Lecture d'article : *An introduction to Variational Methods for graphical models*

COIC Quentin, NAAMANE Chemsdin, ZIANE Krim

1 Introduction

Comme vu en cours, on sait résoudre efficacement les problèmes d'apprentissage et d'inférence sur les arbres, grâce par exemple au *sum-product algorithm*. Si on note N le nombre de noeuds dans l'arbre, cet algorithme permet d'atteindre une complexité linéaire en N, au lieu d'une complexité exponentielle avec un algorithme naïf. Dans le cas d'un modèle graphique plus général, la résolution peut impliquer une complexité trop élevée pour l'utilisation de méthodes exactes. Par exemple, dans le cas d'une famille exponentielle, nous avons vu que l'inférence fait intervenir un programme de maximisation, certes convexe, mais dont l'ensemble admissible et/ou la fonction objectif peuvent se révéler trop complexes pour une résolution efficace. Pour remédier à cela, on utilise des techniques d'approximation, comme les méthodes *MCMC*, qui permettent d'échantillonner des distributions complexes mais qui peuvent aussi être lentes à converger. On a alors vu comme alternative un exemple de méthode variationnelle via l'approche *mean field*.
Ce papier donne d'abord l'opportunité d'illustrer la complexité des méthodes exactes à travers le *Junction Tree Algorithm* cité en cours. Puis, permet d'approfondir la théorie des méthodes variationnelles. Enfin, on terminera par un exemple d'application via les *Boltzmann machines*.

2 Complexité des méthodes exactes

Une illustration de la complexité des méthodes exactes est faite à l'aide de l'exemple du *Junction Tree Algorithm*. Son objectif est de transformer un graphe en arbre dont chaque noeud représente un clique et sur lequel il est possible d'appliquer un algorithme de transmission de messages selon la logique vue en cours. La restructuration se base sur deux étapes.
La première étape est la *moralization* qui s'effectue sur les graphes dirigés. Elle a pour objectif d'obtenir le plus petit graphe non dirigé (en terme de nombre d'arêtes ajoutées) pour lequel la loi jointe du graphe de base factorise également. La seconde étape est la *triangulation*. Son but est de permettre le calcul récursif des probabilités, en conditionnant chaque clique uniquement par rapport à son prédécesseur. Le plus petit graphe pour lequel cela n'est pas possible est le *four-cycle graph*, c'est pourquoi on ajoute à toutes ces structures une arête, qui liera deux nœuds non voisins. On obtient au final un arbre de cliques sur lequel la résolution est théoriquement possible. Cependant, il apparaît que la structure ainsi créée induit une complexité de calcul exponentielle en la taille des cliques. Ainsi, certains problèmes sont numériquement ingérables par cet algorithme.

Compte tenu de cet obstacle, on se tourne vers des méthodes de résolutions approchées, telles que les méthodes variationnelles.

3 Les méthodes variationnelles

Tout d'abord, on se place dans le cadre d'un modèle graphique $G = (S, F)$ pour lequel on considère le problème d'inférence suivant:

$$\mathbb{P}(H|E) = \frac{\mathbb{P}(H, E)}{\mathbb{P}(E)}$$

Avec, E un ensemble de nœuds observés et H un ensemble de nœuds non-observés.

Les méthodes variationnelles consistent en la transformation du graphe originel en une structure simplifiée sur laquelle l'application d'une méthode exacte est envisageable. Cette opération peut être effectuée de deux manières.

Premièrement, l'approche séquentielle consiste à transformer successivement chaque nœud jusqu'à obtenir un graphe qui propose un bon compromis entre la complexité de calcul et la précision du résultat.
Pour illustrer ces transformations, on se place dans le cadre d'un modèle graphique $G = (S, F)$ dirigé. Une probabilité $\mathbb{P}$ qui factorise ce dernier s'écrit alors:

$$\mathbb{P}(S) = \prod_{i=1}^{|S|} \mathbb{P}(S_i|S_{\pi_i})$$

L'objectif est ici de borner chacun des termes du produit. La méthode consiste à se baser sur des concepts du domaine d'optimisation convexe, et plus particulièrement sur le principe de la fonction conjuguée. Pour pouvoir appliquer cela, il faut que chaque terme de ce produit soit convexe ou concave à une transformation près. Cette transformation peut s'effectuer par composition avec une autre fonction ou par changement de variable.
Rappelons qu'une fonction convexe f peut s'écrire:

$$\forall x, f(x) = \max_{\lambda}\{\lambda^T x - f^*(\lambda)\} \tag{1}$$

avec f^* sa fonction conjuguée. On obtient alors une famille paramétrée par λ de fonctions linéaires en x qui bornent inférieurement f. Si on avait pris f concave, on aurait pu obtenir une famille qui borne supérieurement, via un argument analogue.
En considérant le cas où les $\mathbb{P}(S_i|S_{\pi_i})$ vérifient les conditions précédentes, on a:

$$\forall \lambda_i^L, \mathbb{P}(S) = \prod_{i=1}^{|S|} \mathbb{P}(S_i|S_{\pi_i}) \geq \prod_{i=1}^{|S|} F_i(S_i, S_{\pi_i}, \lambda_i^L)$$

Avec $\forall i$ F_i est une fonction linéaire en (S_i, S_{π_i}) obtenue par le procédé décrit au paragraphe précédent à l'aide de (1) appliqué à $\mathbb{P}(S_i|S_{\pi_i})$.
C'est cette linéarisation qui diminue la complexité de résolution. Cependant, le contrecoup de ces méthodes est l'introduction de paramètres supplémentaires $(\lambda_i)_{1\leq i\leq |S|}$, dits variationnel. On devra alors optimiser la borne inférieure obtenue par rapport à ces paramètres, afin d'obtenir une précision satisfaisante. Notons qu'on aurait pu borner seulement une partie des probabilités conditionnelles (logique séquentielle) et que pour approximer une probabilité marginale il suffit de sommer sur la borne obtenue.

La seconde approche, dite "par bloc", consiste à transformer en une fois tout ou partie d'un graphe notamment lorsqu'une structure simplifiée est observable sur le graphe d'origine. On obtient alors une famille $(\mathbb{Q}_\lambda)_\lambda = (\mathbb{Q}(H|E,\lambda))_\lambda$ associée au graphe simplifié, qui approche $\mathbb{P}(H|E)$. Puis on optimise par rapport à λ de sorte à approcher au mieux la distribution du graphe d'origine. Plus précisément, comme en cours, on minimise la divergence de Kullback-Leibler entre la distribution initiale et la distribution simplifiée. Une des justifications de l'utilisation de ce critère pour l'approximation se base sur la démonstration de l'algorithme EM vu en cours. En effet en s'intéressant à la log-vraisemblance des nœuds observés E et en notant H les nœuds non observés, qui jouent le rôle de variables latentes, on a que :

$$\begin{aligned} ln(\mathbb{P}(E)) = ln(\sum_{\{H\}} \mathbb{P}(H,E)) &= ln(\sum_{\{H\}} \mathbb{Q}(H|E,\lambda)\frac{\mathbb{P}(H,E)}{\mathbb{Q}(H|E,\lambda)}) \\ &\geq \sum_{\{H\}} \mathbb{Q}(H|E,\lambda) ln(\frac{\mathbb{P}(H,E)}{\mathbb{Q}(H|E,\lambda)}) \end{aligned} \tag{2}$$

Or:

$$ln(\mathbb{P}(E)) - \sum_{\{H\}} \mathbb{Q}(H|E,\lambda) ln(\frac{\mathbb{P}(H,E)}{\mathbb{Q}(H|E,\lambda)}) = D(\mathbb{Q}(H|E,\lambda)||\mathbb{P}(H|E)) \geq 0$$

Ainsi, la minimisation par rapport à λ de la Divergence de Kullback Leibler permet à la fois d'obtenir la meilleure approximation de $\mathbb{P}(H|E)$ et la meilleure borne inférieure sur la log-vraisemblance en se restreignant à la famille $(\mathbb{Q}_\lambda)_\lambda$.

Si l'on voit maintenant la log-vraisemblance $ln(\mathbb{P}(E))$ comme une fonction d'un paramètre θ, on peut, via la démonstration précédente, effectuer un algorithme EM afin d'estimer ce paramètre. La seule différence par rapport à l'algorithme EM original est que lors de l'*Expectation step*, la distribution $\mathbb{Q}$ sera déterminée en minimisant la divergence de Kullback-Leibler dans la famille paramétrée par λ.

On va désormais illustrer les méthodes décrites à l'aide de l'exemple des *Boltzmann machines*.

4 Exemples d'application: *Boltzmann Machines*

Une *Boltzmann machine* est un modèle graphique non dirigé composé de nœuds à valeurs binaires et tel que sa loi jointe s'écrit :

$$\mathbb{P}(S) = \frac{exp(\sum_{i<j} \theta_{ij} S_i S_j + \sum_i \theta_{i0} S_i)}{Z} \text{ , avec } \theta_{ij} = 0 \text{ Si } S_i \text{ et } S_j \text{ ne sont pas voisins.}$$

Notons que si on s'intéresse à la distribution $\mathbb{P}(H|E)$, on a toujours une *Boltzmann machine* sur les noeuds de H où $\tilde{\theta}_{ij} = \theta_{ij}$ et $\tilde{\theta}_{i0} = \theta_{i0} + \sum_{j\in E} \theta_{ij} S_j$, $\forall i, j \in H$.

4.1 Méthodes variationnelles pour l'inférence

Ce modèle est caractérisé par une paramétrisation particulière des *clique potentials* mais sa structure reste générale et donc, dans la plupart des cas, ingérable par une méthode de résolution exacte.
On peut remarquer que cette modélisation est semblable à celle de l'*Ising model* pour lequel on a déjà vu la résolution par l'approche *mean field*. A noter que cette dernière est un cas "extrême" de l'approche par blocs, au sens où elle amène à considérer le graphe indépendant associé, c'est-à-dire dont la loi jointe s'écrit, pour un $\lambda \in [0,1]^{|S|}$:

$$\mathbb{Q}_\lambda(S) = \prod_i \lambda_i^{S_i} (1-\lambda_i)^{1-S_i}$$

On va maintenant s'intéresser à l'approche séquentielle.

Pour approximer une probabilité conditionnelle, il faut réussir à borner des probabilités marginales. On se concentrera ici sur le calcul d'une borne inférieure de ces dernières.
Une telle probabilité est en fait un ensemble de sommes sur un certain nombre de nœuds du graphe. L'approche séquentielle consistera donc à borner une à une ces sommes comme suit :

$$ln(\sum_{S_i\in\{0,1\}} exp(\sum_{j<k} \theta_{jk} S_j S_k + \sum_j \theta_{j0} S_j)) = \sum_{\{j<k\}\neq i} \theta_{jk} S_j S_k + \sum_{j\neq i} \theta_{j0} S_j + ln(1+exp(\sum_{j\neq i} \theta_{jk} S_j + \theta_{i0}))$$

$$\geq \sum_{\{j<k\}\neq i} \theta_{jk} S_j S_k + \sum_{j\neq i} \theta_{j0} S_j + \lambda_i^L (\sum_{j\neq i} \theta_{jk} S_j + \theta_{i0}) + H(\lambda_i^L)$$

En effet $\forall x, \lambda$: $ln(1+e^x) \geq \lambda x + H(\lambda)$ où H est la *binary entropy function*, l'opposée de la fonction conjuguée de $x \mapsto ln(1+e^x)$. On a : $H : \lambda \mapsto \lambda ln(\lambda) + (1-\lambda) ln(1-\lambda)$.

On peut donc observer que l'effet de l'approximation de la marginalisation du noeud i par la méthode variationnelle est l'obtention d'une *Boltzmann machine* où ce noeud a été déconnecté, tout en laissant une trace sur les paramètres de ses voisins (qui forment sa *Markov blanket*) :

$$\begin{array}{lcll} \tilde{\theta}_{jk} & = & \theta_{jk} & j,k \neq i \\ \tilde{\theta}_{j0} & = & \theta_{j0} + \lambda_i^L \theta_{ij} & j \neq i \end{array}$$

Si on effectue cette opération pour tous les noeuds du graphe on obtient la borne suivante sur Z, la *Partition function* :

$$\sum_{i<j} \theta_{ij}\lambda_i^L\lambda_j^L + \sum_i \theta_{i0}\lambda_i^L - \sum_i [\lambda_i^L ln(\lambda_i^L) + (1-\lambda_i^L)ln(1-\lambda_i^L)]$$

On peut observer que maximiser cette dernière expression par rapport à λ_i^L revient à minimiser la divergence de Kullback-Leibler dans le cadre *mean field* mais sans le terme $ln(Z)$ qui n'intervient pas pour trouver les λ_i^L optimaux. Si on note σ la fonction *sigmoid*, et qu'on identifie θ_{ij} à θ_{ji} si $j < i$ on obtient une formule identique au cas *mean field* :

$$\lambda_i^L = \sigma(\sum_{j\neq i} \theta_{ij}\lambda_j^L + \theta_{i0})$$

On pourrait donc interpréter l'approximation de la marginalisation d'un noeud via l'approche séquentielle comme une approximation de $\mathbb{P}(S)$ ou $\mathbb{P}(H|E)$ en déconnectant successivement les noeuds au lieu de tous les déconnecter à la fois comme dans l'approche *mean field*. Une stratégie pour calculer $\mathbb{P}(S)$ ou $\mathbb{P}(H|E)$ serait donc de transformer nœud après nœud le graphe, tout en évitant de perdre trop en précision, jusqu'à obtenir une structure assez simple pour être géré par un algorithme de résolution exact. Remarquons que cela est possible car dans l'approche séquentielle, on garde toujours une *Boltzmann machine*, après déconnection d'un noeud, qui n'induit pas de nouveaux liens entre les voisins de ce dernier.

4.2 Méthodes variationnelles pour l'apprentissage

On s'intéresse pour finir au problème d'apprentissage des paramètres θ_{ij} et θ_{i0}, $\forall i, j$. Dans cet exemple, l'inégalité (2) s'écrit :

$$ln(\mathbb{P}(E|\theta)) \geq \sum_{i<j} \theta_{ij}\lambda_i\lambda_j + \sum_i \theta_{i0}\lambda_i - ln(Z) - \sum_i [\lambda_i ln(\lambda_i) + (1-\lambda_i)ln(1-\lambda_i)]$$

Pour obtenir la formule de mise à jour de θ_{ij} qu'on obtiendrait avec la variante de l'algorithme EM décrit précédemment, on devrait dériver la borne inférieure ci-dessus. On aurait alors que la dérivée partielle du terme $ln(Z)$ par rapport à θ_{ij} est l'espérance par rapport à $\mathbb{P}(S|\theta)$ de S_iS_j. Malheureusement, le calcul de ce terme n'est évidemment pas réalisable compte tenu de notre hypothèse que le calcul de la probabilité jointe n'est pas gérable par une méthode de résolution exacte. On pourrait calculer une approximation de cette espérance mais on aurait alors aucune garantie d'accroître la log-vraisemblance à chaque étape.

5 Conclusion

Comme vu précédemment, plusieurs cas d'application sont sujets à une compléxité de calcul trop importante pour une résolution exacte. Pour répondre à ce problème, les méthodes variationnelles peuvent apporter une réponse intéressante en fournissant une solution approchée aux problèmes d'inférence et d'apprentissage. Par exemple pour les *Boltzmann machines*, il a été observé que l'approche *mean field* fournissait des résultats équivalents en précision à ceux d'une approche par *Gibbs sampling* mais en un temps de calcul bien plus rapide. Cependant, pour l'apprentissage le calcul de la dérivée partielle de la *log-partition function* est problématique, ce qui n'est pas le cas dans d'autres modèles abordés dans l'article. Ceci souligne alors une des limites des méthodes variationnelle : il n'existe pas de processus d'application systématique. En effet, on connaît des cas pour lesquels ces méthodes s'appliquent bien, mais rien ne garantit leur universalité.