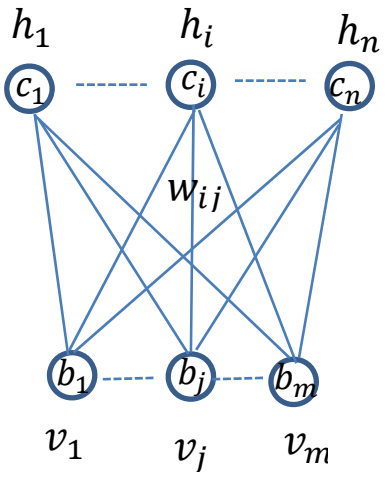
Examen Apprentissage Statistique

Master DAC (Informatique) et Spécialité Big Data (master de Math.)

10/02/2015

Durée 2 h, notes de cours autorisées

Les notations en **gras** désignent des vecteurs et les notations normales des scalaires.



Une machine de Boltzmann restreinte (RBM) est un réseau de neurones avec une couche d'unités visibles ***V*** et une couche d'unités cachées ***H*** (voir Figure). On va associer à ce réseau de neurones un modèle probabiliste. Les unités visibles et les unités cachées sont des variables aléatoires, notées respectivement $\boldsymbol{V} = (V_1, \dots, V_m)$ et $\boldsymbol{H} = (H_1, \dots, H_n)$. On considère par la suite des RBM binaires où V_i et H_i prennent des valeurs $v_j \in \{0,1\}$ et $h_i \in \{0,1\}$.

La distribution jointe de V et H est décrite par une distribution de Gibbs $p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}$ où $E(\boldsymbol{v}, \boldsymbol{h})$ est une fonction énergie qui a la forme suivante :

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\mathbf{h}^{\mathrm{T}}\mathrm{W}\mathbf{v} - \mathbf{c}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{b}^{T}\boldsymbol{h} = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

$w_{ij} \in R$ est le poids de la connexion entre la cellule visible V_j et la cellule cachée H_i, $b_j \in R$ et $v_i \in R$ sont des termes de biais associés à V_j et H_i, et Z est la fonction de partition[1] $Z = \sum_{\boldsymbol{v'},\boldsymbol{h'}} e^{-E(\boldsymbol{v'},\boldsymbol{h'})}$ qui sert à normaliser les probabilités de façon à ce qu'elles somment à 1.

Dans ce modèle, les variables cachées sont conditionnellement indépendantes connaissant les variables visibles et réciproquement :

$$p(\boldsymbol{h}|\boldsymbol{v}) = \prod_{i=1}^{n} p(h_i|\boldsymbol{v}) \text{ et } p(\boldsymbol{v}|\boldsymbol{h}) = \prod_{j=1}^{m} p(v_j|\boldsymbol{h})$$

[1] Ici la somme est calculée sur toutes les valeurs possibles des vecteurs ***v'*** et ***h'***. Si ils sont binaires, de taille respective n et m, il y aura 2^m valeurs possibles pour ***v'*** et 2^n pour ***h'***.

Une RBM va être entrainée à apprendre la distribution jointe $p(V,H)$. Pour cela on va maximiser la vraisemblance du modèle sur un ensemble d'exemples d'apprentissage, avec comme paramètres du modèle les w, b et c. Une fois entrainée, elle pourra servir à effectuer des inférences

1. Inférence

On suppose que la RBM a été entrainée sur un corpus d'apprentissage. Elle modélise alors une distribution $p(V,H)$ apprise sur ces données. Pour une donnée $\boldsymbol{v}$, on peut calculer la probabilité $p(\boldsymbol{v})$ que cette donnée ait été générée par le modèle appris.

Pour cela on a besoin de calculer la probabilité marginale $p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h})$ où la somme est prise sur toutes les valeurs possible de $\boldsymbol{h} \in \{0,1\}^n$.

1. On peut réécrire $p(\boldsymbol{v})$ sous la forme $p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}|\boldsymbol{h})p(\boldsymbol{h})$. A quel type de modèle est ce que cela vous fait penser ? Combien y-a-t-il de composantes et est-ce que ce calcul direct vous parait raisonnable en terme de complexité ?
2. On va regarder plus précisément comment se décompose $p(\boldsymbol{v})$. Montrer

$$p(\boldsymbol{v}) = \frac{1}{Z}\sum_{h_1=0}^{1} \dots \sum_{h_n=0}^{1} e^{\sum_{j=1}^{m} b_j v_j} \prod_{i=1}^{n} e^{h_i(c_i+\sum_{j=1}^{m} w_{ij}v_j)}$$

puis

$$p(\boldsymbol{v}) = \frac{1}{Z}\prod_{j=1}^{m} e^{b_j v_j} \prod_{i=1}^{n} (1 + e^{c_i+\sum_{j=1}^{m} w_{ij}v_j})$$

3. Quelle est la complexité du calcul du numérateur $\prod_{j=1}^{m} e^{b_j v_j} \prod_{i=1}^{n}(1 + e^{c_i+\sum_{j=1}^{m} w_{ij}v_j})$ et celle du dénominateur $Z = \sum_{\boldsymbol{v}',\boldsymbol{h}'} e^{-E(\boldsymbol{v}',\boldsymbol{h}')}$? Commenter.
4. Classification

Plusieurs méthodes sont possibles si on veut utiliser ce modèle pour faire de la classification.

4.1. on peut entrainer une RBM par classe et ensuite présenter une donnée test à chacune des RBM. Si on suppose que l'on a 2 classes, on notera $p_1(V)$et $p_2(V)$ les 2 distributions apprises.
- Comment peut-on effectuer la classification d'une donnée de test $\boldsymbol{x}$?
- Quelle est la complexité de ce classifieur et est-ce que cela vous parait réaliste ?

4.2. Une autre façon est de considérer quel la couche visible contient des cellules d'entrée $\boldsymbol{x}$ qui coderont les caractéristiques de la donnée et des cellules de classe $\boldsymbol{y}$ qui coderont les classes suivant un codage 1 parmi *k* si on a *k* classes.
- Illustrer cette RBM par un dessin
- Donner informellement une solution permettant après apprentissage de déterminer la classe la plus probable quand on présente un exemple de test[2].
- En utilisant la relation $p(y|\boldsymbol{x}) = \frac{\sum_{\boldsymbol{h}} p(\boldsymbol{x},y,\boldsymbol{h})}{\sum_{y',\boldsymbol{h}} p(\boldsymbol{x},y',\boldsymbol{h})}$ expliquer comment on peut faire la classification.
- Quelle est la complexité de ce classifieur et est-ce que cela vous parait réaliste ?

[2] Pour classer, il suffit de savoir quelle est la classe dont la probabilité est la plus élevée sans avoir besoin de connaitre cette probabilité.

2. Apprentissage

Neurones stochastiques

La RBM peut être interprétée comme un réseau de neurones stochastiques avec (résultat admis):

$$p(H_i = 1|\boldsymbol{v}) = \sigma(\sum_{j=1}^{m} w_{ij} v_j + c_i) \qquad \text{(i)}$$

et

$$p(\mathrm{V_j} = 1|\boldsymbol{h}) = \sigma(\sum_{i=1}^{n} w_{ij} \mathrm{h_i} + \mathrm{b_j}) \qquad \text{(ii)}$$

Où σ est la fonction logistique $\sigma(x) = \frac{1}{1+e^{-x}}$

Gradient de la log vraisemblance

Pour entrainer une RBM, on va maximiser la log vraisemblance du modèle sur les exemples d'apprentissage. On peut faire cela avec une montée de gradient. On note $\boldsymbol{\theta}$ les paramètres du modèle (ici les paramètres w, b, c).

Si on considère un unique exemple $\boldsymbol{v}^0$, la log vraisemblance s'écrit :

$$\log p(\boldsymbol{v}^0|\theta) = \log \frac{1}{Z} \sum_h e^{-E(\boldsymbol{v}^0, \boldsymbol{h})} = \log \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}^0, \boldsymbol{h})} - \log \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

1. Montrer

$$p(\boldsymbol{h}|\boldsymbol{v}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{h'} e^{-E(\boldsymbol{v},\boldsymbol{h}')}}$$

2. En utilisant ce résultat, montrer

$$\partial \log \frac{p(\boldsymbol{v}^0|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^0) \frac{\partial E(\boldsymbol{v}^0, \boldsymbol{h})}{\partial \theta} + \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \theta}$$

La première expression est l'espérance par rapport à la distribution conditionnelle des cellules cachées étant donné l'exemple $\boldsymbol{v}^0$**,** la seconde expression est l'espérance calculée par rapport à la distribution jointe $p(V, H)$ apprise par le modèle.

3. On va calculer ces dérivées pour le paramètre w_{ij}, montrer

$$\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^0) \frac{\partial E(\boldsymbol{v}^0,\boldsymbol{h})}{\partial \boldsymbol{w_{ij}}} = -\, p(H_i = 1|\boldsymbol{v}^0) v_j^0 \qquad \text{(iii)}$$

$$\sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{w_{ij}}} = -\sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1|\boldsymbol{v}) \mathrm{v_j} \qquad \text{(iv)}$$

Le gradient de la vraisemblance par rapport à w_{ij} en un point s'écrit donc

$$\partial \log \frac{p(\boldsymbol{v}^0|w_{ij})}{\partial \mathrm{w_{ij}}} = p(H_i = 1|\boldsymbol{v}^0) v_j^0 - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1|\boldsymbol{v}) \mathrm{v_j}$$

Calculer (iii) facile grace à (i). Par contre calculer (iv) nécessite une somme sur toutes les valeurs possibles de $\boldsymbol{v}$, ce qui est prohibitif. On va utiliser une approximation qui repose sur 2 idées. La première consiste à remplacer l'espérance (iv) par la valeur du gradient prise en un point $\boldsymbol{v}$**.** La seconde consiste à échantillonner ce point selon la distribution apprise par la RBM. On va utiliser pour cet échantillonnage la procédure suivante.

4. Algorithme : Apprentissage RBM

Input : RBM($V_1, .., V_m, H_1, \dots, H_n$),
Output : approximation du gradient Δw_{ij} pour $i = 1 \dots n, j = 1, \dots m$
Init $\Delta w_{ij} = 0$ pour $i = 1 \dots n, j = 1, \dots m$
Répeter
Choisir $\boldsymbol{x}$ dans l'ensemble d'apprentissage
$\boldsymbol{v}^0 = \boldsymbol{x}$
pour $i = 1 \dots n$ faire échantillonner $h_i^0 \sim p(h_i|\boldsymbol{v}^0)$
pour $j = 1 \dots m$ faire échantillonner $v_j^1 \sim p(v_j|\boldsymbol{h}^0)$
pour $i = 1 \dots n, j = 1 \dots m$ faire
$\Delta w_{ij} = p(H_i = 1|\boldsymbol{v}^0)v_j^0 - p(H_i = 1|\boldsymbol{v}^1)v_j^1$

- Expliquer comment on peut réaliser les deux étapes d'échantillonnage.
- A quelle famille d'algorithme de gradient est ce que cet algorithme s'apparente ?
- Donner la formule de gradient utilisée pour l'apprentissage.