Fondements de l'apprentissage machine

Automne 2014

Roland Memisevic

Leçon 9

Plan

- Modèles graphiques orientés. (Directed graphical models.)
- Modèles graphiques non-orientés. (Undirected graphical models.)
- Passage de message et la propagation des croyances. (Message passing and belief propagation.)

Rappel : Les deux règles fondamentales pour la manipulation des probabilités

- **Règle de la somme (Sum rule) :**

$$p(X) = \sum_Y p(X, Y)$$

- **Règle du produit (Product rule) :**

$$p(X, Y) = p(Y|X)p(X)$$

Des exemples

$$p(a, b, c, d, e) = p(a, c, d|b, e)p(b, e)$$

$$p(a, b|c) = p(a|b, c)p(b|c)$$

$$p(a) = \sum_b p(a, b)$$

$$p(a|c) = \sum_b p(a, b|c)$$

Modèles graphiques probabilistes (Probabilistic Graphical Models)

- Un graphe est un ensemble de *noeuds* et *d'arêtes*.
- Les graphes sont utiles pour représenter les distributions probabilistes, parce qu'ils peuvent traduire des opérations mathématiques complexes en manipulations simples et intuitives.
- Il existe deux types de modèles graphiques probabilistes :
  1. Modèles graphiques orientés (directed graphical models, Bayesian network/Bayes net.)
  2. Modèles graphiques non-orientés (undirected graphical models.)
- (Le récent "factor graph" peut représenter les deux dans un seul formalisme.)

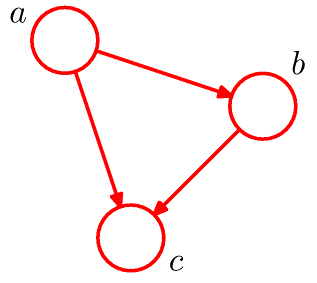
Modèles graphiques orientés

- Nous pouvons écrire

$$p(a, b, c) = p(c|a, b)p(a, b) = p(c|a, b)p(b|a)p(a)$$

pour n'importe quelle distribution jointe sur a, b, c.

- Le modèle graphique orienté pour cette factorisation représente chaque variable comme un noeud, et chaque distribution conditionnelle comme une arête :

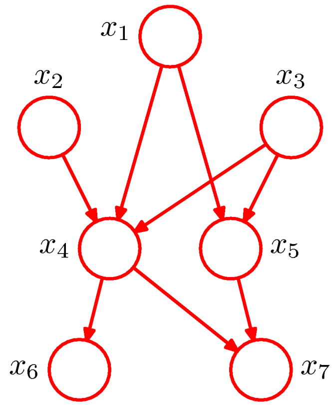


Modèles graphiques orientés

$$p(x_1, \dots, x_K) \quad = \quad p(x_K|x_1, \dots, x_{K-1}) \cdots p(x_2|x_1)p(x_1)$$

- Si K est grand, il faut utiliser beaucoup de paramètres pour représenter toutes les distributions conditionnelles.
- L'idée principale des modèles graphiques est de laisser quelques variables de côté dans certaines distributions conditionnelles. Cela implique que nous faisons des *suppositions d'indépendance conditionnelle*.

Exemple



$$p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

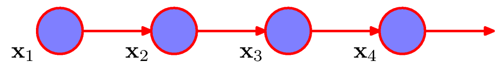
Graphe orienté acyclique

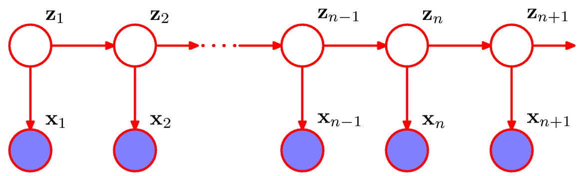
- Plus généralement, nous définissons

$$p(\mathbf{x}) \quad = \quad \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

  où pa_k représente l'ensemble des noeuds parents du noeud k.
- La seule restriction qui est nécessaire : il ne faut *pas* qu'il y ait de *cycles orientés* : Le graphe doit être un graphe orienté acyclique (DAG).

Modèles de Markov, HMM

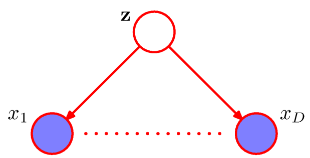




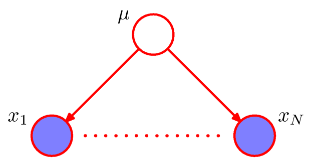
- L'*ombrage* (shading) est une façon courante de distinguer les variables observées des variables non observées (latentes).
- (Notez que dans les deux modèles ci-dessus, les paramètres sont généralement liés, de sorte que toutes les distributions conditionnelles sont les mêmes.)

Plus d'exemples

- Naive Bayes (le graphe représentant une seule point) :

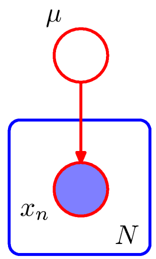


- Un modèle gaussien Bayésien :



Des plaques (Plates)

- Souvent, un modèle contient plusieurs copies du même type de variable avec la même distribution conditionnelle.
- Pour les représenter, nous utilisons une plaque (plate), qui est une boîte dans laquelle le nombre de copies de la variable est indiqué dans le coin.
- Exemple : le modèle gaussien Bayésien :



Indépendance conditionnelle

- Des variables aléatoires a, b sont conditionnellement indépendantes sachant c si

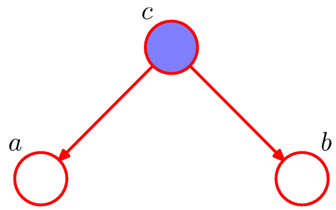
$$p(a|b,c) = p(a|c)$$

- Dans ce cas

$$\begin{aligned} p(a,b|c) &= p(a|b,c)p(b|c) \\ &= p(a|c)p(b|c) \end{aligned}$$

- Des propriétés de l'indépendance conditionnelle d'une distribution peuvent être déduites en regardant son graphe.
- Ceci est connu comme la d-séparation (d-separation), ou "Bayes Ball algorithm", qui peut être derivée en considérant les trois composantes suivantes d'un graphe orienté :

Trois graphes fondamentaux

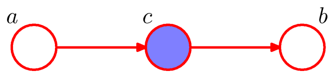


Graphe 1

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

a, b sont conditionnellement indépendants sachant c

Trois graphes fondamentaux

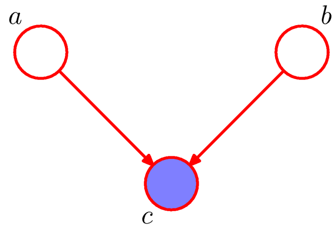


Graphe 2

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

a, b sont conditionnellement indépendants sachant c

Trois graphes fondamentaux



Graphe 3

$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

a, b ne sont pas nécessairement indépendants. (il existe des distributions compatibles avec ce graphe où ils ne sont pas)

Explaining away

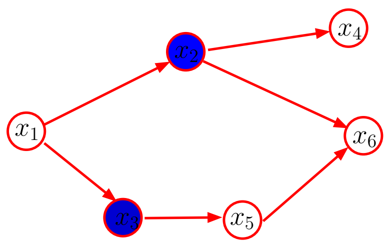
- Dans les cas 1 et 2, conditionner sur c va *bloquer le chemin* de a à b.
- Dans le cas 3, conditionner sur c ne va pas bloquer le chemin.
- En fait, en marginalisant les deux côtés de l'expression $p(a, b, c) = p(a)p(b)p(c|a, b)$ on voit que $p(a, b) = p(a)p(b)$. Donc *sans* conditionner sur c, les variables a et b sont indépendantes.
- Le phénomène selon lequel conditionner sur c peut rendre a et b dépendants est connu comme explaining away.
- Un exemple de explaining away : Soit a et b les résultats indépendants du lancer de deux pièces. Soit $c = 1$ si les deux résultats sont les mêmes, et 0 sinon. Sachant c, a et b deviennent dépendants !

d-separation

- On peut montrer que explaining away (le déblocage de chemin par conditionnement) se produit si *l'un des descendants du noeud* c est observé.
- Deux variables a, b sont appelées d-separated par le noeud c si tous les chemins de a à b via c sont bloqués (donc c ne fait pas partie d'un *explaining away*-sous-graphe).
- La *d-separation* et équivalente à l'indépendance conditionelle. Cela permet de déterminer l'indépendance conditionelle en examinant le graphe :

d-separation/Bayes Ball algorithm

- Prenez trois groupes de variables aléatoires A, B, C, qui sont des sous-ensembles de noeuds dans un modèle graphique orienté. A et B sont conditionnellement indépendants sachant C si aucun chemin d'un noeud dans A à un noeud dans B via un noeud dans C n'est bloqué.

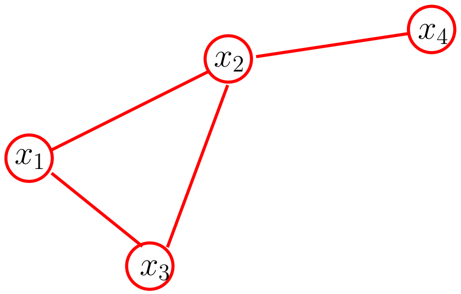


x_1 et x_6, sont-elles indépendentes sachant x_2 et x_3 ?

Modèles graphiques non-orientés

- Des graphes non-orientés peuvent également être utilisés pour définir des distributions probabilistes.
- Pour les modèles graphiques non-orientés il n'y a pas de explaining away, mais l'apprentissage peut être plus difficile.
- Les définitions suivantes seront importantes : Une clique est un ensemble de noeuds dans un graphe qui sont entièrement connectés (fully connected).
- Une clique maximale est une clique qui cesse d'en être une si on inclut un autre noeud du graphe.

Cliques and cliques maximales



- x_1, x_2 forment une clique.
- x_1, x_2, x_3 forment une clique maximales.
- x_2, x_4 forment une cliques maximales.
- etc.

Fonctions potentielles (potential functions)

- Il est courant de laisser $\mathbf{x}_C$ représenter le vecteur contenant toutes les variables qui font partie de la clique C.
- Les modèles graphiques non-orientés sont définis en utilisant des fonctions potentielles (potential functions) positives $\psi_C(\mathbf{x}_C)$ sur les instantiations des variables dans les cliques maximales du graphe.
- Les fonctions potentielles sont comme les probabilités non-normalisées. Elles permettent d'exprimer des configurations désirables (grande valeur pour $\psi(\mathbf{x}_C)$) ainsi que des configurations peu désirables (petite valeur pour $\psi(\mathbf{x}_C)$) de $\mathbf{x}$.

Modèles graphiques non-orientés

Modèles graphiques non-orientés

Soient $\mathbf{x}_C$, $C = 1, \ldots, M^C$, les cliques maximales du graphe. Un modèle graphique non-orienté définit la distribution

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

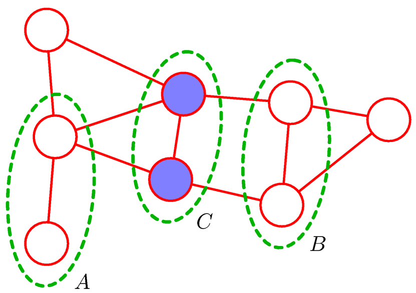
$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

- La constante Z s'appelle fonction de partition.

Modèles graphiques non-orientés

Des indépendances conditionnelles dans des modèles non-orientés

Dans un modèle non orienté, deux sous-ensembles de variables, A, B, sont conditionnellement indépendants, sachant un ensemble C, si chaque chemin d'un noeud dans A à un noeud dans B passe par un noeud dans C.



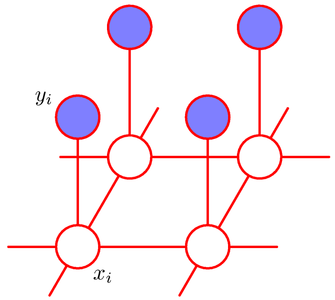
Définir les fonctions potentielles

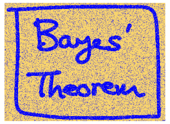
- En pratique, il est courant d'utiliser la définition

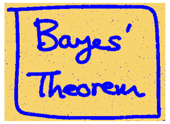
$$\psi(\mathbf{x}_C) = \exp(-E(\mathbf{x}_C))$$

  afin d'assurer la positivité, avec une fonction "d'énergie" $E(\mathbf{x}_C)$ qui retourne de grandes valeurs pour des configurations improbables et de petites valeurs pour des configurations probables d'une clique.
- L'apprentissage correspond a régler les paramètres de la fonction d'énergie pour maximiser la probabilité des données d'entraînement.

Markov Random Fields







- Le Markov Random Field est un modèle graphique non-orienté avec des variables latentes x_i définies sur des pixels y_i. Il est utilisé, par exemple, pour le débruitage.

Inférence dans les modèles graphiques

- Nous pouvons exprimer chaque modèle graphique dirigé comme un modèle non-orienté en utilisant les fonctions potentielles appropriées.
- Ainsi, ce qui suit est valable pour les deux types de modèles.

Inférence dans les modèles graphiques

- Pour les modèles qui définissent les chaines calculer les distributions marginales $p(x_n)$ pour n'importe quel noeud, x_n, il faut calculer la somme

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(\mathbf{x})$$

- Naivement cela correspond à des calculs d'ordre K^N. Mais en utilisant la distributivité, nous pouvons réduire cela à K^2 (voir leçon 8) :

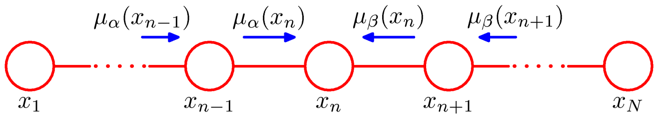
$$\begin{aligned} p(x_n) \quad = \quad & \Big[\underbrace{\frac{1}{Z} \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \Big[\sum_{x_2} \psi_{2,3}(x_2, x_3) \Big[\sum_{x_1} \psi_{1,2}(x_1, x_2) \Big] \Big]}_{\mu_\alpha(x_n)} \cdots \Big] \\ & \underbrace{\Big[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \Big[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \Big] \cdots \Big]}_{\mu_\beta(x_n)} \end{aligned}$$

Inférence comme le passage de messages (message passing)

- La probabilité marginale est

$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n)$$

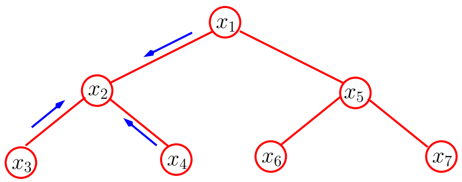
- Chaque variable $\mu_\alpha(x_n)$ et $\mu_\beta(x_n)$ est un vecteur (de taille K si chaque x_i a K états).
- Une idée essentielle est que nous puissions interpréter les vecteurs $\mu_\alpha(x_n)$ et $\mu_\beta(x_n)$ comme des messages qui sont passés du noeud x_{n-1} ou x_{n+1} au noeud x_n.
- Les messages sont *locaux*, et les calculer est une opération locale :



Inférence comme le passage de messages (message passing)

- Supposons que nous voulons calculer $p(x_m)$ pour un $m \neq n$.
- Nous pourrions le faire en utilisant un autre $O(K^2)$ calculs. Mais cela serait toujours du gaspillage parce que nous aurions à recalculer un grand nombre de fois les mêmes messages.
- Comme chaque $p(x_m) = \frac{1}{Z}\mu_\alpha(x_m)\mu_\beta(x_m)$, une meilleure idée est de calculer *tous* les $p(x_m)$ à la fois, en utilisant :
  - Un seul forward pass pour les μ_α.
  - Un seul backward pass pour les μ_β.
- Nous pouvons également calculer les marginaux pur des paires $p(x_{m-1}, x_m)$ de cette façon.

L'inférence dans les arbres



- Mais l'avantage fondamental de penser à la sommation partielle comme un message, c'est qu'il permet de généraliser l'idée d'inférence efficace à n'importe quel modèle graphique qui est un arbre.
- La raison pour laquelle cela fonctionne est que dans un arbre, il y a exactement un chemin de tout noeud à la racine.
- Ceci est connu comme l'algorithme somme-produit (sum-product algorithm), ou belief propagation :

Belief propagation

L'algorithme somme-produit

1. Déclarer un noeud de l'arbre comme noeud racine.
2. Passer des messages de feuilles à la racine.
3. Passer des messages de la racine aux feuilles.
4. Calculer les marginaux $p(\mathbf{x}_C)$ comme des produits locaux (normalisés si necessaire).

- (En remplaçant des sommes par des $\max$, nous pouvons également trouver les valeurs qui *maximisent* la probabilité, par analogie avec l'algorithme de Viterbi pour HMM. Cela s'appelle l'algorithme max-produit (max-product algorithm).)

L'inférence conditionnellement

- Souvent, il est nécessaire de calculer des marginaux conditionnels, tel que le $p(x_n|\mathbf{y})$.
- Mais le conditionnement peut être considéré comme la définition d'un nouveau graphe sur les variables non conditionnées (un graphe qui est une fonction de variables de conditionnement).
- Donc conditionner ne change pas la façon dont nous faisons l'inférence.

Loopy belief propagation

- Si le graphe n'est pas un arbre, l'algorithme sum-produit n'est bien défini, car il y aurait des dépendances circulaires entre les messages.
- Mais on a trouvé que, si on utilise l'algorithme quand même, itérativement, dans certains conditions, il va converger vers une solution. Ceci est connu comme loopy belief propagation (Loopy BP).
- Cette approche est commune dans les tâches de traitement d'image (comme dans l'exemple de débruitage d'auparavant).