**Introduction aux modèles graphiques** **2007/2008**

Cours 9 — 19 decembre 2007

*Enseignant: Francis Bach* *Scribe: Nicolas Charon, Xavier Fernandez*

# 9.1 Entropie

On suppose que la variable aléatoire $X$ prend des valeurs dans un ensemble finite $\mathcal{X}$.

**Définition 9.1 (Entropie H)**

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x) = \mathbf{E}(\log \frac{1}{p(x)})$$

**Proposition 1** *1.* $H(X) \geq 0$

*2.* $H(X) \leq Card\mathcal{X}$

**Démonstration** Pour 1, $\frac{1}{p(x)} \geq 1, \forall x \in \mathcal{X}$ implique le résultat. Pour le 2, on peut appliquer l'inégalité de Jensen pour la fonction convexe $p \mapsto p \log p$. ■

**Lecture :** Elements of Information Theory, pat Thomas M. Cover, Joy A. Thomas, Wiley.

# 9.2 KL-divergence (Kullback–Leibler)

## 9.2.1 Définition et propriètés de base

**Définition 9.2 (KL-divergence D)**

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log(\frac{p(x)}{q(x)})$$

*avec p(x) et q(x) deux lois*

**Proposition 2** *1.* $D(p||q) \geq 0$ *avec égalité ssi* $p=q$

*2. non symétrique :* $D(p||q) \neq D(q||p)$

**Démonstration**
$D(p||q) = -\sum_{x \in X} p(x) \log(\frac{q(x)}{p(x)}) \geq -\log(\sum_{x \in X} p(x) \cdot \frac{q(x)}{p(x)}) = 0$, par Jensen
Jensen : égalité ssi p=q ■

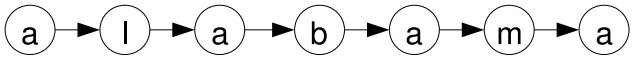


FIG. 9.1 :

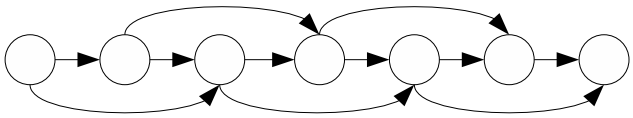

FIG. 9.2 :

### 9.2.2 Lien entre KL-divergence et Maximum Likelihood

$x \in \mathcal{X}$ fini
$x_1, ..., x_n$ observations iid

**Définition 9.3 (Loi empirique)**

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \delta(x = x_n)$$

**Proposition 3**

$$\begin{aligned} D(\hat{p}||q) &= \sum_{x \in \chi} \hat{p}(x) \log(\frac{\hat{p}(x)}{q(x)}) \\ &= -H(\hat{p}(x)) - \sum_{x \in \chi} \hat{p}(x) \log(q(x)) \\ &= -H(\hat{p}(x)) - \sum_{n=1}^{N} \log(p(x_n)) \end{aligned}$$

**Lien :** maximiser $\sum_{n=1}^{N} logq(x_n) \Leftrightarrow minimiser D(\hat{p}||q)$

ceci n'est pas équivalent à $D(q||\hat{p})$

### 9.2.3 Lien entre KL-divergence et entropie

$D(p||uniforme) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{1/card\mathcal{X}} = -H_{p(x)}(X) + \log card\mathcal{X}$

## 9.3 "Features"

Dans le cadre du traitement du texte, nous avons un sommet par lettre *cf.*figure 9.1.
Dans ce cas, le nombre de paramètres est de 26x26 (transition)
Pour un modele d'ordre 2, on en a 26x26x26 (ordre 2, *cf.*figure 9.2.)
Cependant, certaines transitions sont plus ou moins rare (i.e., z-t est assez rare contrairement à i-n-g en anglais).
On considère un modèle non-orienté G = (V,E), de loi $p(x) \in L(G) \Leftrightarrow p(x) = \frac{1}{2} \prod_{c \in C} \Psi_c(x_c)$
avec $\Psi_c$ potentiels et $c \in C$ cliques(maximales)

**Cas général :**
Il faut spécifier $\Psi_c(x_c)$ pour toutes les valeurs possibles.
Utilisation de "features" : $\Psi_c(x_c) = \prod_{i \in I_c} \exp(\theta_{i,c} f_{i,c}(x_c))$
(les $f_i$ sont les features)

**Cas particulier :** $f_{i,c}$ contiennent tous les diracs (on retrouve le modele complet, dit saturé).
C=$\{1;2\}$ (variables binaires)
$\Psi_{12}(x_1, x_2) = e^{\theta_1(x_1=1,x_2=1)} e^{\theta_2(x_1=0,x_2=1)} e^{\theta_3(x_1=1,x_2=0)} e^{\theta_4(x_1=0,x_2=0)}$

**Bénèfices :** $\Rightarrow$ moins de paramètres
$\Rightarrow$ reste factorisé (infèrence efficace)

## 9.4 Apprentissage des paramètres

**Vu :** DAG, paramètrisation découplée

**Simplification avec utilisation de features :** $p(x) = \frac{1}{Z} \prod_{c \in C} \prod_{i \in I_c} \exp(\theta_{i,c} f_{i,c}(x))$
$p(x|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{d} \exp(\theta_i f_i(x))$
loi de la famille "exponentielle"
$\theta$ : paramètres naturels
$\log Z(\theta) = \log \sum_x \prod_i e^{\theta_i p_i(x)}$ fonction de répartition

**Maximum de vraisemblance :** $x^1, ..., x^n$ observations iid

**But :** $\max_\theta \sum_{n=1}^{N} \log p(x_n|\theta)$
$\Leftrightarrow \max_\theta \sum_{n=1}^{N} \sum_{i=1}^{d} \theta_i f_i(x^n) - \log(\theta)$ fonction concave de $\theta$ (car $\log Z(\theta)$ convexe)
$\Rightarrow$ unique maximum global
$\log Z(\theta) = log \sum_x \prod_{i=1}^{d} e^{\theta_i f_i(x)}$

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \log Z(\theta) &= \frac{1}{Z(\theta)} \sum_x f_i(x) \prod_{i=1}^{d} e^{\theta_i f_i(x)} \\ &= \sum_x f_i(x) p(x|\theta) \\ &= \mathop{\mathbf{E}}_{p(x|\theta)} f_i(x) \end{aligned}$$

$$\begin{aligned} \frac{\partial l}{\partial \theta_i} &= \sum_{n=1}^{N} f_i(x^n) - N \frac{\partial}{\partial \theta_i} \log Z(\theta) \\ &= N[\mathop{\mathbf{E}}_{\hat{p}(x)} f_i(x) - \mathop{\mathbf{E}}_{p(x|\theta)} f_i(x)] \end{aligned}$$

**Théorème 9.4** *$\theta$ correspond en ML* $\Leftrightarrow \forall i, \mathbf{E}_{\hat{p}(x)} f_i(x) = \mathbf{E}_{p(x|\theta)} f_i(x)$

**Algorithme (montée de gradient)**

a) $\theta^0 = 0$

b) $\theta_i^{t+1} \leftarrow \theta_i^t + \epsilon_t N(\mathbf{E}_{\hat{p}(x)} f_i(x) - \mathbf{E}_{p(x|\theta^t)} f_i(x)$

$\epsilon_t$ : learning rate

$\mathbf{E}_{p(x|\theta^t)} f_i(x)$ : inférence

**Propriétés :**

1. $\epsilon_t = \frac{1}{t} \Rightarrow \theta^t$ converge vers $\theta_{ML}$
2. si $\epsilon_t$ est choisi par "line search" $\Rightarrow \theta_t$ converge vers $\theta_{ML}$

à l'instant t, $f(\epsilon) = l(\theta^t + \epsilon \nabla l(\theta^t))$

1. trouver $\epsilon^* = argmax f(\epsilon)$
2. "backtracking line search" $\Rightarrow \theta^t$ converge vers $\theta_{ML}$

Voir livre de Stephen Boyd and Lieven Vandenberghe (Convex Optimization, Cambridge University Press), disponible gratuitement en ligne.

## 9.5 Lien avec le maximum d'entropie

Dans cette section, on cherche à faire le lien entre l'approche précédente basée sur le maximum de vraisemblance et le maximum d'entropie. On considère le cas de variables aléatoires à valeurs dans un ensemble fini $\chi$ et des fonctions "features" $f_i : \chi \longrightarrow \Re$. Le but est alors de déterminer une loi p(x) vérifiant :

$$E_{p(x)} f_i(x) = \alpha_i \ \ pour\ des\ \alpha_i\ donns$$

$$p(x)\ est\ d'entropie\ maximale$$

Le problème est donc de trouver p satisfaisant :

$$-\sum p(x) \log p(x) = max_p(-\sum p(x) \log p(x)) \ , \tag{9.1}$$

$$avec\ les\ contraintes\ \sum p(x) = 1 \ ,\ p(x) \geq 0 \ ,\ et\ \forall i,\ \sum p(x) f_i(x) = \alpha_i$$

Il s'agit donc simplement d' un problème de minimisation de fonctionnelle convexe sous contraintes affines donc on peut passer par le Lagrangien :

$$L(p, \mu, \theta) = \sum_x p(x) \log p(x) - \mu(\sum_x p(x) - 1) - \sum_{i=1}^{d} \theta_i (\sum_x p(x) f_i(x) - \alpha_i)$$

Puis en calculant le gradient de L par rapport à p :

$$\frac{\partial L}{\partial p} = \log p(x) + 1 - \mu - \sum_{i=1}^{d} \theta_i f_i(x) = 0$$

On déduit qu'au maximum de (1) on a la relation :

$$\log p(x) = \mu - 1 + \sum_{i=1}^{d} \theta_i f_i(x)$$

D'où le résultat suivant

Proposition : La solution de (1) est de la forme

$$p(x|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{d} e^{\theta_i f_i(x)}$$

On retombe bien sur la forme exponentielle de p qu'on avait adoptée dans l'approche précèdente.
En supposant donnés $x_1, x_2, ..., x_n$, on doit avoir $E_{p(x)} f_i(x) = E_{\hat{p}(x)} f_i(x)$. Ainsi, d'après ce qui précède, dans le cas où $\forall i,\ \alpha_i = E_{\hat{p}(x)} f_i(x)$ le maximum de vraisemblance et le maximum d'entropie aboutissent au même résultat.

## 9.6 Inférence variationnelle

On revient au problème de l'inférence pour un modèle graphique

$$p(x) = \frac{1}{Z} \prod_{c \in C} \psi_c(x_c)$$

Les algorithmes vus jusqu'à présent ont d'une complexité en $O(e^{tw})$ où tw est la largeur arborescente (treewidth). Le principe de l'inférence variationnelle consiste à approcher la loi p(x) par une famille de lois $q(x|\mu)$, puis trouver $\mu^\star$ tel que p(x) et $q(x|\mu^\star)$ soient les plus proches possibles. On réalise alors l'infèrence sur la loi $q(x|\mu^\star)$ plutôt que sur p(x). Ceci n'a d'intérêt que si l'on peut trouver $\mu^\star$ facilement et bien sûr si l'infèrence sur $q(x|\mu^\star)$ est plus simple, autrement dit si la largeur arborescente du nouveau modèle graphique est plus faible.
Comment choisir $q(x|\mu)$ ?
Si p(x) se factorise dans un certain graphe G, on définit $q(x|\mu)$ comme une loi se factorisant dans un sous-graphe H de G i.e :

$$\{\mu,\ q(x|\mu)\} = L(H)$$

L'idée est de prendre un sous-arbre H de faible largeur arborescente. On doit ensuite trouver la loi $q(x|\mu)$ la plus proche de p. On en vient donc naturellement à utiliser la distance de Kullback-Lieber introduite au début de chapitre :

$$\mu^\star = argmin_{q \in L(H)} D(q||p)$$

On peut résoudre ce problème dans le cas plus simple dit de champ moyen. Les variables sont supposées binaires : $\forall i = 1..n,\ x_i \in \{0, 1\}$ et p(x) se mettant sous la forme :

$$p(x) = \frac{1}{Z} \prod_{i=1}^{n} e^{\theta_i x_i} \prod_{(i,j) \in E} e^{\theta_{ij} x_i x_j}$$

On va s'intéresser aux lois q se factorisant dans le graphe G privé de toutes ses arêtes, ce qui revient à prendre le modèle très simple de var indépendantes. On a alors :

$$q(x|\mu) \propto \prod_{i=1}^{n} \mu_i^{\delta(x_i=1)}(1-\mu_i)^{\delta(x_i=0)}$$

On cherche alors $min_\mu D(q(x|\mu)||p(x|\theta))$. On a :

$$\begin{aligned} D(q(x|\mu)||p(x|\theta)) &= E_{q(x|\mu)} \log \frac{q(x|\mu)}{p(x|\theta)} \\ &= -\sum_{i=1}^{n} H(\mu_i) - E_{q(x|\mu)}(\sum_{i=1}^{n} \theta_i x_i + \sum_{i\sim j} \theta_{ij} x_i x_j) \\ &- \log Z(\theta) \\ &= -\sum_{i=1}^{n} (\mu_i \log \mu_i - (1-\mu_i)\log(1-\mu_i)) - \log Z(\theta) \\ &- \sum_{i=1}^{n} \theta_i \mu_i - \sum_{i\sim j} \theta_{ij} \mu_i \mu_j \end{aligned}$$

la notation $i \sim j$ signifiant que i et j sont voisins dans le graphe G. En exprimant que la dérivée par rapport à $\mu_i$ est nulle au niveau du minimum, on obtient la condition nécessaire :

$$-\log \mu_i - 1 + \log(1-\mu_i) + 1 - \theta_i - \sum_{j\sim i} \theta_{ij}\mu_j = 0$$

$$\iff \mu_i = \sigma(\theta_i + \sum_{j\sim i} \theta_{ij}\mu_j)$$

C'est une équation intrinsèque par rapport aux $\mu_i$. On peut obtenir algorithmiquement le minimum en initialisant un $\mu^0$ arbitraire et en itèrant :

$$\forall i \ , \ \mu_i^{t+1} = \sigma(\theta_i + \sum_{j\sim i} \theta_{ij}\mu_j^t)$$

On peut montrer que l'algorithme converge vers un **minimum local**. Toutefois, en général, $\mu_i^\star \neq p(x_i = 1|\theta)$.