critic to choose from, in contrast to our approach which deals with offline learning and uses approximate MIPS algorithms to define a proposal to better estimate the gradient.

Recently Chen et al. (2019a) showed that offline policy learning methods can perform well on large scale production systems, introducing a correction to the REINFORCE gradient with little focus on the scalability of the method. Our work addresses the computational issues linked to offline policy learning, making it fast to achieve without deteriorating the performance of the obtained policy.

7.7 Conclusion

Offline Policy learning is a powerful paradigm for recommender system training, as it seeks to align the offline optimization problem with the real world reward measured at A/B test time. Unfortunately, the $\mathcal{O}(P)$ cost of training traditional policy learning algorithms has limited the widespread application of these methods in large scale decision systems when P is often very large. To deal with this issue, we introduced an efficient offline optimization routine for softmax policies, tailored for the problem of large scale recommendation. Our algorithm makes the training orders of magnitude faster (up to 30x faster in our experiments). The quality of our policies were at least as good as those found with slower baselines. This work can enable practitioners to explore policy learning methods when dealing with large action spaces without relying on huge computational resources. We hope that the solution provided in this chapter is a first step towards the adoption of offline policy methods for large scale recommender systems.

There are a number of avenues of open research. Self normalized importance sampling produces biased gradients that can affect the convergence of stochastic gradient descent, convergence in this setting is not well studied apart from some special cases (Robbins and Monro, 1951a; Hsieh et al., 2021). We showed that fixing ϵ works well in our experiments, but maybe an adaptive ϵ may work better still. We would expect $\epsilon = 0$ to work well in the early stages of training and $\epsilon \to 1$ in the late stages, but robustly determining how to evolve ϵ is unclear. Finally, we would also want to explore ways to enable fast training without fixing the item embeddings β.

Chapter 8

Fast Offline Learning for Slate Recommendation

Abstract

An increasingly important building block of large scale machine learning systems is based on returning *slates*; an ordered list of items given a query. Applications of this technology include: search, information retrieval and recommender systems. When the action space is large, decision systems are restricted to a particular structure to complete online queries quickly. This chapter addresses the optimization of these large scale decision systems given an arbitrary reward function. We cast this learning problem in a policy optimization framework and propose a new class of policies, born from a novel relaxation of decision functions. This results in a simple, yet efficient learning algorithm that scales to massive action spaces. We compare our method to the commonly adopted Plackett-Luce policy class and demonstrate the effectiveness of our approach on problems with action space sizes in the order of millions.

Contents

8.1 Introduction

Large scale online decision systems, ranging from search engines to recommender systems, are constantly queried to deliver ordered lists of content given contextual information. As an example a user who has just seen the film 'Batman', might be recommended: Superman, Batman Returns, Bird Man, or a user that is reading a page about 'technology' might be interested in other stories about biotechnology, solar power, and large language models. A large scale production system must therefore be able to rapidly respond to a query like 'Batman' or 'technology' with an ordered list of relevant items.

A proven solution to this problem is to generate the ordered list using an approximate maximum inner product search (MIPS) algorithm (Shrivastava and Li, 2014), which at the expense of a constraint in the decision rule provides extremely rapid querying even for massive catalogues. While MIPS based systems are a proven technology at deployment time, the offline optimization of them is not, and the standard algorithm based on policy learning using the Plackett-Luce distribution is infeasibly slow as the algorithm both iterates slowly and requires very large numbers of iterations to converge. Fortunately, other approaches are possible which both iterate faster and require fewer iterations. In this chapter we demonstrate such an algorithm and show it has vastly better convergence properties than competitors.

We denote by $x \in \mathcal{X}$ a context; it can be the history of the user, a search query or even a whole web page. The decision system is tasked to deliver, given the context x, an ordered list of actions $A_K = [a_1, ..., a_K]$, coined *slates*, of arbitrary size K. This slate can be an ad banner, a list of recommended content or search results. Our decision system, given contextual information x, constructs a slate by selecting a subset of actions $\{a_1, ..., a_K\} \subset \mathcal{A}$ from a potentially large discrete set $\mathcal{A}$ and ordering them. Let $P = |\mathcal{A}|$ be the size of the action set. Fixing the slate size K, we model our system by a decision function $d : \mathcal{X} \to \mathcal{S}_K(\mathcal{A})$ that maps contexts x to the space $\mathcal{S}_K(\mathcal{A})$ of ordered lists of size K. Each pair of context x and slate A_K is associated with a reward function[1] $r(A_K, x)$ that encodes the relevance of the slate A_K for x. Our objective is to find decision systems that maximize the expected reward under the unknown distribution of contexts $\nu(\mathcal{X})$:

$$\mathbb{E}_{x \sim \nu(\mathcal{X})} \left[r(A_K = d(x), x) \right].$$

Assuming that we have access to the reward function, the solution of this optimization problem is given by:

$$\forall x \in \mathcal{X}, \quad d(x) = \underset{A_K \in \mathcal{S}_K(\mathcal{A})}{\arg\max}\ r(A_K, x) \tag{8.1}$$

This solution, while optimal, is intractable as it requires, for a given context x, the search in the enormous space $\mathcal{S}_K(\mathcal{A})$ of size $\mathcal{O}(P^K)$ to find the best slate. Instead of maximizing the expected reward over the whole space, we want to restrict ourselves to decision functions of practical use. In this direction, we begin by defining the parametric relevance function $f_\theta : \mathcal{A} \times \mathcal{X} \to \mathbb{R}$:

$$\forall (a, x) \in \mathcal{A} \times \mathcal{X}, \quad f_\theta(a, x) = h_\Xi(x)^\intercal \beta_a$$

with the learnable parameter $\theta = [\Xi, \beta]$, a parametric transform $h_\Xi : \mathcal{X} \to \mathbb{R}^L$ that creates a context embedding of size L, and β_a the embedding of action a in $\mathbb{R}^L$. the embedding dimension L is usually taken to be much smaller than P, the size of the action space. We then define our decision function:

$$\forall x \in \mathcal{X}, \quad d_\theta(x) = \underset{a \in \mathcal{A}}{\text{argsort}^{\text{K}}} \left\{ h_\Xi(x)^\intercal \beta_a \right\}. \tag{8.2}$$

[1] motivated by business metrics and/or users engagement.

with $\text{argsort}^{\text{K}}$ the argsort function truncated at the K-th item. Restricting the space of decision functions to this parametric form reduces the complexity of returning a slate for the context x from $\mathcal{O}(P^K)$ to $\mathcal{O}(P\log K)$. This complexity can be further decreased. Equation (8.2) transforms the querying problem to the MIPS: Maximum Inner Product Search problem (Shrivastava and Li, 2014), for which different algorithms were proposed (Gionis et al., 1999; Malkov and Yashunin, 2020; Guo et al., 2020) to find a solution in a logarithmic time complexity. These algorithms build fixed indexes over the action embeddings β, with particular structures to allow the identification (sometimes approximate) of the K actions with the largest inner product with the query $h_\Xi(x)$. This allows us to reduce even further the complexity of the $\text{argsort}^{\text{K}}$ operator from $\mathcal{O}(P\log K)$ to a logarithmic time complexity $\mathcal{O}(\log P)$, making fast decisions possible in problems with massive action spaces $\mathcal{A}$. This leaves us with the problem of finding the optimal decision function within the constraints of this parametric form. This is achieved by solving the following:

$$\theta^* \in \underset{\theta=[\Xi,\beta]}{\arg\max}\, \mathbb{E}_{x\sim\nu(\mathcal{X})}\left[r(A_K = d_\theta(x), x)\right].$$

As we do not have access to $\nu(\mathcal{X})$, we replace the previous objective with its empirical counterpart:

$$\theta^* \in \underset{\theta=[\Xi,\beta]}{\arg\max}\, \frac{1}{N}\sum_{i=1}^{N} \hat{r}\left(A_K = d_\theta(x_i), x_i\right) \tag{8.3}$$

with $\{x_i\}_{i\in[N]}$ observed contexts and $\hat{r}$ an offline reward estimator; it includes the Direct Method, Inverse Propensity Scoring (Horvitz and Thompson, 1952), Doubly Robust Estimator (Dudík et al., 2014) and many other variants, as presented in (Sakhi et al., 2023c). The optimization problem of Equation (8.3) is complicated by the fact that the reward can be non-smooth and that our decision function is not differentiable. A way to handle this is by relaxing the optimization objective. Differentiable sorting algorithms (Grover et al., 2019; Prillo and Eisenschlos, 2020) address a similar problem but make strong assumptions about the structure of the reward function, and cannot scale to large action space problems. To be as general as possible, we take another direction and relax the problem into an offline policy learning formulation (Bottou et al., 2013; Swaminathan and Joachims, 2015a). We extend our space of parametrized decision functions to a well-chosen space of stochastic policies $\pi_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{S}_K(\mathcal{A}))$, that given a context x, define a probability distribution over the space of slates of size K. Given a policy π_θ, we relax Equation (8.3), taking an additional expectation under π_θ to obtain:

$$\theta^* \in \underset{\theta=[\Xi,\beta]}{\arg\max}\, \hat{R}(\pi_\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{A_K\sim\pi_\theta(\cdot|x_i)}\left[\hat{r}(A_K, x_i)\right]. \tag{8.4}$$

The most common policy class for this type of problem is Plackett-Luce (Plackett, 1975), that generalises the softmax parametrization to slates of size $K > 1$. Under this policy class, computing exact gradients is intractable, but we can obtain approximate gradients w.r.t to θ of Equation (8.4). Common approximations (Williams, 1992) are based on sampling from the policy, are computed in $\mathcal{O}(P\log K)$, and suffer from a variance that grows with the slate size K. In the special case of decomposable rewards over items on the slate (Swaminathan et al., 2017), exploiting this linearity structure (Oosterhuis, 2022) provides gradient estimates with better variance. However, training speed still scales linearly with P making policy learning infeasible in large action spaces.

In this work, we propose **LGP: Latent Gaussian Perturbation**, a new policy class based on smoothing the latent space, that is perfectly suitable to optimize decision functions of the form

	Arbitrary Reward	Low Variance	Time Complexity	Space Complexity
`PL-PG`	✓	✗	$\mathcal{O}(SP)$	$\mathcal{O}(SP)$
`PL-Rank`	✗	✓	$\mathcal{O}(SP)$	$\mathcal{O}(SP)$
`LGP`	✓	✓	$\mathcal{O}(SP)$	$\boldsymbol{\mathcal{O}(SL)}$
`LGP-MIPS`	✓	✓	$\boldsymbol{\mathcal{O}(S\log P)}$	$\boldsymbol{\mathcal{O}(SL)}$

Table 8.1: High level comparison between the different optimization algorithms for slate decision functions, with P the size of the action space, L the size of the embedding space and S the number of samples used to approximate the gradient. `PL-PG` is Plackett-Luce trained with the Score Function Gradient (Williams, 1992). `PL-Rank` is the algorithm proposed in Oosterhuis (2022). `LGP` is our proposed method and `LGP-MIPS` is its accelerated variant. Our method works with arbitrary rewards, scales logarithmically with P and have low memory footprint ($L \ll P$).

described in (8.2). As shown in Table 8.1, our method provides fast sampling, low memory usage, gradient estimates with better computational and statistical properties while being agnostic to the reward structure. When the embeddings are prefixed, this class naturally benefits from approximate MIPS technology, making sampling logarithmic in the action space size and opening the possibility for policy optimization over billion-scale space sizes.

This chapter will be structured as follows. In Section 8.2, we will review the Plackett-Luce policy class and present its limitations. Section 8.3 will introduce our newly proposed relaxation, motivate its use and propose a learning algorithm. We focus in Section 8.4 on experiments to validate our findings empirically. Section 8.5 will cover the related work, and we conclude with Section 8.6.

8.2 Plackett-Luce Policies

8.2.1 A Simple Definition

We relax our objective function and model online decision systems as stochastic policies over the space of slates; ordered lists of actions. There are some natural parametric forms to define a policy on discrete action spaces. If we are dealing with the simple case of $K = 1$ (slate of one action), we can adopt the softmax policy (Swaminathan and Joachims, 2015a; Sakhi et al., 2023c) that, conditioned on the context $x \in \mathcal{X}$ and for a particular action $a \in \mathcal{A}$, is of the form :

$$\pi_\theta(a|x) = \frac{\exp\{f_\theta(a,x)\}}{\sum_b \exp\{f_\theta(b,x)\}} = \frac{\exp\{f_\theta(a,x)\}}{Z_\theta(x)}$$

This softmax parametrization found great success (Swaminathan and Joachims, 2015a; Chen et al., 2019a; Sakhi et al., 2023c), and is ubiquitous in applications where the goal is to learn policies or distributions over discrete actions. Once we deal with $K > 1$, one can generalize the previous form, giving us the Plackett-Luce policy (Plackett, 1975). For a given x and a particular slate $A_K = [a_1, ..., a_K]$, we write its probability:

$$\pi_\theta(A_K|x) = \prod_{i=1}^{K} \frac{\exp\{f_\theta(a_i,x)\}}{Z_\theta^{i-1}(x)} = \prod_{i=1}^{K} \pi_\theta(a_i|x, A_{1:i-1}) \tag{8.5}$$

with $Z_\theta^0(x) = Z_\theta(x)$ and $Z_\theta^i(x) = Z_\theta^{i-1}(x) - \exp\{f_\theta(a_i,x)\}$.

Computing these probabilities can be done in $\mathcal{O}(P)$ which is comparable to the simple case where $K = 1$. The probabilities given by the Plackett-Luce policy are intuitive. Equation (8.5) can be seen as the probability to generate the slate $A_K = [a_1, ..., a_K]$ by sampling without replacement from a categorical distribution over the discrete space $\mathcal{A}$ with action probabilities proportional to $\exp\{f_\theta(a,x)\}$. This sampling procedure can be done in $\mathcal{O}(KP)$, but its sequential nature is a bottleneck for parallelization. Another way to sample from this distribution is to exploit the following expression:

$$\pi_\theta(A_K|x) = \mathbb{E}_{\gamma\sim\mathcal{G}^P(0,1)}\left[\mathbb{1}\left[A_K = \underset{a'\in\mathcal{A}}{\mathrm{argsort}^{\mathrm{K}}}\left\{f_\theta(a',x)+\gamma_i\right\}\right]\right]$$

with $\gamma \sim \mathcal{G}^P(0,1)$ a vector of P independent Gumbel random variables. This is known in the literature as the Gumbel trick (Huijben et al., 2021). This means that sampling from a Plackett-Luce boils down to sampling P independent Gumbel random variables, which costs $\mathcal{O}(P)$, and then computing an $\mathrm{argsort}^{\mathrm{K}}$ of the noised $f(\cdot,x)$ over the discrete action space. We cannot exploit approximate MIPS for this computation as the noise is added after computing the inner product $f(a,x)$, making this step cost $\mathcal{O}(P\log K)$, which makes the total complexity of sampling $\mathcal{O}(P\log K)$, slightly better than the first procedure while compatible with parallel acceleration.

8.2.2 Optimizing The Objective

We want to learn slate policies that can maximize the objective in Equation (8.4). As our objective is decomposable over contexts, stochastic optimization procedures can be adopted Ruder (2016) making optimization over large datasets possible. For this reason, we can focus on the gradient of the objective for a single context x. We derive the score function gradient (Williams, 1992):

$$\nabla_\theta \hat{R}(\pi_\theta|x) = \mathbb{E}_{\pi_\theta(\cdot|x)}\left[\hat{r}(A_K,x)\nabla_\theta \log \pi_\theta(A_K|x)\right] \tag{8.6}$$

$$= \sum_{i=1}^{K} \mathbb{E}_{\pi_\theta(\cdot|x)}\left[\hat{r}(A_K,x)\nabla_\theta \log \pi_\theta(a_i|x,A_{i-1})\right]. \tag{8.7}$$

This gradient is defined as an expectation under $\pi_\theta(\cdot|x)$ over all possible slates. Computing it exactly requires summing $\mathcal{O}(P^K)$ terms which is infeasible. This allows us to approximate the gradient by sampling, which reduces the computation complexity, but we will see that this gradient suffers from further problems.

Computational Burden. The computational complexity of the gradient is crucial for allowing fast learning of slate, as it impacts the running time of every gradient step. Even if we avoid computing the gradient exactly, its approximation can still be a bottleneck when dealing with large action spaces for the following reasons: **(1) Sampling:** Approximating the expectation by sampling slates from $\pi_\theta(\cdot|x)$ can be done in $\mathcal{O}(P\log K)$. However, if the action space is large (P in the order of millions), even a linear complexity on P can be problematic, massively slowing down our optimization procedure. **(2) The Normalizing Constant:** Approximating the gradient needs the computation of $\nabla_\theta \log \pi_\theta(A_K^i|x)$ for the sampled slates $\{A_K^i\}_{i\in[S]}$. This can slow down the optimization procedure as computing the normalizing constant $Z_\theta(x)$ requires summing over all actions, making the complexity of the operation linear in P.

We can solve this computational burden by tackling the two previous problems separately. We can get rid of the normalizing constant in the gradient by generalizing the results of Sakhi et al. (2023c). For a single context x, we can derive a covariance gradient that does not require $Z_\theta(x)$:

$$\nabla_\theta \hat{R}(\pi_\theta|x) = \mathbb{C}\mathrm{ov}_{A_K\sim\pi_\theta(.|x)}\left[\hat{r}(A_K,x), \sum_{i=1}^{K}\nabla_\theta f_\theta(a_i,x)\right] \tag{8.8}$$

with $\mathbb{Cov}[A, \boldsymbol{B}] = \mathbb{E}[(A - \mathbb{E}[A]).(\boldsymbol{B} - \mathbb{E}[\boldsymbol{B}])]$ a covariance between A a scalar function, and $\boldsymbol{B}$ a vector. The proof of this new gradient expression is developed in the Appendix. One can see that for $K = 1$, we recover the results of Sakhi et al. (2023c). This form of gradient does not involve the computation of a normalizing constant, which solves the second problem, but still requires sampling from the policy $\pi_\theta(\cdot|x)$ to get a good covariance estimation. To lower the time complexity of this step, we can use Monte Carlo techniques such as Importance Sampling/Rejection Sampling (Owen, 2013) with carefully chosen proposals to achieve fast sampling without sacrificing the accuracy of the gradient approximation. We develop a discussion around accelerating Plackett-Luce training in the Appendix. While we may have ways to deal with the computation complexity, the Plackett-Luce Policy gradient estimate still suffers from the following problems:

Variance Problems. Let us focus on the gradient derived in Equation (8.6). Its exact computation is intractable, and we need to estimate it by sampling from $\pi_\theta(\cdot|x)$. Let us imagine we sample a slate $A_K = [a_1, ..., a_K]$ to estimate the gradient:

$$\begin{aligned} G_\theta(x) &= \hat{r}(A_K, x)\nabla_\theta \log \pi_\theta(A_K|x) \\ &= \hat{r}(A_K, x)\sum_{i=1}^{K} g_\theta^i(x) \end{aligned}$$

with $g_\theta^i(x)$ set to $\nabla_\theta \log \pi_\theta(a_i|x, A_{i-1})$ to simplify the notation. $G_\theta(x)$ is an unbiased estimator of the gradient $\nabla_\theta \hat{R}(\pi_\theta|x)$ and can be used in a stochastic optimization procedure in a principled manner (Ruder, 2016). However, the efficiency of any stochastic gradient descent algorithm depends on the variance of the gradient estimate (Ajalloeian and Stich, 2020), which is defined for vectors X as:

$$\mathbb{V}[X] = \mathbb{E}\left[||X - \mathbb{E}[X]||^2\right] \in \mathbb{R}^+.$$

Gradients with small variances allow practitioners to use bigger step sizes, which reduces the number of iterations as it makes the whole optimization procedure converge faster. Naturally, we would want the variance of our estimator to be small. Unfortunately, the variance of $G_\theta(x)$ grows with the slate size K. Writing down the variance w.r.t $\pi_\theta(\cdot|x)$ of $G_\theta(x)$:

$$\begin{aligned} \mathbb{V}[G_\theta(x)] &= \mathbb{V}[\hat{r}(A_K, x)\nabla_\theta \log \pi_\theta(A_K|x)] \\ &= \sum_{i=1}^{K} \mathbb{V}[\hat{r}(A_K, x)g_\theta^i(x)] + 2\sum_{i<j} \mathbb{Cov}[\hat{r}(A_K, x)g_\theta^i(x), \hat{r}(A_K, x)g_\theta^j(x)]. \end{aligned}$$

The first term of this variance is a sum over the slate of individual variances, which clearly grows as K grows. For the covariance terms, we argue that, especially when initializing the parameter θ randomly, the gradients $g_\theta^i(x)$ and $g_\theta^j(x)$ will have in expectation different signs making the covariance terms cancel out, leaving the sum of the individual variance terms dominate. This gives a variance that grows in $\mathcal{O}(K)$.

Previous work already showed empirically that the score function gradient for the Plackett-Luce distribution has a large variance (Gadetsky et al., 2020; Buchholz et al., 2022), large enough that learning is not possible in difficult scenarios without considering variance reduction methods (Gadetsky et al., 2020). A possible solution is Randomized Quasi-Monte Carlo (Buchholz et al., 2022; L'Ecuyer, 2016), which produces more accurate estimates by covering the sampling space better. Its value is only significant when we sample few slates $\{A_K^s\}_{s\in[S]}$ to approximate the gradient (Buchholz et al., 2022; L'Ecuyer, 2016). Another direction explores control variates (Gadetsky et al., 2020) as a variance reduction technique. This method requires additional computational costs and a perfect modelling of a differentiable reward proxy to expect variance reduction (Grathwohl et al., 2018).

We want to find a way to both reduce the variance of our method and the computational burden. One of the simplest methods to reduce the variance and gain in computation speed is to reduce the number of parameters we want to train (Koch et al., 2021). In our problem of online decision systems, some parameters can be learned independently, making policy learning easier (Sakhi et al., 2023c).

8.2.3 Fixing The Action Embeddings

As we are dealing with large action spaces, we are constrained to the following structure on the relevance function f_θ for fast querying:

$$f_\theta(a, x) = h_\Xi(x)^\intercal \beta_a, \quad \forall a \in \mathcal{A}.$$

with both $h_\Xi(x)$ and β_a living in an embedding space $\mathbb{R}^L$ with $L \ll P$. The dimension of the matrix β is $[L \times P]$, which is enormous when P is large and dominates Ξ in terms of number of parameters (Koch et al., 2021; Sakhi et al., 2023c). If we can fix the matrix β, it would benefit our approach both in terms of computational efficiency and variance reduction. Indeed, reducing the number of parameters accelerates learning, makes the problem more identifiable, and reduces drastically the gradient variance as:

$$\begin{aligned}\mathbb{V}[G_\theta(x)] &= \mathbb{E}[||\overline{G}_\theta(x)||^2] = \mathbb{E}[||\overline{G}_\Xi(x)||^2] + \mathbb{E}[||\overline{G}_\beta(x)||^2] \\ &= \mathbb{V}[G_\Xi(x)] + \mathbb{V}[G_\beta(x)] \gg \mathbb{V}[G_\Xi(x)].\end{aligned}$$

with $\overline{G}_\theta(x) = G_\theta(x) - \nabla_\theta \hat{R}(\pi_\theta|x)$ the centred gradient estimate. In many applications (think about information retrieval, recommender systems or ad placement), the action embeddings can be learned from the massive data we have on the actions. In these scenarios, actions boil down to web pages, products we want to recommend or place in an ad. We usually have collaborative filtering signal (Sakhi et al., 2020a; Liang et al., 2018) and product descriptions (Vasile et al., 2016) to learn embeddings from. These signals help us obtain good action embeddings β and allow us to fix the action matrix before proceeding to the downstream task we are solving. This approach of fixing β is not new, Koch et al. (2021) fix β to learn a large scale recommender system deployed in production, and Sakhi et al. (2023c) show empirically that learning β actually hurts the performance of softmax policies in large scale scenarios.

We can still learn more about the actions. Even with β fixed, there are sufficient degrees of freedom to solve our downstream task. If we write down:

$$h_\Xi(x) = h_{\Xi'}(x)Z$$

with $\Xi = [\Xi', Z]$, Z being a learnable matrix of size $[L', L]$. the relevance function can be written as:

$$f_\theta(a, x) = h_{\Xi'}(x)^\intercal (Z\beta_a), \quad \forall a \in \mathcal{A}.$$

This means that, even though the matrix β is fixed, we can learn a linear transform of β with the help of Z, injecting information from the downstream task and learning a transformed representation of the actions in the embedding space. In the rest of the chapter, we will fix the action embeddings β making the parametrization of the relevance function f_θ reduce to $\theta = \Xi$, giving for any x:

$$f_\theta(a, x) = h_\theta(x)^\intercal \beta_a, \quad \forall a \in \mathcal{A}.$$

8.3 Latent Random Perturbation

Fixing the embeddings helps to decrease the variance and the number of parameters to optimize substantially, which makes our policy learning routine converge faster. This approach, however, does not deal with the fundamental limit of the Plackett-Luce variance, which grows with the slate size K. This policy class also needs particular care to accelerate its learning; we should adopt the new gradient formula stated in (8.8) combined with advanced Monte Carlo techniques to approximate the gradient efficiently, making the implementation of such methods difficult to achieve. A discussion can be found in the Appendix.

These issues come intrinsically with the adoption of the Plackett-Luce policy suggest we should think differently about how we define policies over slates. Let us look at how the Plackett-Luce policy is defined. For a particular context x and a slate A_K, we write down its Gumbel trick (Huijben et al., 2021) expression:

$$\pi_\theta(A_K|x) = \mathbb{E}_{\gamma\sim\mathcal{G}^P(0,1)}\left[\mathbb{1}\left[A_K = \operatorname*{argsort}^{\mathrm{K}}_{a'\in\mathcal{A}}\left\{h_\theta(x)^\intercal\beta_{a'} + \gamma_i\right\}\right]\right].$$

The Plackett-Luce policy is a smoothed, differentiable relaxation of the following deterministic policy:

$$\begin{aligned} b_\theta(A_K|x) &= \mathbb{1}\left[A_K = \operatorname*{argsort}^{\mathrm{K}}_{a'\in\mathcal{A}}\left\{h_\theta(x)^\intercal\beta_{a'}\right\}\right] \\ &= \mathbb{1}\left[A_K = d_\theta(x)\right]. \end{aligned}$$

b_θ is a deterministic policy putting all its mass on the actions chosen by our decision function d_θ. Note that, taking an expectation under b_θ in Equation (8.4) recovers Equation (8.3). It means that introducing noise relaxed Equation (8.3) to a differentiable objective. This relaxation is achieved by randomly perturbing the scores of the different actions with Gumbel noise (Huijben et al., 2021). As this perturbation is done in the action space level, it induces properties that are hard to deal with: **(1)** The gradient of this policy is an expectation under a potentially large action space, accentuating variance problems. **(2)** The perturbation scales with the size of the action space, as we need P random draws of Gumbel noises. **(3)** Sampling from this policy cannot naturally benefit from approximate MIPS algorithms, as discussed previously.

We observe that the majority of these problems emerge from doing this perturbation in the action space level. With this in mind, we introduce the **LRP: Latent Random Perturbation** policy, that for a context x and a slate A_K, is given by:

$$\pi_\theta^{\mathcal{Q}}(A_K|x) = \mathbb{E}_{\epsilon\sim\mathcal{Q}}\left[\mathbb{1}\left[A_K = \operatorname*{argsort}^{\mathrm{K}}_{a'\in\mathcal{A}}\left\{(h_\theta(x)+\epsilon)^\intercal\beta_{a'}\right\}\right]\right]$$

with $\mathcal{Q}$ a continuous distribution on the latent space $\mathbb{R}^L$. the **LRP** policy defines a smoothed, differentiable relaxation of the deterministic policy b_θ by adding noise in the latent space $\mathbb{R}^L$. This class of policies present desirable properties:

Fast Sampling. For a given x, sampling from a **LRP** policy boils down to sampling from $\mathcal{Q}$ and computing an argsort as:

$$A_K \sim \pi_\theta^{\mathcal{Q}}(\cdot|x) \iff A_K = \operatorname*{argsort}^{\mathrm{K}}_{a'\in\mathcal{A}}\left\{(h_\theta(x)+\epsilon)^\intercal\beta_{a'}\right\}, \epsilon\sim\mathcal{Q}.$$

Let us suppose the sampling from $\mathcal{Q}$ is easy. As the action embeddings β are fixed, and we are performing a perturbation in the latent space, we can set $h_\theta^\epsilon(x) = h_\theta(x) + \epsilon$ which makes sampling compatible with approximate MIPS technology making the sampling achievable in $\mathcal{O}(\log P)$, better than the sampling complexity $\mathcal{O}(P\log K)$ of the Placett-Luce family.

Low variance gradient. Similar to the gradient under the Plackett-Luce policy (8.6), we can derive a score function gradient for **LRP** policies. For a given x, let us write its expected reward under $\pi_\theta^{\mathcal{Q}}$:

$$\begin{aligned}\hat{R}(\pi_\theta^{\mathcal{Q}}|x) &= \mathbb{E}_{A_K \sim \pi_\theta^{\mathcal{Q}}(\cdot|x)}\left[\hat{r}(A_K, x)\right] \\ &= \mathbb{E}_{\epsilon\sim\mathcal{Q}}\left[\hat{r}(A_K\{h_\theta(x)+\epsilon\}, x)\right] \\ &= \mathbb{E}_{h\sim\mathcal{Q}_\theta(x)}[\hat{r}(A_K\{h\}, x)]\end{aligned}$$

$$\begin{gathered} h \sim \mathcal{Q}_\theta(x) \iff h = h_\theta(x) + \epsilon, \quad \epsilon \sim \mathcal{Q} \\ A_K\{h\} = \operatorname*{argsort}^{\mathrm{K}}_{a'\in\mathcal{A}}\left\{h^\intercal \beta_{a'}\right\}. \end{gathered}$$

$\mathcal{Q}_\theta(x)$ is the induced distribution on the user embeddings. $\mathcal{Q}_\theta(x)$ has a tractable density if we choose a classical noise distribution $\mathcal{Q}$ (e.g. Gaussian, Laplace). Working with the induced distribution $\mathcal{Q}_\theta(x)$ transforms the learning parameters θ to parameters of the distribution, allowing us to derive the following gradient:

$$\begin{aligned}\nabla_\theta \hat{R}(\pi_\theta^{\mathcal{Q}}|x) &= \nabla_\theta \mathbb{E}_{h\sim\mathcal{Q}_\theta(x)}[\hat{r}(A_K\{h\}, x)] \\ &= \mathbb{E}_{h\sim\mathcal{Q}_\theta(x)}[\hat{r}(A_K\{h\}, x)\nabla_\theta \log q_\theta(x,h)]\end{aligned}$$

with $\log q_\theta(x,h)$ the log density of $\mathcal{Q}_\theta(x)$ evaluated in h. We can obtain an unbiased gradient estimate by sampling $h \sim \mathcal{Q}_\theta(x)$:

$$G_\theta^{\mathcal{Q}}(x) = \hat{r}(A_K\{h\}, x)\nabla_\theta \log q_\theta(x,h) \tag{8.9}$$

This gradient expression solves all issues that the Plackett-Luce gradient estimate suffered from:

Fast Gradient Estimate. This gradient can be approximated in a sublinear complexity $\mathcal{O}(\log P)$. Building an estimator of the gradient follows these three steps: **(1)** We sample $h \sim \mathcal{Q}_\theta(x)$, which boils down to adding the noise $\epsilon \sim \mathcal{Q}$ to $h_\theta(x)$. This can be done in a complexity $\mathcal{O}(L) \ll \mathcal{O}(P)$ if $\mathcal{Q}$ is chosen properly. **(2)** We evaluate the gradient of the log density $\nabla_\theta \log q_\theta(x,h)$ on h. With a well-chosen $\mathcal{Q}$, this can be done in $\mathcal{O}(L)$ as we do not need to normalize over a large discrete action space. **(3)** We generate the slate $A_K\{h\}$. This boils down to computing an argsort which can be accelerated using approximate MIPS technology, giving a complexity of $\mathcal{O}(\log P)$.

Better Variance. **LRP**'s gradient estimate have better statistical properties for two main reasons: **(1)** The gradient is defined as an expectation under a continuous distribution on the latent space $\mathbb{R}^L$, instead of an expectation under a large discrete action space $\mathcal{A}$ of size $P \gg L$. This can have an impact on the variance of the gradient estimator as the sampling space is smaller. **(2)** The approximate gradient defined in Equation (8.9) does not depend on the slate size K. This results in a variance that does not grow with K. This means that we will notice substantial gains when training policies with larger slates.

LRP policies present themselves as good candidates for learning slate policies, with both computational and statistical benefits compared to the Plackett-Luce family. To derive practical policies, we still need to specify the noise distribution $\mathcal{Q}$ on the latent space. A natural choice is to set $\mathcal{Q}$ to a Centred Independent Gaussian distribution on $\mathbb{R}^L$, with a shared standard deviation σ giving:

$$\epsilon \sim \mathcal{N}(0, \sigma^2 I_L) \iff h = h_\theta(x) + \epsilon \sim \mathcal{N}(h_\theta(x), \sigma^2 I_L).$$

This choice of distribution makes sampling the noise ϵ fast. It also means that the induced distribution on the user embeddings $\mathcal{Q}_\theta(x)$ is a normal distribution with mean parameter $h_\theta(x)$, making the evaluation of its the log density gradient easy for any h:

$$\begin{aligned}\nabla_\theta \log q_\theta(x, h) = -\frac{1}{2\sigma^2}\sum_{i=1}^{L}\nabla_\theta(h^i - h_\theta^i(x))^2 &= \frac{1}{\sigma^2}\sum_{i=1}^{L}(h^i - h_\theta^i(x))\nabla_\theta h_\theta^i(x)\\ &= \frac{1}{\sigma^2}\sum_{i=1}^{L}\epsilon^i\nabla_\theta h_\theta^i(x) = \frac{1}{\sigma}\nabla_\theta(\epsilon_0^\intercal h_\theta(x))\end{aligned}$$

with $\epsilon_0 \sim \mathcal{N}(0, I_L)$. This gradient is a sum over the latent space of size L and does not depend on the size of the action space P nor the slate size K. The expression of this gradient also suggests that the standard deviation σ will play an important role in the optimization process, as the variance of the gradient estimate defined in Equation (8.9) will scale in $\mathcal{O}(1/\sigma^2)$. Even if σ can be treated as an additional parameter, we fix it to $\sigma = 1/L$ in all our experiments for a fair comparison.

The resulting policy, that we name **LGP: Latent Gaussian Perturbation**, will be hyperparameter free, and will show both statistical and computational benefits. With our policy defined, we give a sketch of its optimization procedure in Algorithm 5. This procedure is easy to implement in any automatic differentiation package (Paszke et al., 2019) and is compatible with stochastic first order optimization algorithms (Ruder, 2016). In the next section, we will measure empirically the performance of the different algorithms and show our algorithm efficiency in different scenarios.

Algorithm 5: Learning with Latent Gaussian Perturbation

```
Inputs: $D = \{x_i\}_{i=1}^N$, reward estimator $\hat{r}$, the action embeddings $\beta$
Parameters: $T \geq 1$, Monte Carlo samples number $S \geq 1$
Initialise: $\theta = \theta_0$, MIPS index of $\beta$, $\sigma = 1/L$
for $t = 0$ to $T - 1$ do
    sample a context $x \sim D$
    sample $S$ standard Gaussian noises $\epsilon_1, ..., \epsilon_S \sim \mathcal{N}(0, I_L)$
    compute for $s \in [S]$, $h_s = h_\theta(x) + \epsilon_s$
    compute slates $A_K\{h_s\}$ for $s \in [S]$ with MIPS
    Estimate the gradient:
        $grad_\theta \leftarrow \frac{1}{S\sigma}\sum_{s=1}^{S}\hat{r}(A_K\{h_s\}, x)\nabla_\theta(\epsilon_s^\intercal h_\theta(x))$
    Update the policy parameter $\theta$:
    $\theta \leftarrow \theta - \alpha grad_\theta$
end
return $\theta$
```

8.4 Experiments

8.4.1 Experimental Setting

For our experiments, we focus on learning slate decision functions for the particular case of recommendation as collaborative filtering datasets are easily accessible, facilitating the reproducibility of our results. We choose three collaborative filtering datasets with varying action space size, MovieLens25M (Harper and Konstan, 2015), Twitch (Rappaz et al., 2021) and GoodReads (Wan and McAuley, 2018; Wan et al., 2019). We process these datasets to transform them into user-item interactions of shape $[U, P]$ with U and P the number of users and actions. The statistics of these datasets are described in Table 8.2.

We follow the same procedure as Sakhi et al. (2023c) to build our experimental setup. Given a dataset, we split randomly the user-item interaction session $UI = [X, Y]$ into two parts; the observed interactions X and the hidden interactions Y. the observed part X represents all the information we know about the user, and will be used by our policy π_θ to deliver slates of interest. The hidden part Y is used to define a reward function that will drive the policy to solve a form of session completion task. For a given slate $A_K = [a_1, ..., a_K]$, we define the reward as:

$$\hat{r}(A_K, X) = \sum_{k=1}^{K} \frac{\mathbb{1}[a_k \in Y]}{2^{k-1}}.$$

Although we can adopt an arbitrary form for the reward function, we want rewards that depend on the whole slate (Aouali et al., 2023b) and that take into account the ordering of the items. We choose a linear reward to be able to compare our method to `PL-Rank` (Oosterhuis, 2022), which exploits the reward structure to improve the training of Plackett-Luce policies. The objective we want to optimize is the following:

$$\hat{R}(\pi_\theta) = \frac{1}{U} \sum_{i=1}^{U} \mathbb{E}_{A_K \sim \pi_\theta(\cdot|X_i)} \left[\hat{r}(A_K, X_i)\right].$$

The next step is to parametrize the policy π_θ. For large scale problems, and for a given X, we are restricted to use the following parametrization of the relevance function f_θ:

$$f_\theta(a, X) = h_\theta(X)^\intercal \beta_a, \quad \forall a \in \mathcal{A}.$$

Given the observed interactions X, we compute the action embeddings β using an SVD matrix decomposition (Klema and Laub, 1980). This allows us to project the different action into a latent space of lower dimension $L \ll P$, making β of dimension $[L, P]$. In all experiments, β will be fixed unless we want to study the impact of training the embeddings. When β is fixed, we create an approximate MIPS index using the HNSW algorithm (Malkov and Yashunin, 2020) with the help of the FAISS library (Johnson et al., 2019). This index will accelerate decision-making online as described in Equation (8.2) and can also be exploited to speed up the training of **LGP** policies. With β defined, we still need to parametrize the user embedding function h_θ. Given X, we first define the mean embedding function $M : \mathcal{X} \to \mathbb{R}^L$:

$$M(X) = \frac{1}{|X|} \sum_{a \in X} \beta_a.$$

The function M computes the average of the item embeddings the user interacted with in X (Koch et al., 2021). h_θ follows as:

$$h_\theta(X) = M(X)^\intercal \theta \tag{8.10}$$

with θ a parameter of dimension $[L, L]$, much smaller than $[L, P]$, the dimension of β. All policies in these experiments will use this parametrization. Experiments with deep policies can be found in the Appendix. The training is conducted on a CPU machine, using the Adam optimizer (Kingma and Ba, 2014), with a batch size of 32. We tune the learning rate on a validation set

	#Actions	#Users	Interactions Density
MovieLens 25M	55K	162K	0.24%
Twitch	750K	580K	0.008%
GoodReads	2.23M	400K	0.01%

Table 8.2: The statistics of the datasets after preprocessing

for all algorithms. We adopt Algorithm 5 to train `LGP` and its accelerated variant, `LGP-MIPS`. We denote by `PL-PG`, the algorithm that trains the Plackett-Luce policy trained with the score function gradient; we sample $S \geq 1$ slates $\{A_K^1, ..., A_K^S\}$ from π_θ to derive the gradient estimate for a given X:

$$G_\theta^S(X) = \frac{1}{S}\sum_{i=1}^{S} \hat{r}(A_K^s, X)\nabla_\theta \log \pi_\theta(A_K^s|X). \tag{8.11}$$

We also compare our results to `PL-Rank` (Oosterhuis, 2022) that exploits the linearity of the reward to have a better gradient estimate. As we are mostly interested in the performance of the decision system d_θ, all the rewards reported in the experiments are computed using:

$$\hat{R}(d_\theta) = \frac{1}{U}\sum_{i=1}^{U} \hat{r}(d_\theta(X_i), X_i).$$

In the next section, we study empirically the performance of these approaches by training them with the same time budget on the different datasets. Additional experiments can be found in the Appendix to better understand the behaviour of both the Plackett-Luce policy and the newly introduced **LGP** policy.

8.4.2 Performance

To measure the performance of our algorithms, we use all three datasets with their statistics described in Table 8.2. We fix the latent space dimension $L = 100$ and use a slate size of $K = 5$ for all experiments. We split each dataset by users and keep 10% to create a validation set, on which the reward of the decision function is reported. As these algorithms present different iteration speeds, we fix the same time budget for all training methods for a fair comparison. Training with a time budget also simulates a real production environment, where practitioners are bounded by time constraints and scheduled deployments. For all datasets and training routines, we allow a runtime of 60 minutes, and evaluate our policies on the validation set for 10 equally spaced intervals. The results of these experiments are presented in Figure 8.1 where we plot the evolution of the validation reward on all datasets, for different values of $S \in \{1, 10, 100\}$; the number of samples used to approximate the gradient.

Our first observation from the graph is that `PL-PG` cannot compete with other algorithms, even in the simplest scenario of the MovieLens dataset. Its poor performance is mainly due to the high variance of its gradient estimate. This is confirmed by the performance of `PL-Rank`. Indeed, the `PL-Rank` algorithm works with the same policy class, has the same iteration cost (scales linearly in P) and only differs on the quality of the gradient estimate; exploiting the structure of the reward allow us to obtain an estimate with lower variance. These results confirm our first intuition. `PL-PG` suffer from large variance problems (even in modest sized problems) and is not suitable to solve large scale slate decision problems.

Let us now focus on our newly proposed algorithms; `LGP` and its accelerated variant, `LGP-MIPS`. We observe that in all scenarios considered, the acceleration brought by the approximate MIPS index benefits our algorithm in terms of performance. For the same time budget, `LGP-MIPS`

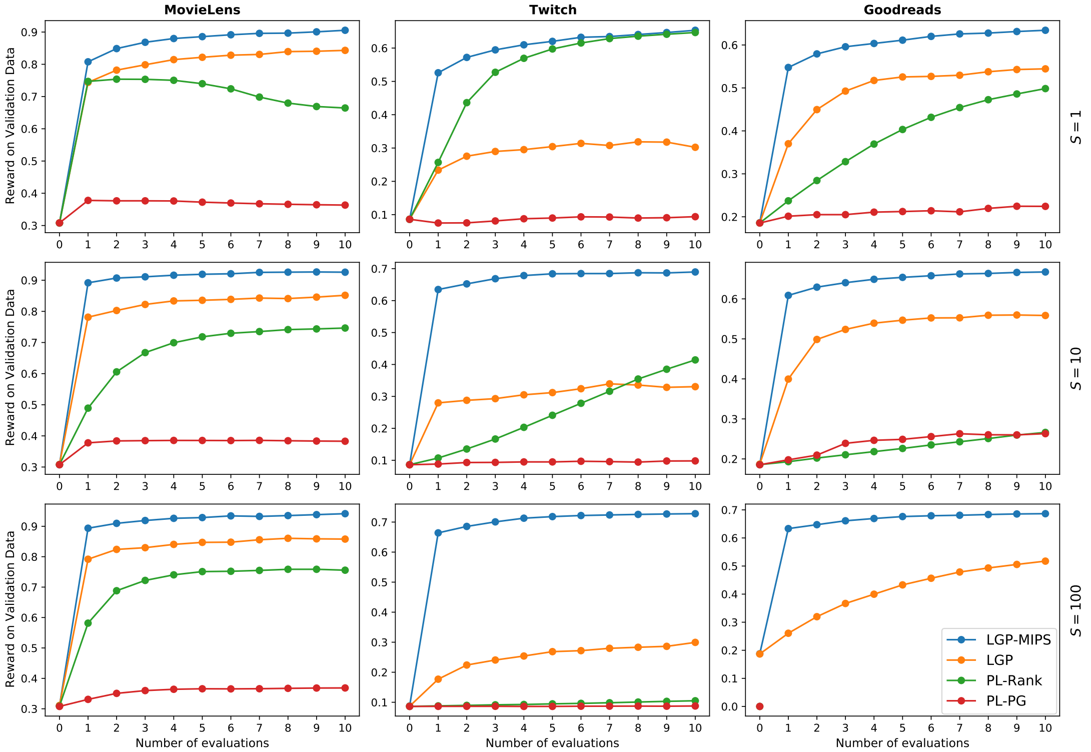


Figure 8.1: The performance of slate decision functions obtained after optimizing them by different training algorithms for the same time budget of 60 minutes. Each evaluation on the validation data is done after 6 minutes of training.

obtains better reward than `LGP`, with the biggest differences observed on datasets with large action spaces; Twitch and GoodReads. `LGP-MIPS` always gives the best performing decision function, for all datasets and number of Monte Carlo samples S considered. These results are promising as our algorithm which is agnostic to the form of the reward outperforms `PL-Rank` that is solely designed to tackle the particular case of linear rewards. This performance is due to `LGP-MIPS`'s superior sampling complexity combined with a low variance gradient estimate. It is also worthy to note that we were unable to run algorithms optimizing Plackett-Luce policies (`PL-PG` and `PL-Rank`) on the GoodReads dataset with $S = 100$ due to its massive memory footprint. As sampling is done on the action space, **Plackett-Luce**-based methods need for each iteration samples of size $\mathcal{O}(SP)$ compared to **LGP**-based method for which the sampling is done in the latent space requiring $\mathcal{O}(SL)$ memory usage. Being able to increase the number of Monte Carlo Samples S is desirable, as it helps reduce the variance of the gradient estimates and accelerates further the training.

These results demonstrate the utility of our newly proposed method over Plackett-Luce for learning slate decision systems. The **LGP** policy class combined with accelerated MIPS indices produces low variance gradient estimates that are fast to compute, can scale to massive action spaces and exhibit low memory usage, making our algorithm the best candidate for optimizing large scale slate decision systems.

8.5 Related work

Learning from interactions. Recent advances in learning large scale decision systems adopt the offline Contextual Bandit/Reinforcement Learning framework (Chen et al., 2022; Wang et al., 2022; Ma et al., 2020; Chen et al., 2019a), proving itself as a powerful paradigm to align business metrics with the offline optimization problem. Research in this direction either explore the use of known policy learning algorithms (Chen et al., 2022) for learning online decision systems or define better rewards to guide this learning (Wang et al., 2022; Christakopoulou et al., 2022). Chen et al. (2019a) showed that large scale recommender systems can benefit from the contextual bandit formulation and introduced a correction to the REINFORCE gradient to encourage softmax policies to recommend multiple items. Their method can be seen as a heuristic applicable when the slate level reward is decomposable into a sum of single item rewards. This assumption is violated in settings where strong interactions exist between the items in the slate. Our method is versatile, does not assume any structure on the reward, and is able to optimize slate decision systems by introducing a new relaxation, smoothing them into a policy that has a better learning behaviour than classical Plackett-Luce.

Scalability. The question of scaling offline policy learning to large scale decision systems has received limited attention. It has been shown that offline softmax policy learning can be scaled to production systems (Chen et al., 2019a). The previous chapter focused on studying the scalability of optimizing policies tailored to one item recommendation, using the covariance gradient estimator combined with importance sampling (Owen, 2013) to exploit approximate MIPS technology in the training phase. The proposed gradient approximation is provably biased, sacrificing the convergence guarantees provided by stochastic gradient descent (Ruder, 2016). Our method extends the scope of the previous chapter, as it can be applied to slate recommendation. It provides a simpler relaxation than Plackett-Luce, producing a learning algorithm that benefits from approximate MIPS technology, naturally obtaining unbiased gradient estimates with better statistical properties.

Learning to Rank. The learning to rank literature separates algorithms by the output space they operate on, making a clear distinction between pointwise, pairwise and listwise approaches (Liu, 2009) . Our method falls in the latter (Xia et al., 2008), as we operate with policies on the slate level. The majority of work in LTR trains decision systems through the optimization of ranking losses (Wang et al., 2018; Oosterhuis, 2022), defined as differentiable surrogates of ranking metrics or by an adapted *maximum likelihood estimation* (Rendle et al., 2009; Ma et al., 2021). In the same direction, differentiable sorting algorithms (Grover et al., 2019; Prillo and Eisenschlos, 2020) aim at producing a differentiable relaxation to sorting functions that handle reward on the item level These methods also require linear rewards in addition to training an item-item interaction matrix, *quadratic* on the action space size, making them unsuitable to massive action spaces. We are interested in training reward-driven, large scale slate decision systems, to learn rankings that are more aligned with arbitrary, complex rewards functions.

Smoothing non-differentiable objectives. Our procedure can be interpreted as a stochastic relaxation of non-differentiable decision functions. This relaxation is achieved by introducing a well-chosen noise in the latent space. PAC-Bayesian policy learning (London and Sandler, 2019; Sakhi et al., 2023a; Aouali et al., 2023a) and Black-Box optimization algorithms (Bajaj et al., 2021; Beyer and Schwefel, 2002; Staines and Barber, 2012) adopt a similar paradigm to optimize non-smooth loss functions. They proceed by injecting noise in the parameter space and derive a gradient with respect to the noise distribution. This parameter level perturbation can suffer from computational issues when the number of parameters increases. Our method is agnostic to the number of parameters. By perturbing the latent space directly, we bypass this problem and simplify the sampling procedure, resulting in faster and more efficient training.

8.6 Conclusion

Countless large scale online decision systems are tasked with delivering *slates* based on contextual information. We formulate the learning of these systems in the offline contextual bandit setting. This framework relaxes decision systems to stochastic policies, and proceeds at learning them through REINFORCE-like algorithms. Plackett-Luce provides an interpretable distribution over ordered lists of items, but its use in a large scale policy learning context is far from being optimal. In this chapter, we motivate the **Latent Gaussian Perturbation**, a new policy class over ordered lists defined as a stochastic, differentiable relaxation of the argsort decision function, induced by perturbing the latent space. The **LGP** policy provides gradient estimates with better computational and statistical properties. We built an intuition on why this new policy class is better behaved and demonstrated through extensive experiments that not only **LGP** is faster to train, considerably reducing the computational cost, but it also produces policies with better performance, making the use of Plackett-Luce policies in this context obsolete. This work gives practitioners a new way to address large scale slate policy learning and aim at contributing into the adoption of REINFORCE decision systems. The results obtained in this chapter suggest that the relaxations used to define policies have a big role in optimization, and that we might need to take a step backward and reconsider the massively adopted Softmax/Plackett-Luce parametrization. As we only focused on simple noise distributions, a nice avenue of research will be to study the impact of the choice of these distributions on the learning algorithm produced.

8.7 Appendix

8.7.1 Accelerating Plackett-Luce training.

As it was discussed previously, we can derive a covariance gradient that does not require the computation of the normalizing constant $Z_\theta(x)$:

$$\nabla_\theta \hat{R}(\pi_\theta|x) = \mathbb{Cov}_{A_K \sim \pi_\theta(.|x)} \left[\hat{r}(A_K, x), \sum_{i=1}^{K} \nabla_\theta f_\theta(a_i, x) \right]$$

with $\mathbb{Cov}[A, \boldsymbol{B}] = \mathbb{E}[(A - \mathbb{E}[A]).(\boldsymbol{B} - \mathbb{E}[\boldsymbol{B}])]$ a covariance between A a scalar function, and $\boldsymbol{B}$ a vector. The proof follows:

$$\begin{aligned}
\nabla_\theta \hat{R}(\pi_\theta|x) &= \mathbb{E}_{\pi_\theta(\cdot|x)}[\hat{r}(A_K, x) \nabla_\theta \log \pi_\theta(A_K|x)] \\
&= \sum_{i=1}^{K} \mathbb{E}_{\pi_\theta(\cdot|x)}[\hat{r}(A_K, x) \nabla_\theta \log \pi_\theta(a_i|x, A_{i-1})] \\
&= \sum_{i=1}^{K} \mathbb{E}_{\pi_\theta(\cdot|x)} \left[\hat{r}(A_K, x) \left(\nabla_\theta f_\theta(a_i, x) - \nabla_\theta \log Z_\theta^{i-1}(x) \right) \right]
\end{aligned}$$

Using the log trick, we derive the following equality:

$$\nabla_\theta \log Z_\theta^{i-1}(x) = \mathbb{E}_{\pi_\theta(a_i|x, A_{i-1})}[\nabla_\theta f_\theta(a, x)].$$

This equality is then injected in the gradient formula derived above to obtain:

$$\nabla_\theta \hat{R}(\pi_\theta|x) = \mathbb{Cov}_{A_K \sim \pi_\theta(.|x)} \left[\hat{r}(A_K, x), \sum_{i=1}^{K} \nabla_\theta f_\theta(a_i, x) \right].$$

This concludes the proof. One can see that for $K = 1$, we recover the results of Sakhi et al. (2023c). This form of gradient still requires sampling from the policy $\pi_\theta(\cdot|x)$ to get a good covariance estimation. To lower the time complexity of this step, we can use Monte Carlo techniques such as Importance Sampling/Rejection Sampling (Owen, 2013) with carefully chosen proposals to achieve fast sampling without sacrificing the accuracy of the gradient approximation.

Algorithm 6: Categorical Distribution: Rejection sampling using MIPS

```
Input: $h_\Xi$, $\beta$, $x$, and indexes on $\beta$ and parameter $K$, catalogue size $\mathcal{P}$
Output: $a$ which is a sample from $P(A = a) = \frac{\exp(h_\Xi(x)^\intercal \beta_a)}{\sum_{a'} \exp(h_\Xi(x)^\intercal \beta_{a'})}$
$\alpha_1, ..., \alpha_K = \mathrm{argsort}(h_\Xi(x)^\intercal \beta)_{1:K}$
$\mathcal{Z}' = \mathcal{P} \exp(h_\Xi(x)^\intercal \beta_{\alpha_K})$
$\mathcal{Z}'' = \sum_{a'}^{K} \exp(h_\Xi(x)^\intercal \beta_{a'}) - \exp(h_\Xi(x)^\intercal \beta_{\alpha_K})$
$P_K = [\exp(h_\Xi(x)^\intercal \beta_{\alpha_1})/\mathcal{Z}'', ..., \exp(h_\Xi(x)^\intercal \beta_{\alpha_K})/\mathcal{Z}'']$
while True do
    $d \sim \mathrm{cat}([\mathrm{tail}, \mathrm{head}], [\frac{\mathcal{Z}'}{\mathcal{Z}' + \mathcal{Z}''}, \frac{\mathcal{Z}''}{\mathcal{Z}' + \mathcal{Z}''}])$
    if $d$=head then
        $r \sim \mathrm{cat}(\alpha_1, ..., \alpha_K, P_K)$
        return $r$
    end
    if $d$=tail then
        Sample $q$ uniformly from the set $\{1, ..., P\}$
        Sample $u$ from a uniform distribution
        if $\frac{\exp(h_\Xi(x)^\intercal \beta_q)}{\exp(h_\Xi(x)^\intercal \beta_{\alpha_K})} > u$ then
            return $q$
        end
    end
end
```

In Sakhi et al. (2023c), a softmax policy learning algorithm was accelerated by approximating the gradients using a self normalized importance sampling algorithm with a proposal distribution that can both exploit the MIPS structure and is a good approximation of the target softmax distribution. This idea can also be used to motivate a rejection sampling algorithm, as a similar proposal can be shown to form an envelope of the target density. It can also be extended from softmax to Plackett-Luce. When the idea of rejection sampling is combined with the MIPS proposal, it results in the rejection sampling algorithm shown in Algorithm 6. While Algorithm 6 can be extended to the slate policy case, enabling the fast evaluation of the covariance gradient estimator, it will still suffer from high variance gradient estimates.

8.7.2 Additional Experiments

Effect of fixing β

In this section, we want to validate the intuition we built throughout the chapter about the behaviours of both the Plackett-Luce and **LGP** policy classes. We focus on MovieLens (Harper and Konstan, 2015), a medium scale dataset that will allow us to test all our methods, regardless of their potential to scale to harder problems. For all experiments in this section, we fix $L = 100 \ll P$ and we study the impact of fixing the action embeddings β. As discussed in Section 3, having the embeddings fixed is a natural solution to improve the learning of these large scale decision systems, as it can reduce both the variance of the gradient estimates and the running

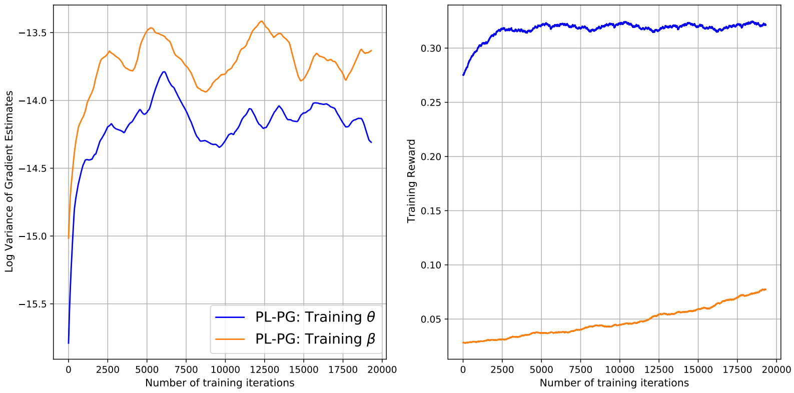


Figure 8.2: Experiments on the MovieLens dataset: We look at the effect of fixing the action embeddings β on the training of Plackett-Luce policies. Training β results in a slow optimization procedure and gradient estimates with bigger variance.

time of the optimization procedure. To validate this, we focus on the Plackett-Luce policy and define two slightly different parametrizations:

- Learn θ: this is the parametrization introduced in Equation (8.10), with β fixed and we only learn the parameter θ.
- Learn β: We treat β as a parameter after initializing it with the SVD values. Because our user embedding function h_θ is linear in θ, we get rid of this parameter as it becomes redundant once β can be optimized. This gives the following parametrization:

$$\forall (x,a) \in \mathcal{X} \times \mathcal{A}, \quad f_\beta(a,x) = M(X)^\intercal \beta_a$$

We train a Plackett-Luce policy with both parametrizations for one epoch, while fixing the slate size $K = 2$ and the Monte Carlo samples to $S = 1$. In this experiment, our objective is not to produce the best policies, but to understand how fixing the embeddings can impact the optimization procedure. We report in Figure 8.2 the evolution of both the gradient estimate variance and the reward on the training data for both parametrizations. We can observe that treating β as a parameter to optimize, even when initialized properly, leads to slow learning. The variance of the gradient estimate when learning β is bigger than the variance when learning θ with β fixed, and this will get bigger for problems with larger action spaces as $P \gg L$. We suspect that this is one of the reasons that explain the pace of learning when optimizing β. The same phenomenon was observed in Sakhi et al. (2023c). The same experiments demonstrated that training θ alone was **twice** as fast as training β in this experiment. This suggests that fixing β to a good value is beneficial for training large scale decision systems, both in terms of iteration efficiency. We advocate for fixing the action embeddings β when learning large scale MIPS systems.

Impact of the slate size K.

One of the caveats of the Plackett-Luce slate policy is that its gradient estimate has a variance that grows with the slate size K, reducing its scope of applications to modest slate sizes. the gradient estimate of LGP however does not suffer from this issue, and we want to showcase that with a simple experiment. We derived gradient estimates for **LGP**-based methods that have a variance that scales in $\mathcal{O}(1/\sigma^2)$. Although the standard deviation σ can be treated as a hyperparameter depending on the task, To allow a fair comparison of the gradient variance

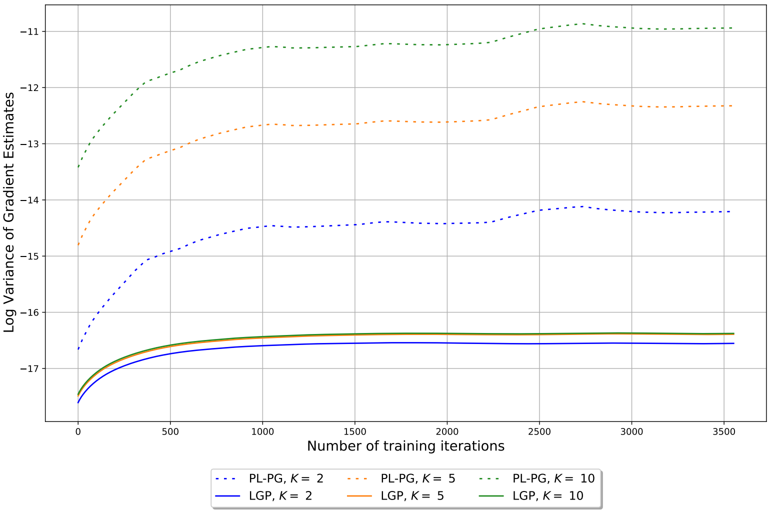


Figure 8.3: Impact of the slate size K on the log variance of the gradient estimate of both `PL-PG` and `LGP` on MovieLens. Contrary to `LGP`, `PL-PG` has a gradient estimate with a variance that grows with K.

of these methods, we set σ to a particular value coming from the following observation. For a particular action a:

$$\begin{aligned} h \sim \mathcal{N}(h_\theta(x), \sigma^2 I_L) &\implies h^\intercal \beta_a \sim \mathcal{N}(h_\theta(x)^\intercal \beta_a, \sigma^2 ||\beta_a||^2) \\ &\implies h^\intercal \beta_a = h_\theta(x)^\intercal \beta_a + \epsilon_a \end{aligned}$$

with $\epsilon_a \sim \mathcal{N}(0, (\sigma||\beta_a||)^2)$. This can be interpreted as adding a scaled Gaussian noise to the score of action a. As we add standardized Gumbel noise $\gamma_a \sim \mathcal{G}(0,1)$ to the action scores to define the Plackett-Luce policy, we want, for a fair comparison, to have:

$$\forall a \in \mathcal{A}, \quad \sigma||\beta_a|| \approx 1$$

One heuristic to approximately achieve that is to compute the empirical mean of the β norms $B = \frac{1}{P}\sum_{a\in\mathcal{A}} ||\beta_a||$ and set $\sigma = 1/B$. This value will be used for this experiment.

We use the MovieLens dataset, and train the `LGP` policy *without exploiting the MIPS index on* β, and Plackett-Luce with the parametrization of Equation (8.10) for 10 epochs with $S = 1$, while varying the slate size $K \in \{2, 5, 10\}$. Note that all policies are initialized with the same random seed for a fair comparison. We report the evolution of the variance of the gradient estimate alongside the reward on the training data. The results of these experiments are presented in Figure 8.3.

Focusing on the evolution of the variance, we can see that Plackett-Luce does indeed have a variance that grows with K contrary to **LGP** that has a variance of its gradient estimate staying at the same scale no matter the value of K. We argue that this has a direct impact on the optimization procedure as for the same value of K, we observe in Figure 8.1 that **LGP**-based methods outperform Plackett-Luce learning schemes consistently, making it a good candidate for learning slate policies.

Experiments with Neural Networks.

For these experiments, we want to explore deep policies and see if we can still empirically validate our findings in this case as well. For this, we adopt the following function to compute the user

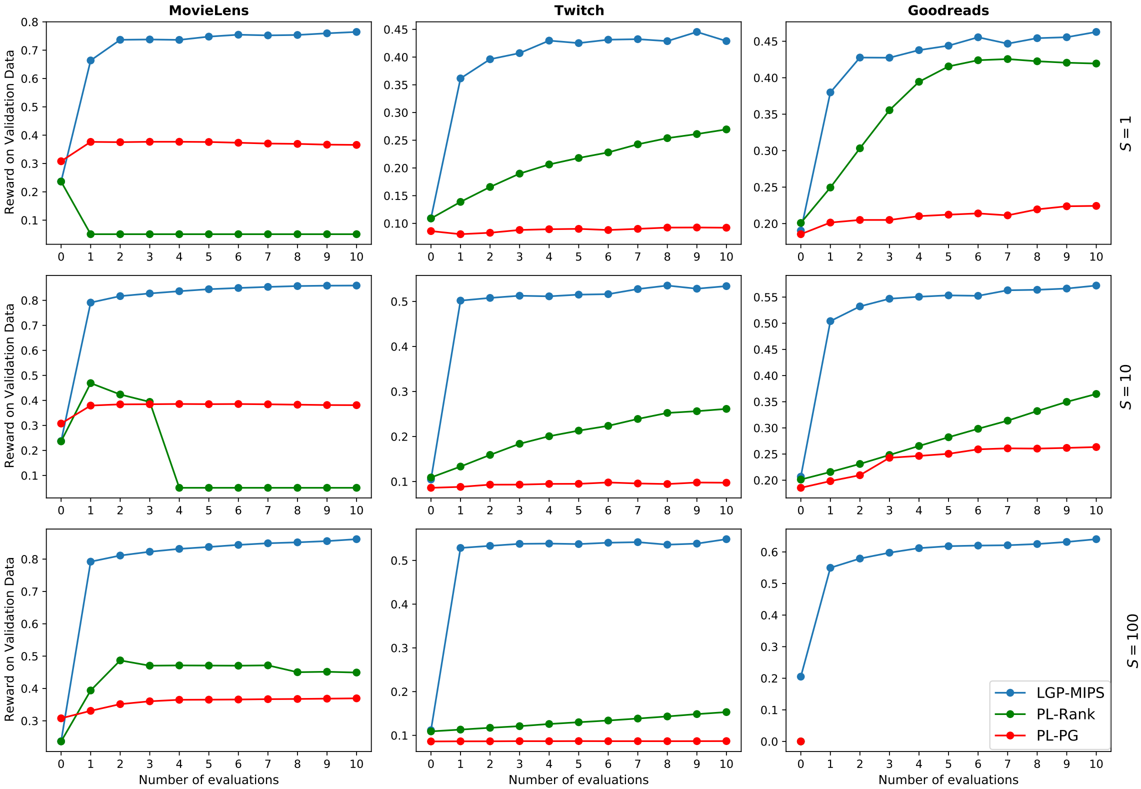


Figure 8.4: The performance of slate decision functions, with Neural Network Backbones, obtained by our different training algorithms after running the optimization for 60 minutes.

embedding h_{θ_1,θ_2}:

$$h_{\theta_1,\theta_2}(X) = \texttt{sigmoid}\left(M(X)^\intercal\theta_1\right)\theta_2, \tag{8.12}$$

which boils down to a sigmoid, two layer feed forward neural network with both θ_1 and θ_2 of size $[L, L]$. We run `PL-PG`, `PL-Rank` and `LGP-MIPS` on the three datasets, for the same running time (60 minutes) and cross-validate the learning rate choosing the best value for each algorithm. We aggregate the results on Figure 8.4. The plot suggests that even for deep policies, `LGP-MIPS` outperforms **Plackett-Luce**-based methods on all datasets and for different values of the number of Monte Carlo samples S. We also observe that training in this case is more unstable, especially for **Plackett-Luce**-based methods, as having more parameters accentuate the variance problems of their gradient estimates. It is noteworthy that, the reward obtained with deep policies in our experiment is less than the one achieved by linear policies. This suggests that deep policies require additional care when training, and we might want to stick to simple policies if we are interested in fast and reliable optimization.

Bibliography

Marc Abeille and Alessandro Lazaric. Linear Thompson Sampling Revisited. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 176–184. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/abeille17a.html.

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

M. Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Comput. Surv.*, 55(7), dec 2022. ISSN 0360-0300. doi: 10.1145/3543846. URL https://doi.org/10.1145/3543846.

Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pages 405–431, 2017.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 127–135, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/agrawal13.html.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 151–160. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ahmed19a.html.

Ahmad Ajalloeian and Sebastian U. Stich. On the convergence of sgd with biased gradients, 2020. URL https://api.semanticscholar.org/CorpusID:234358812.

Aymen Al-Marjani, Andrea Tirinzoni, and Emilie Kaufmann. Active coverage for pac reinforcement learning. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5044–5109. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/al-marjani23a.html.

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. *ArXiv*, abs/2110.11216, 2021. URL https://api.semanticscholar.org/CorpusID:239049660.

Pierre Alquier and Benjamin Guedj. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning*, 107(5):887–902, 2018. doi: 10.1007/s10994-017-5690-0. URL https://doi.org/10.1007/s10994-017-5690-0.

Imad Aouali, Amine Benhalloum, Martin Bompaire, Achraf Ait Sidi Hammou, Sergey Ivanov, Benjamin Heymann, David Rohde, Otmane Sakhi, Flavian Vasile, and Maxime Vono. Reward optimizing recommendation using deep learning and fast maximum inner product search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4772–4773, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3542622. URL https://doi.org/10.1145/3534678.3542622.

Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. Exponential smoothing for off-policy learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 984–1017. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/aouali23a.html.

Imad Aouali, Achraf Ait Sidi Hammou, Sergey Ivanov, Otmane Sakhi, David Rohde, and Flavian Vasile. Probabilistic Rank and Reward: A Scalable Model for Slate Recommendation. working paper or preprint, January 2023b. URL https://hal.science/hal-03959643.

Imad Aouali, Branislav Kveton, and Sumeet Katariya. Mixed-effect thompson sampling. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 2087–2115. PMLR, 25–27 Apr 2023c. URL https://proceedings.mlr.press/v206/aouali23a.html.

Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4(none):40 – 79, 2010. doi: 10.1214/09-SS054. URL https://doi.org/10.1214/09-SS054.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Ishan Bajaj, Akhil Arora, and M. M. Faruque Hasan. *Black-Box Optimization: Methods and Applications*, pages 35–65. Springer International Publishing, Cham, 2021. ISBN 978-3-030-66515-9. doi: 10.1007/978-3-030-66515-9_2. URL https://doi.org/10.1007/978-3-030-66515-9_2.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3(null):463–482, mar 2003. ISSN 1532-4435.

Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38, dec 1992. ISSN 0001-0782. doi: 10.1145/138859.138861. URL https://doi.org/10.1145/138859.138861.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Yoshua Bengio and Jean-Sébastien Senecal. Quick training of probabilistic neural nets by importance sampling. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 17–24. PMLR, 03–06 Jan 2003. URL https://proceedings.mlr.press/r4/bengio03a.html. Reissued by PMLR on 01 April 2021.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: a review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798—1828, August 2013. ISSN 0162-8828. doi: 10.1109/tpami.2013.50. URL http://arxiv.org/pdf/1206.5538.

James O Berger, Robert L Wolpert, MJ Bayarri, MH DeGroot, Bruce M Hill, David A Lane, and Lucien LeCam. The likelihood principle. *Lecture notes-Monograph series*, 6:iii–199, 1988.

H.G. Beyer and HP. Schwefel. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1(1):3–52, March 2002. doi: 10.1023/A:1015059928466.

Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 129–138. ACM, 2009.

Guy Blanc and Steffen Rendle. Adaptive sampled softmax with kernel based sampling. In *International Conference on Machine Learning*, pages 590–599. PMLR, 2018.

David M. Blei and John D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 147–154, 2005. URL http://papers.nips.cc/paper/2906-correlated-topic-models.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003. ISSN 1532-4435.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, apr 2017. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080%2F01621459.2017.1285773.

Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. Casmos: A framework for learning candidate selection models over structured queries and documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 441–450, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939718. URL https://doi.org/10.1145/2939672.2939718.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(65):3207–3260, 2013. URL http://jmlr.org/papers/v14/bottou13a.html.

Guillaume Bouchard. Efficient bounds for the softmax function, applications to inference in hybrid models, 2007.

Nicolas Boulle and Alex Townsend. A generalization of the randomized singular value decomposition. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hgKtwSb4S2.

David Brandfonbrener, William Whitney, Rajesh Ranganath, and Joan Bruna. Offline contextual bandits with overparameterized models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1049–1058. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/brandfonbrener21a.html.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A%3A1010933404324.

Alexander Buchholz, Jan Malte Lichtenberg, Giuseppe Di Benedetto, Yannik Stein, Vito Bellini, and Matteo Ruffini. Low-variance estimation in the plackett-luce model via quasi-monte carlo sampling, 2022. URL https://arxiv.org/abs/2205.06024.

Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3): 593–613, 2009.

Olivier Catoni. PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *IMS Lecture Notes Monograph Series*, page 1–163, 2007. ISSN 0749-2170. doi: 10.1214/074921707000000391. URL http://dx.doi.org/10.1214/074921707000000391.

Chong Chen, Weizhi Ma, Min Zhang, Chenyang Wang, Yiqun Liu, and Shaoping Ma. Revisiting negative sampling vs. non-sampling in implicit recommendation. *ACM Trans. Inf. Syst.*, 41(1), feb 2023. ISSN 1046-8188. doi: 10.1145/3522672. URL https://doi.org/10.1145/3522672.

Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3470–3477, Apr. 2020. doi: 10.1609/aaai.v34i04.5751. URL https://ojs.aaai.org/index.php/AAAI/article/view/5751.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 456–464, New York, NY, USA, 2019a. Association for Computing Machinery. ISBN 9781450359405. doi: 10.1145/3289600.3290999. URL https://doi.org/10.1145/3289600.3290999.

Minmin Chen, Ramki Gummadi, Chris Harris, and Dale Schuurmans. Surrogate objectives for batch policy optimization in one-step decision making. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/84899ae725ba49884f4c85c086f1b340-Paper.pdf.

Minmin Chen, Can Xu, Vince Gatto, Devanshu Jain, Aviral Kumar, and Ed Chi. Off-Policy Actor-Critic for Recommender Systems. In *Proceedings of the 16th ACM Conference on*

Recommender Systems, RecSys '22, page 338–349, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392785. doi: 10.1145/3523227.3546758. URL https://doi.org/10.1145/3523227.3546758.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014. URL http://aclweb.org/anthology/D/D14/D14-1179.pdf.

Nicolas Chopin and Omiros Papaspiliopoulos. *An introduction to Sequential Monte Carlo / Nicolas Chopin, Omiros Papaspiliopoulos.* Springer Series in Statistics. Springer, Cham, Switzerland, 1st ed. 2020. edition, 2020. ISBN 3-030-47845-9.

Konstantina Christakopoulou, Can Xu, Sai Zhang, Sriraj Badam, Trevor Potter, Daniel Li, Hao Wan, Xinyang Yi, Ya Le, Chris Berg, Eric Bencomo Dixon, Ed H. Chi, and Minmin Chen. Reward shaping for user satisfaction in a reinforce recommender, 2022. URL https://arxiv.org/abs/2209.15166.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 208–214, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/chu11a.html.

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584805. doi: 10.1145/1646396.1646452. URL https://doi.org/10.1145/1646396.1646452.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

Patrick L Combettes. Perspective Functions: Properties, Constructions, and Examples. *Set-Valued and Variational Analysis*, 26(2):247–264, 2018.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Bo Dai, Ofir Nachum, Yinlam Chow, Lihong Li, Csaba Szepesvari, and Dale Schuurmans. Coindice: Off-policy confidence interval estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9398–9411. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6aaba9a124857622930ca4e50f5afed2-Paper.pdf.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296, 2010.

Rémy Degenne, Thomas Nedelec, Clément Calauzènes, and Vianney Perchet. Bridging the gap between regret minimization and best arm identification, with application to a/b tests. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1988–1996. PMLR, 2019.

John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019. URL http://jmlr.org/papers/v20/17-750.html.

John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3): 946–969, 2021.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In Jyrki Kivinen and Robert H. Sloan, editors, *Computational Learning Theory*, pages 255–270, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45435-9.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456. PMLR, 2018.

Louis Faury, Ugo Tanielian, Flavian Vasile, Elena Smirnova, and Elvis Dohmatob. Distributionally robust counterfactual risk minimization. In *AAAI*, 2020.

Artyom Gadetsky, Kirill Struminsky, Christopher Robinson, Novi Quadrianto, and Dmitry Vetrov. Low-variance black-box gradient estimates for the plackett-luce distribution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10126–10135, Apr. 2020. doi: 10.1609/aaai.v34i06.6572. URL https://ojs.aaai.org/index.php/AAAI/article/view/6572.

F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. Offline and Online Evaluation of News Recommender Systems at Swissinfo.Ch. In *Proc. of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 169–176, 2014.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL https://proceedings.mlr.press/v49/garivier16a.html.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian Learning of Linear Classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 353–360, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553419. URL https://doi.org/10.1145/1553374.1553419.

Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, page 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606157.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, dec 1992. ISSN 0001-0782. doi: 10.1145/138859.138867. URL https://doi.org/10.1145/138859.138867.

Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4), dec 2016. ISSN 2158-656X. doi: 10.1145/2843948. URL https://doi.org/10.1145/2843948.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyzKd1bCW.

Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1eSS3CcKX.

Benjamin Guedj. A Primer on PAC-Bayesian Learning, 2019.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1908.10396.

Somit Gupta, Ronny Kohavi, Diane Tang, Ya Xu, Reid Andersen, Eytan Bakshy, Niall Cardin, Sumita Chandran, Nanyu Chen, Dominic Coey, Mike Curtis, Alex Deng, Weitao Duan, Peter Forbes, Brian Frasca, Tommy Guy, Guido W. Imbens, Guillaume Saint Jacques, Pranav Kantawala, Ilya Katsev, Moshe Katzwer, Mikael Konutgan, Elena Kunakova, Minyong Lee, MJ Lee, Joseph Liu, James McQueen, Amir Najmi, Brent Smith, Vivek Trehan, Lukas Vermeer, Toby Walker, Jeffrey Wong, and Igor Yashkov. Top challenges from the first practical online controlled experiments summit. *SIGKDD Explor. Newsl.*, 21(1):20–35, may 2019. ISSN 1931-0145. doi: 10.1145/3331651.3331655. URL https://doi.org/10.1145/3331651.3331655.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Y.W. Teh and M. Titterington, editors, *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *JMLR W&CP*, pages 297–304, 2010.

Maxime Haddouche and Benjamin Guedj. Online PAC-bayes learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=4pwCvvel8or.

Maxime Haddouche and Benjamin Guedj. PAC-bayes generalisation bounds for heavy-tailed losses through supermartingales. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=qxrwt6F3sf.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *J. Mach. Learn. Res.*, 16(1):3367–3402, jan 2015. ISSN 1532-4435.

Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 549–558, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340694. doi: 10.1145/2911451.2911489. URL https://doi.org/10.1145/2911451.2911489.

David A. Hensher and William H. Greene. The mixed logit model: The state of practice. *Transportation*, 30(2):133–176, 2003. doi: 10.1023/A:1022558715350. URL https://doi.org/10.1023/A:1022558715350.

Miguel A Hernan and James M Robins. Causal inference, 2010.

Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang, editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 843–852. ACM, 2018. ISBN 978-1-4503-6014-2. doi: 10.1145/3269206.3271761. URL https://doi.org/10.1145/3269206.3271761.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952. ISSN 01621459. URL http://www.jstor.org/stable/2280784.

Ya-Ping Hsieh, Panayotis Mertikopoulos, and Volkan Cevher. The limits of min-max optimization algorithms: Convergence to spurious non-critical sets. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4337–4348. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/hsieh21a.html.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2333–2338, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322638. doi: 10.1145/2505515.2505665. URL https://doi.org/10.1145/2505515.2505665.

Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG van Sloun. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *arXiv preprint arXiv:2110.01515*, 2021.

Edward L Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008. doi: 10.1198/106186008X320456. URL https://doi.org/10.1198/106186008X320456.

Tommi Jaakkola and Michael Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, page 4, 1997.

Darko Janeković and Dario Bojanjac. Randomized algorithms for singular value decomposition: Implementation and application perspective. In *2021 International Symposium ELMAR*, pages 165–168, 2021. doi: 10.1109/ELMAR52657.2021.9550979.

Kyoungseok Jang, Kwang-Sung Jun, Ilja Kuzborskij, and Francesco Orabona. Tighter PAC-Bayes Bounds Through Coin-Betting. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 2240–2264. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/jang23a.html.

Dietmar Jannach and Michael Jugovac. Measuring the business value of recommender systems. *ACM Trans. Manage. Inf. Syst.*, 10(4), dec 2019. ISSN 2158-656X. doi: 10.1145/3370082. URL https://doi.org/10.1145/3370082.

Olivier Jeunen and Bart Goethals. *Pessimistic Reward Models for Off-Policy Learning in Recommendation*, page 63–74. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450384582. URL https://doi.org/10.1145/3460231.3474247.

Olivier Jeunen, Jan Van Balen, and Bart Goethals. Closed-form models for collaborative filtering with side-information. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 651–656, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3418480. URL https://doi.org/10.1145/3383313.3418480.

Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJaP_-xAb.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, pages 3020–3029, 2016.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *AISTATS*, 2018.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 227–234, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1238–1246, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/karnin13.html.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL https://openreview.net/group?id=ICLR.cc/2014.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980. doi: 10.1109/TAC.1980.1102314.

David A. Knowles and Tom Minka. Non-conjugate variational message passing for multinomial and binary regression. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1701–1709. Curran Associates, Inc., 2011.

Olivier Koch, Amine Benhalloum, Guillaume Genthial, Denis Kuzin, and Dmitry Parfenchik. Scalable representation learning and retrieval for display advertising. *arXiv preprint arXiv:2101.00870*, 2021.

Ron Kohavi, Alex Deng, Brian Frasca, Roger Longbotham, Toby Walker, and Ya Xu. Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 786–794, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450314626. doi: 10.1145/2339530.2339653. URL https://doi.org/10.1145/2339530.2339653.

Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 77–118. Springer, 2015.

Ilja Kuzborskij and Csaba Szepesvári. Efron-Stein PAC-Bayesian Inequalities, 2020.

Ilja Kuzborskij, Claire Vernade, Andras Gyorgy, and Csaba Szepesvari. Confident off-policy evaluation and selection through self-normalized importance weighting. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 640–648. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/kuzborskij21a.html.

John D. Lafferty and David M. Blei. Correlated topic models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 147–154. MIT Press, 2006. URL http://papers.nips.cc/paper/2906-correlated-topic-models.pdf.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, page 28, 2018.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. doi: 10.1017/9781108571401.

Pierre L'Ecuyer. Randomized Quasi-Monte Carlo: An Introduction for Practitioners. In *12th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC 2016)*, Stanford, United States, August 2016. URL https://hal.inria.fr/hal-01561550.

Gaël Letarte, Pascal Germain, Benjamin Guedj, and Francois Laviolette. Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/7ec3b3cf674f4f1d23e9d30c89426cce-Paper.pdf.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World Wide Web*, pages 661–670, 2010.

Xiangyang Li, Bo Chen, Huifeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, Jinxing Liu, Zhenhua Dong, and Ruiming Tang. Inttower: The next generation of two-tower model for pre-ranking system. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 3292–3301, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557072. URL https://doi.org/10.1145/3511808.3557072.

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 689–698, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3186150. URL https://doi.org/10.1145/3178876.3186150.

Tie-Yan Liu. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.*, 3(3): 225–331, mar 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL https://doi.org/10.1561/1500000016.

Ben London and Ted Sandler. Bayesian counterfactual risk minimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4125–4133. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/london19a.html.

Malte Ludewig and Dietmar Jannach. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 28(4-5):331–390, oct 2018. doi: 10.1007/s11257-018-9209-6. URL https://doi.org/10.1007%2Fs11257-018-9209-6.

Alberto Lumbreras, Louis Filstroff, and Cédric Févotte. Bayesian mean-parameterized nonnegative binary matrix factorization. *arXiv preprint arXiv:1812.06866*, 2018.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H. Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web*

Conference 2020, WWW '20, page 463–473, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380130. URL https://doi.org/10.1145/3366423.3380130.

Jiaqi Ma, Xinyang Yi, Weijing Tang, Zhe Zhao, Lichan Hong, Ed Chi, and Qiaozhu Mei. Learning-to-Rank with Partitioned Preference: Fast Estimation for the Plackett-Luce Model. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 928–936. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/ma21a.html.

Yifei Ma, Yu-Xiang Wang, and Balakrishnan Narayanaswamy. Imitation-regularized offline learning. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2956–2965. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/ma19b.html.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836, apr 2020. ISSN 0162-8828. doi: 10.1109/TPAMI.2018.2889473. URL https://doi.org/10.1109/TPAMI.2018.2889473.

Yury A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Vaden Masrani, Tuan Anh Le, and Frank Wood. The thermodynamic variational objective. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/618faa1728eb2ef6e3733645273ab145-Paper.pdf.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample-variance penalization. In *Annual Conference Computational Learning Theory*, 2009. URL https://api.semanticscholar.org/CorpusID:17090214.

David McAllester. Simplified PAC-Bayesian margin bounds. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 203–215, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

David A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 230–234, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130570. doi: 10.1145/279943.279989. URL https://doi.org/10.1145/279943.279989.

Colin McDiarmid. Concentration. In *Probabilistic methods for algorithmic discrete mathematics*, pages 195–248. Springer, 1998.

Jincheng Mei, Chenjun Xiao, Bo Dai, Lihong Li, Csaba Szepesvari, and Dale Schuurmans. Escaping the gravitational pull of softmax. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21130–21140. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f1cf2a082126bf02de0b307778ce73a7-Paper.pdf.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6820–6829. PMLR, 13–18 Jul 2020b. URL https://proceedings.mlr.press/v119/mei20b.html.

Alberto Maria Metelli, Alessio Russo, and Marcello Restelli. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8119–8132. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4476b929e30dd0c4e8bdbcc82c6ba23a-Paper.pdf.

Zakaria Mhammedi, Peter Grünwald, and Benjamin Guedj. PAC-Bayes Un-Expected Bernstein Inequality. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3dea6b598a16b334a53145e78701fa87-Paper.pdf.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Thanh Nguyen-Tang, Sunil Gupta, A. Tuan Nguyen, and Svetha Venkatesh. Offline neural contextual bandits: Pessimism, optimization and generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=sPIFuucA3F.

Tui H Nolan and Matt P Wand. Accurate logistic variational message passing: algebraic and numerical details. *Stat*, 6(1):102–112, 2017.

Kento Nozawa, Pascal Germain, and Benjamin Guedj. PAC-Bayesian Contrastive Unsupervised Representation Learning. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 21–30. PMLR, 03–06 Aug 2020. URL https://proceedings.mlr.press/v124/nozawa20a.html.

Harrie Oosterhuis. Learning-to-rank at the speed of sampling: Plackett-luce gradient estimation with minimal computational complexity. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2266–2271, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531842. URL https://doi.org/10.1145/3477495.3531842.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000. ISSN 01621459. URL http://www.jstor.org/stable/2669533.

Art B Owen. *Empirical likelihood.* CRC press, 2001.

Art B. Owen. *Monte Carlo theory, methods and examples.* https://artowen.su.domains/mc/, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2346567.

Sebastian Prillo and Julian Martin Eisenschlos. Softsort: A continuous relaxation for the argsort operator. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Jérémie Rappaz, Julian McAuley, and Karl Aberer. *Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption*, page 390–399. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450384582. URL https://doi.org/10.1145/3460231.3474267.

Ankit Singh Rawat, Jiecao Chen, Felix Xinnan X Yu, Ananda Theertha Suresh, and Sanjiv Kumar. Sampled softmax with random fourier features. *Advances in Neural Information Processing Systems*, 32, 2019.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 452–461, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, page 175–186, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916891. doi: 10.1145/192844.192905. URL https://doi.org/10.1145/192844.192905.

C. J. Van Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, 2nd edition, 1979.

Ya'acov Ritov, Peter J Bickel, Anthony C Gamst, Bastiaan Jan Korneel Kleijn, et al. The bayesian analysis of complex, high-dimensional models: Can it be coda? *Statistical Science*, 29(4):619–639, 2014.

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951a. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, 22(3):400–407, 1951b.

David Rohde and Matt P Wand. Semiparametric mean field variational bayes: General principles and numerical issues. *The Journal of Machine Learning Research*, 17(1):5975–6021, 2016.

David Rohde, Stephen Bonner, Travis Dunlop, Flavian Vasile, and Alexandros Karatzoglou. Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. In *REVEAL workshop, ACM Conference on Recommender Systems 2018*, 2018.

David Rohde, Flavian Vasile, Sergey Ivanov, and Otmane Sakhi. Bayesian value based recommendation: A modelling based alternative to proxy and counterfactual policy based recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 742–744, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3411544. URL https://doi.org/10.1145/3383313.3411544.

D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, oct 2013. doi: 10.1613/jair.3987. URL https://doi.org/10.1613%2Fjair.3987.

Nicolas Le Roux. Tighter bounds lead to improved classifiers. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyAbMKwxe.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Francisco JR Ruiz, Michalis K Titsias, Adji B Dieng, and David M Blei. Augment and reduce: Stochastic inference for large categorical distributions. *arXiv preprint arXiv:1802.04220*, 2018.

Yuta Saito and Thorsten Joachims. Off-policy evaluation for large action spaces via embeddings. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19089–19122. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/saito22a.html.

Yuta Saito and Thorsten Joachims. Counterfactual evaluation and learning for interactive systems: Foundations, implementations, and recent advances. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4824–4825, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3542601. URL https://doi.org/10.1145/3534678.3542601.

Yuta Saito, Qingyang Ren, and Thorsten Joachims. Off-Policy Evaluation for Large Action Spaces via Conjunct Effect Modeling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29734–29759. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/saito23b.html.

Otmane Sakhi, Stephen Bonner, David Rohde, and Flavian Vasile. Reconsidering analytical variational bounds for output layers of deep networks, 2019.

Otmane Sakhi, Stephen Bonner, David Rohde, and Flavian Vasile. BLOB: A Probabilistic Model for Recommendation That Combines Organic and Bandit Signals. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining*, KDD '20, page 783–793, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403121. URL https://doi.org/10.1145/3394486.3403121.

Otmane Sakhi, Louis Faury, and Flavian Vasile. Improving Offline Contextual Bandits with Distributional Robustness. In *Proceedings of the ACM RecSys Workshop on Reinforcement Learning and Robust Estimators for Recommendation Systems (REVEAL '20)*, 2020b. URL https://arxiv.org/abs/2011.06835.

Otmane Sakhi, Pierre Alquier, and Nicolas Chopin. PAC-Bayesian Offline Contextual Bandits With Guarantees. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29777–29799. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/sakhi23a.html.

Otmane Sakhi, David Rohde, and Nicolas Chopin. Fast Slate Policy Optimization: Going Beyond Plackett-Luce. *Transactions on Machine Learning Research*, 2023b. ISSN 2835-8856. URL https://openreview.net/forum?id=f7a8XCRtUu.

Otmane Sakhi, David Rohde, and Alexandre Gilotte. Fast Offline Policy Optimization for Large Scale Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 9686–9694, Jun. 2023c. doi: 10.1609/aaai.v37i8.26158. URL https://ojs.aaai.org/index.php/AAAI/article/view/26158.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1257–1264, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/saunshi19a.html.

Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1670–1679, 2016.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

Yevgeny Seldin, Peter Auer, John Shawe-taylor, Ronald Ortner, and François Laviolette. PAC-Bayesian analysis of contextual bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/58e4d44e550d0f7ee0a23d6b02d9b0db-Paper.pdf.

Yevgeny Seldin, Nicolò Cesa-Bianchi, Peter Auer, François Laviolette, and John Shawe-Taylor. PAC-Bayes-Bernstein Inequality for Martingales and its Application to Multiarmed Bandits. In Dorota Glowacka, Louis Dorard, and John Shawe-Taylor, editors, *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, volume 26 of *Proceedings of Machine Learning Research*, pages 98–111, Bellevue, Washington, USA, 02 Jul 2012. PMLR. URL https://proceedings.mlr.press/v26/seldin12a.html.

Li Shen, Yan Sun, Zhiyuan Yu, Liang Ding, Xinmei Tian, and Dacheng Tao. On efficient training of large-scale deep learning models: A literature review, 2023.

Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/310ce61c90f3a46e340ee8257bc70e93-Paper.pdf.

James E. Smith and Robert L. Winkler. The optimizers curse: Skepticism and postdecision surprise in decision analysis. *Manage. Sci.*, 52(3):311–322, mar 2006. ISSN 0025-1909. doi: 10.1287/mnsc.1050.0451. URL https://doi.org/10.1287/mnsc.1050.0451.

Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, page 421–430, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595934472. doi: 10.1145/1180639.1180727. URL https://doi.org/10.1145/1180639.1180727.

Eleftherios Spyromitros-Xioufis, Symeon Papadopoulos, Ioannis Yiannis Kompatsiaris, Grigorios Tsoumakas, and Ioannis Vlahavas. A comprehensive study over VLAD and product quantization in large-scale image retrieval. *IEEE Transactions on Multimedia*, 16(6):1713–1728, 2014. doi: 10.1109/TMM.2014.2329648.

Joe Staines and David Barber. Variational Optimization, 2012. URL https://arxiv.org/abs/1212.4507.

Harald Steck. Autoencoders that don't overfit towards the identity. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19598–19608. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e33d974aae13e4d877477d51d8bafdc4-Paper.pdf.

A. Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pages 3–28, 2009.

Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. Doubly robust off-policy evaluation with shrinkage. In *International Conference on Machine Learning*, pages 9167–9176. PMLR, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 814–823, Lille, France, 07–09 Jul 2015a. PMLR. URL https://proceedings.mlr.press/v37/swaminathan15.html.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *NIPS*, 2015b.

Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudík, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3635–3645, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ugo Tanielian and Flavian Vasile. Relaxed Softmax for PU Learning. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 119–127, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362436. doi: 10.1145/3298689.3347034. URL https://doi.org/10.1145/3298689.3347034.

Ambuj Tewari and Susan A Murphy. From Ads to Interventions: Contextual Bandits in Mobile Health. In *Mobile Health*, pages 495–517. Springer, 2017.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-Confidence Off-Policy Evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Andrea Tirinzoni, Aymen Al-Marjani, and Emilie Kaufmann. Optimistic pac reinforcement learning: the instance-dependent view. In Shipra Agrawal and Francesco Orabona, editors, *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pages 1460–1480. PMLR, 20 Feb–23 Feb 2023. URL https://proceedings.mlr.press/v201/tirinzoni23a.html.

Michalis Titsias. One-vs-each approximation to softmax for scalable estimation of probabilities. In *Advances in Neural Information Processing Systems*, pages 4161–4169, 2016.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral Bandits for Smooth Graph Functions. In *International Conference on Machine Learning*, pages 46–54, 2014.

V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1558602224.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on*

Recommender Systems, RecSys '16, page 225–232, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959160. URL https://doi.org/10.1145/2959100.2959160.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017b. Curran Associates Inc. ISBN 9781510860964.

Paul Viallard, Pascal Germain, Amaury Habrard, and Emilie Morvant. A General Framework for the Practical Disintegration of PAC-Bayesian Bounds, 2023.

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

Mengting Wan and Julian J. McAuley. Item recommendation on monotonic behavior chains. In Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan, editors, *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM, 2018. doi: 10.1145/3240323.3240369. URL https://doi.org/10.1145/3240323.3240369.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. Fine-grained spoiler detection from large-scale review corpora. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1248. URL https://doi.org/10.18653/v1/p19-1248.

Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *Proc. VLDB Endow.*, 14(11):1964–1978, jul 2021a. ISSN 2150-8097. doi: 10.14778/3476249.3476255. URL https://doi.org/10.14778/3476249.3476255.

Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. A survey on session-based recommender systems. *ACM Comput. Surv.*, 54(7), jul 2021b. ISSN 0360-0300. doi: 10.1145/3465401. URL https://doi.org/10.1145/3465401.

Xuanhui Wang, Cheng Li, Nadav Golbandi, Mike Bendersky, and Marc Najork. The LambdaLoss Framework for Ranking Metric Optimization. In *Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, pages 1313–1322, 2018.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pages 3589–3597. PMLR, 2017.

Yuyan Wang, Mohit Sharma, Can Xu, Sriraj Badam, Qian Sun, Lee Richardson, Lisa Chung, Ed H. Chi, and Minmin Chen. Surrogate for Long-Term User Experience in Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4100–4109, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539073. URL https://doi.org/10.1145/3534678.3539073.

Robert West, Smriti Bhagat, Paul Groth, Marinka Zitnik, Francisco M. Couto, Pasquale Lisena, Albert Meroño Peñuela, Xiangyu Zhao, Wenqi Fan, Dawei Yin, Jiliang Tang, Linjun Shou, Ming Gong, Jian Pei, Xiubo Geng, Xingjie Zhou, Daxin Jiang, Benjamin Ricaud, Nicolas Aspert, Volodymyr Miz, Jennifer Dy, Stratis Ioannidis, undefinedlkay Yıldız, Rezvaneh Rezapour, Samin Aref, Ly Dinh, Jana Diesner, Alexey Drutsa, Dmitry Ustalov, Nikita Popov, Daria Baidakova, Shubhanshu Mishra, Arjun Gopalan, Da-Cheng Juan, Cesar Ilharco Magalhaes, Chun-Sung Ferng, Allan Heydon, Chun-Ta Lu, Philip Pham, George Yu, Yicheng Fan, Yueqi Wang, Florian Laurent, Yanick Schraner, Christian Scheller, Sharada Mohanty, Jiawei Chen, Xiang Wang, Fuli Feng, Xiangnan He, Irene Teinemaa, Javier Albert, Dmitri Goldenberg, Flavian Vasile, David Rohde, Olivier Jeunen, Amine Benhalloum, Otmane Sakhi, Yu Rong, Wenbing Huang, Tingyang Xu, Yatao Bian, Hong Cheng, Fuchun Sun, Junzhou Huang, Shobeir Fakhraei, Christos Faloutsos, Onur Çelebi, Martin Müller, Manuel Schneider, Olesia Altunina, Wolfram Wingerath, Benjamin Wollmer, Felix Gessert, Stephan Succo, Norbert Ritter, Evann Courdier, Tudor Mihai Avram, Dragan Cvetinovic, Levan Tsinadze, Johny Jose, Rose Howell, Mario Koenig, Michaël Defferrard, Krishnaram Kenthapadi, Ben Packer, Mehrnoosh Sameki, and Nashlie Sephus. Summary of tutorials at the web conference 2021. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 727–733, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383134. doi: 10.1145/3442442.3453701. URL https://doi.org/10.1145/3442442.3453701.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. doi: 10.1007/BF00992696.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):346–353, jul 2019. doi: 10.1609/aaai.v33i01.3301346. URL https://doi.org/10.1609%2Faaai.v33i01.3301346.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning to Rank: Theory and Algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1192–1199, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390306. URL https://doi.org/10.1145/1390156.1390306.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation, 2021.

Ming Xu, Matias Quiroz, Robert Kohn, and Scott A. Sisson. Variance reduction properties of the reparameterization trick. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2711–2720. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/xu19a.html.

Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 279–287, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016. doi: 10.1145/3240323.3240355. URL https://doi.org/10.1145/3240323.3240355.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. KDEformer: Accelerating transformers via kernel density estimation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 40605–40623. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zandieh23a.html.

Eva Zangerle and Christine Bauer. Evaluating recommender systems: Survey and framework. *ACM Comput. Surv.*, 55(8), dec 2022. ISSN 0360-0300. doi: 10.1145/3556536. URL https://doi.org/10.1145/3556536.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *J. Mach. Learn. Res.*, 2:527–550, 2002. URL http://dblp.uni-trier.de/db/journals/jmlr/jmlr2.html#Zhang02.

Xiaoying Zhang, Junzhou Zhao, and John C.S. Lui. Modeling the assimilation-contrast effects in online product rating systems: Debiasing and recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 98–106, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346528. doi: 10.1145/3109859.3109885. URL https://doi.org/10.1145/3109859.3109885.

Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767, 2002. ISSN 0885-064X. doi: https://doi.org/10.1006/jcom.2002.0635. URL https://www.sciencedirect.com/science/article/pii/S0885064X02906357.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Bandit contextuel hors-ligne : théorie et applications à grande échelle

Mots clés : Bandit contextuel hors-ligne, apprentissage à partir de données d'intéraction, apprentissage PAC-Bayésien, Recommandation en grande échelle.

Résumé : Cette thèse s'intéresse au problème de l'apprentissage à partir d'interactions en utilisant le cadre du bandit contextuel hors ligne. En particulier, nous nous intéressons à deux sujets connexes : **(1)** l'apprentissage de politiques hors ligne avec des certificats de performance, et **(2)** l'apprentissage rapide et efficace de politiques, pour le problème de recommandation à grande échelle. Pour **(1)**, nous tirons d'abord parti des résultats du cadre d'optimisation distributionnellement robuste pour construire des bornes asymptotiques, sensibles à la variance, qui permettent l'évaluation des performances des politiques. Ces bornes nous aident à obtenir de nouveaux objectifs d'apprentissage plus pratiques grâce à leur nature composite et à leur calibrage simple. Nous analysons ensuite le problème d'un point de vue PAC-Bayésien et fournissons des bornes, plus étroites sur les performances des politiques. Nos résultats motivent de nouvelles stratégies, qui offrent des certificats de performance sur nos politiques avant de les déployer en ligne. Les stratégies nouvellement dérivées s'appuient sur des objectifs d'apprentissage composites qui ne nécessitent pas de réglage supplémentaire. Pour **(2)**, nous proposons d'abord un modèle bayésien hiérarchique, qui combine différents signaux, pour estimer efficacement la qualité de la recommandation. Nous fournissons les outils computationnels appropriés pour adapter l'inférence aux problèmes à grande échelle et démontrons empiriquement les avantages de l'approche dans plusieurs scénarios. Nous abordons ensuite la question de l'accélération des approches communes d'optimisation des politiques, en nous concentrant particulièrement sur les problèmes de recommandation avec des catalogues de millions de produits. Nous dérivons des méthodes d'optimisation, basées sur de nouvelles approximations du gradient calculées en temps logarithmique par rapport à la taille du catalogue. Notre approche améliore le temps linéaire des méthodes courantes de calcul de gradient, et permet un apprentissage rapide sans nuire à la qualité des politiques obtenues.

Title : Offline Contextual Bandit : Theory and Large Scale Applications

Keywords : Offline Contextual Bandit, Off-Policy learning, PAC-Bayesian learning, Large Scale Recommendation.

Abstract : This thesis presents contributions to the problem of learning from logged interactions using the offline contextual bandit framework. We are interested in two related topics : **(1)** offline policy learning with performance certificates, and **(2)** fast and efficient policy learning applied to large scale, real world recommendation. For **(1)**, we first leverage results from the distributionally robust optimisation framework to construct asymptotic, variance-sensitive bounds to evaluate policies' performances. These bounds lead to new, more practical learning objectives thanks to their composite nature and straightforward calibration. We then analyse the problem from the PAC-Bayesian perspective, and provide tighter, non-asymptotic bounds on the performance of policies. Our results motivate new strategies, that offer performance certificates before deploying the policies online. The newly derived strategies rely on composite learning objectives that do not require additional tuning. For **(2)**, we first propose a hierarchical Bayesian model, that combines different signals, to efficiently estimate the quality of recommendation. We provide proper computational tools to scale the inference to real world problems, and demonstrate empirically the benefits of the approach in multiple scenarios. We then address the question of accelerating common policy optimisation approaches, particularly focusing on recommendation problems with catalogues of millions of items. We derive optimisation routines, based on new gradient approximations, computed in logarithmic time with respect to the catalogue size. Our approach improves on common, linear time gradient computations, yielding fast optimisation with no loss on the quality of the learned policies.

Institut Polytechnique de Paris
91120 Palaiseau, France