Fondements de l'apprentissage machine

Automne 2014

Roland Memisevic

Leçon 8

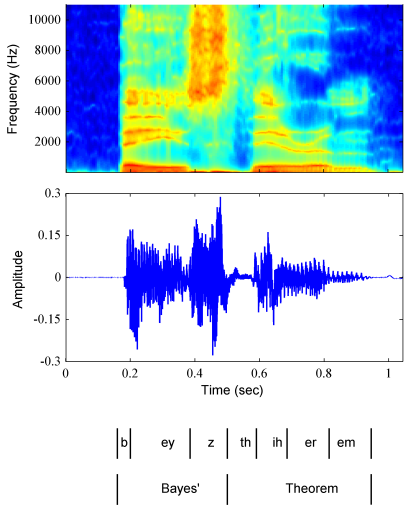
Plan

- Modélisation de séquences
- Markov Model (Modèle de Markov)
- Hidden Markov Model (Modèle de Markov caché)

Modélisation de séquences

- La supposition IID est pratique, mais pas correcte pour beaucoup de données du monde réel.
- L'abandon de la supposition d'indépendance conduirait à des modèles difficiles parce que la distribution deviendrait trop complexe.
- De manière moins stricte, on peut supposer que beaucoup de données ont la structure d'une séquence : (par exemple les signaux sonores/vocaux, la langue, les séries chronologiques financières, les brins d'ADN, les précipitations quotidiennes à un endroit, etc.)

Différentes représentations d'une séquence



Application de modèles de séquences

- Prédiction : Sachant n observations d'une séquence, quelle est la prochaine valeur $(n+1)$.
- Un modèle probabiliste d'une séquence peut être utilisé comme une distribution dans un classifeur génératif.
- Pré-traitement.

Modèle de Markov

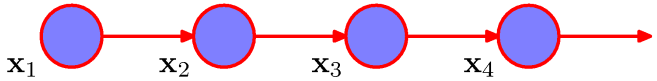
- Le modèle probabiliste le plus simple d'une séquence :

Modèle de Markov/Markov Model

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

- Le Modèle de Markov suppose que chaque observation dépend de son prédécesseur.
- Comme les observations forment une chaîne, un autre terme commun est "chaîne de Markov" (Markov Chain).
- Les probabilités conditionnelles s'appellent les probabilités de transition. $p(\mathbf{x}_1)$ s'appelle la probabilité initiale.

Modèle de Markov



- On peut représenter le modèle graphiquement.
- Dans le graphe, chaque flèche représente une *distribution conditionnelle* d'une variable.
- Représenter des indépendences en utilisant des graphes est un problème qui est étudié dans le domaine *graphical models*.
- La forme exacte des distributions conditionnelles dépend du type de données :

Paramétrisation

- Pour des données continues, il est courant d'utiliser les gaussiennes :

$$p(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}(\mathbf{x}_{n-1}), \boldsymbol{\Sigma}(\mathbf{x}_{n-1}))$$

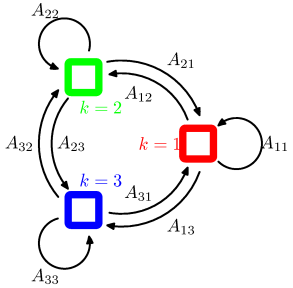
- Pour des données discrètes, il est courant d'utiliser les distributions discrètes :

$$p(\mathbf{x}_n = j | \mathbf{x}_{n-1} = i)$$

  Ces probabilités peuvent être représentées sous forme d'un tableau multidimensionnel, qu'on appelle souvent Conditional probability table (CPT) dans le domaine de *graphical models*.
- Dans le plupart des cas, $p(\mathbf{x}_n | \mathbf{x}_{n-1})$ est **le même** pour tous les n. Le modèle s'appelle Modèle de Markov homogène dans ces cas.

Modèle de Markov discret comme Machine d'état

Une autre manière de visualiser un modèle de Markov discret homogène sous forme de graphique :



- Chaque nœud représente une valeur possible pour $\mathbf{x}_n$, chaque arête représente une probabilité.
- Comme les Modèles de Markov discrets peuvent être considérés comme des machines d'état, $\mathbf{x}_n$ sont parfois appelés des états.

L'estimation de la CPT

- Avec $\mathbf{x}_n$ en codage orthogonal, les probabilités de transition peuvent être estimées comme :

$$A_{ij} = \frac{\sum_n x_{nj} x_{n-1,i}}{\sum_n x_{ni}}$$

- C'est facile à vérifier que cela représente l'estimation de *maximum de vraisemblance* pour le Modèle de Markov.
- Pour les probabilités initiales $\pi_i := p(\mathbf{x}_1 = i)$, on obtient

$$\pi_i = \frac{\sum_n x_{ni}}{N}$$

Modèles de Markov d'ordre supérieur

- En pratique, représenter des dépendances qui couvrent un seul pas de temps est souvent trop restrictif.
- Une solution est de conditionner sur plusieurs pas de temps précédents (*Modèles de Markov d'ordre supérieur*).
- Exemple d'un modèle d'ordre deux :

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) p(\mathbf{x}_2|\mathbf{x}_1) \prod_{n=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2})$$

- Malheureusement, le nombre de paramètres croît de façon exponentielle avec l'ordre du modèle.
- Une approche plus commune pour la modélisation de dépendances à longue portée est d'utiliser *des variables latentes* :

Hidden Markov Model (Modèle de Markov caché)

Hidden Markov Model (HMM) :

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{\mathbf{z}_1, \ldots, \mathbf{z}_N} p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N)$$

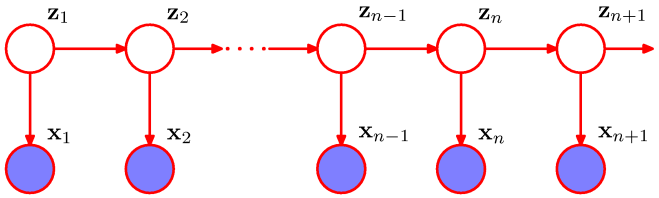
avec

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m)$$

où $\mathbf{x}_1, \ldots, \mathbf{x}_N$ sont des observations
et $\mathbf{z}_1, \ldots, \mathbf{z}_N$ sont des variables latentes

- Cela définit un Modèle de Markov qui est caché $p(\mathbf{z}_1, \ldots, \mathbf{z}_N) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1})$ où chaque état émet une observation stochastique en utilisant les probabilités d'émission $p(\mathbf{x}_n|\mathbf{z}_n)$

Hidden Markov Model (Modèle de Markov caché)



- Le HMM est un Modèle de Markov discret qu'on ne peut observer que de manière indirecte par ses émissions.
- Il est aussi comme un modèle de mélange dont les états sont couplés dans le temps.

Paramétrisation

- Les probabilités du Modèle de Markov caché sont :

$$A_{ij} = p(\mathbf{z}_n = j | \mathbf{z}_{n-1} = i)$$

et

$$\pi_i = p(\mathbf{z}_1 = i)$$

- La choix des probabilités d'émission dépend des données. discrète :

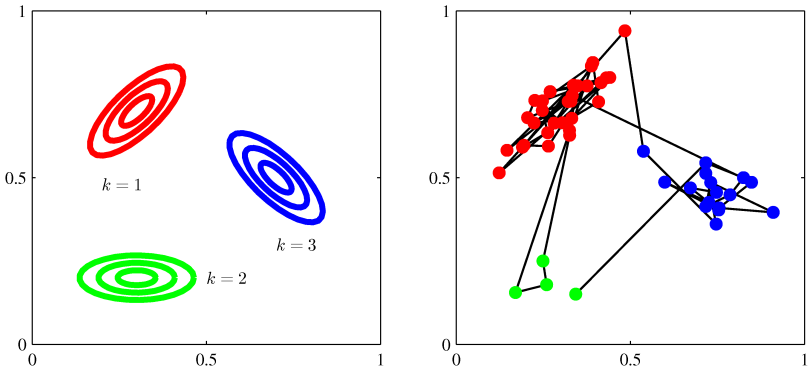
$$\phi_{il} = p(\mathbf{x}_n = l | \mathbf{z}_n = i)$$

gaussienne :

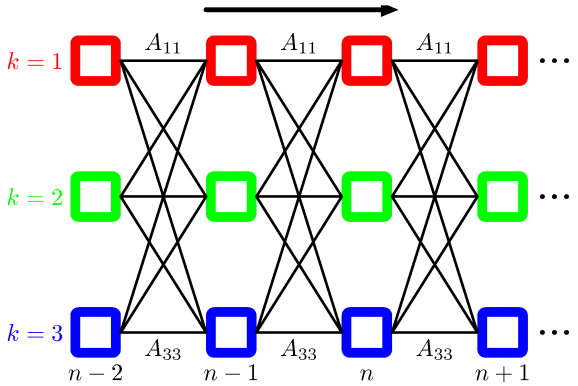
$$p(\mathbf{x}_n | \mathbf{z}_n = i) = \mathcal{N}(\mathbf{x}_n | \mu_i, \mathbf{\Sigma}_i) \quad \boldsymbol{\phi}_i := \mathrm{vec}(\boldsymbol{\mu}_i, \mathbf{\Sigma}_i)$$

...

HMM comme modèle de mélange gaussien avec dépendances entre les états



Représenter les transitions d'état comme un treillis



- Chaque chemin à travers ce réseau représente une séquence possible des états latents $\mathbf{z}_1, \ldots, \mathbf{z}_N$.
- Il y en a un nombre exponentiel !

Des calculs intraitables

- Empilez les observations (d'une seule séquence) et les variables cachées rangée par rangée dans les matrices $\mathbf{X}$ et $\mathbf{Z}$.
- Pour faire l'entraînement en utilisant l'algorithme EM, il faudra itérer :
  1. Calculez $p(\mathbf{Z}|\mathbf{X})$
  2. Optimisez l'*expected complete log-likelihood* $\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z})$
- Pour un modèle de K états et N observations, la variable $\mathbf{Z}$ peut prendre K^N valeurs.
- La probabilité des données, $\mathbf{X}$, selon le modèle est

$$p(\mathbf{X}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}) \quad = \sum_{\mathbf{z}_1, \ldots, \mathbf{z}_N} p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N)$$

C'est également une somme sur K^N termes.

Des indépendances

- Les règles de la probabilité permettent de dériver des indépendances et indépendances conditionnelles entre différentes variables dans le modèle, ce qui nous aide à trouver des expressions moins exigeantes.
- Les indépendances les plus importantes sont :

$$p(\mathbf{X}|\mathbf{z}_n) = p(\mathbf{x}_1, \ldots, \mathbf{x}_n|\mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N|\mathbf{z}_n)$$

et

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m) \text{ (définition du modèle)}$$

- Pour d'autres indépendances, voir Bishop, page 619.

Simplification des sommations

- La loi distributive permet de réécrire la marginalisation sur $\mathbf{Z}$ comme suit :

$$\begin{aligned}
& \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}) \\
= & \sum_{\mathbf{z}_1} \cdots \sum_{\mathbf{z}_N} p(\mathbf{X}, \mathbf{Z}) \\
= & \sum_{\mathbf{z}_1} \cdots \sum_{\mathbf{z}_N} p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m) \\
= & \sum_{\mathbf{z}_1} p(\mathbf{z}_1) p(\mathbf{x}_1|\mathbf{z}_1) \sum_{\mathbf{z}_2} p(\mathbf{z}_2|\mathbf{z}_1) p(\mathbf{x}_2|\mathbf{z}_2) \sum_{\mathbf{z}_3} p(\mathbf{z}_3|\mathbf{z}_2) \cdots
\end{aligned}$$

- Ceci est efficace si nous commençons le calcul de la droite !

Simplification de l'*expected complete log-likelihood*

- Le *complete log-likelihood* est un peu plus compliqué.
- Nous pouvons écrire :

$$\begin{aligned}
& \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z}) \\
= & \sum_{\mathbf{Z}} p(\mathbf{z}_1, \ldots, \mathbf{z}_N|\mathbf{X}) \log p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m) \\
= & \sum_{k=1}^{K} \gamma(z_{1k}) \log \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \log A_{jk} \\
& + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \log p(\mathbf{x}_n|\phi_k)
\end{aligned}$$

où $\gamma(\mathbf{z}_n) := p(\mathbf{z}_n|\mathbf{X})$, $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) := p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X})$ et ϕ sont les paramètres d'émission.
(il faut toujours calculer γ, ξ)

M-step

- En mettant la dérivée à zéro, on obtient :

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})}$$

$$A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j} z_{nl})}$$

M-step

- HMM gaussien :

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

- HMM discret :

$$\mu_{ik} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

L'algorithme forward-backward

- Nous avons

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{z}_n)p(\mathbf{z}_n)}{p(\mathbf{X})}$$

- Grâce à la propriété d'indépendance conditionnelle (diapo 18), nous pouvons écrire

$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N|\mathbf{z}_n)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

avec

$$\alpha(\mathbf{z}_n) := p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n)$$

et

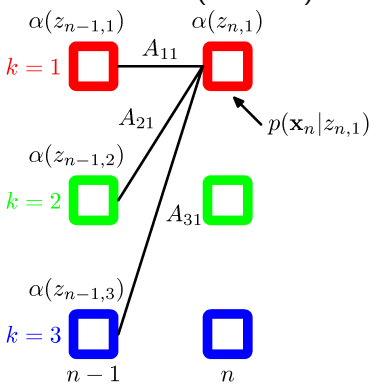
$$\beta(\mathbf{z}_n) := p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N|\mathbf{z}_n)$$

L'algorithme forward-backward

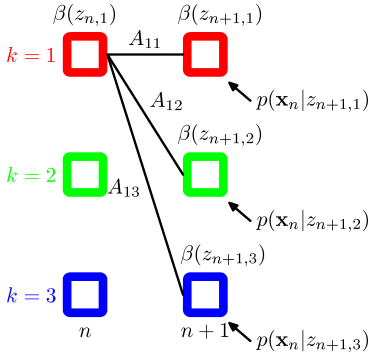
- $\alpha(\mathbf{z}_n)$ représente la probabilité de voir la séquence *jusqu'au temps* n *et* d'être dans l'état $\mathbf{z}_n$ au temps n.
- $\beta(\mathbf{z}_n)$ représente la probabilité de voir la séquence *à partir du temps* $n+1$, *sachant* l'état $\mathbf{z}_n$ au temps n.
- Nous pouvons calculer tous les α et β (et donc aussi γ et ξ) de manière récursive. Cela est connu comme algorithme forward-backward.
- C'est un cas de *programmation dynamique*.

L'algorithme forward-backward

récursion α (forward) :



récursion β (backward) :



L'algorithme forward-backward

récursion α

$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1})$$

$$\alpha(\mathbf{z}_1) = p(\mathbf{x}_1, \mathbf{z}_1) = \prod_{k=1}^{K} \left(\pi_k p(\mathbf{x}_1|\phi_k)\right)^{z_{1k}}$$

récursion β

$$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n)$$

$$\beta(\mathbf{z}_N) = 1$$

Calculer ξ et $p(\mathbf{X})$

$$\begin{aligned}
& \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) \\
= \; & p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}) \\
= \; & \frac{p(\mathbf{X}|\mathbf{z}_{n-1}, \mathbf{z}_n) p(\mathbf{z}_{n-1}, \mathbf{z}_n)}{p(\mathbf{X})} \\
= \; & \frac{p(\mathbf{x}_1, .., \mathbf{x}_{n-1}|\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{x}_{n+1}, .., \mathbf{x}_N|\mathbf{z}_n) p(\mathbf{z}_n|\mathbf{z}_{n-1}) p(\mathbf{z}_{n-1})}{p(\mathbf{X})} \\
= \; & \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{z}_n|\mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})}
\end{aligned}$$

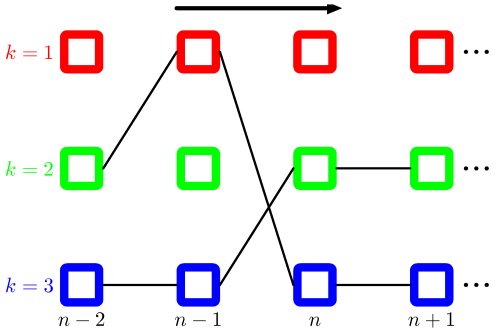
- Pour calculer $p(\mathbf{X})$, utilisez la définition efficace de $\gamma(\mathbf{z}_n)$ (diapo 23) :

$$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n) \beta(\mathbf{z}_n)$$

pour n'importe quelle n

Décodage Viterbi

- Le calcul de $\gamma(\mathbf{z}_n)$ dépend de la marginalisation de $p(\mathbf{Z}|\mathbf{X})$ sur (un nombre exponentiel de) $\mathbf{Z}$.
- En remplaçant la somme avec un $\max$, on peut calculer le *chemin le plus probable* vers les états cachés, sachant $\mathbf{X}$.
- Cela est connu comme l'algorithme Viterbi.



Kalman filter

- Le HMM est un modèle de mélange où les variables latentes sont couplées dans le temps.
- Il existe également des modèles avec des variables cachées continues.
- Si les variables latentes sont des gaussiennes et toutes les dépendances entre les variables sont linéaires, le modèle s'appelle *Kalman filter* ou *Linear Dynamical System* (LDS).