Examen Apprentissage Statistique

Master Informatique UPMC – spécialité DAC

31-01-2017

Notes de cours autorisées

Skip Gram avec embeddings gaussiens

1. En ½ page, le principe du modèle skip gram pour les embeddings de mots.

Au lieu de représenter les mots par des vecteurs, on va les représenter par une distribution Gaussienne : à chaque mot on associe une moyenne et une matrice de variance-covariance. Le modèle est plus riche et permet de représenter des incertitudes liées à la représentativité des données. On note w un terme du dictionnaire, et $c(w)$ un contexte de w. Par la suite, ce contexte sera réduit à 1 terme uniquement. Au mot w_i on associera la représentation $z_i \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i, \Sigma_i)$, $z_i \in R^n$. Le modèle skip-gram emploie le produit scalaire pour mesurer la similarité de deux représentations vectorielles. Le produit scalaire sera remplacé ici par une similarité entre deux distributions. On considère par la suite deux alternatives pour cela.

2. Similarité symétrique : noyau produit de probabilités.

Pour deux distributions f et g, le noyau, qui étend le produit scalaire vectoriel, est défini par $K(f,g) = \int_{z \in R^n} f(\boldsymbol{z}) g(\boldsymbol{z}) d\boldsymbol{z}$. On note pour simplifier $K\left(\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i, \Sigma_i), \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_j, \Sigma_j\right)\right) = K(\mathcal{N}_i, \mathcal{N}_j)$. On emploiera comme similarité $S\left(\mathcal{N}_i, \mathcal{N}_j\right) = K(\mathcal{N}_i, \mathcal{N}_j)$. On a alors (résultat admis) :

$$\log K\left(\mathcal{N}_i, \mathcal{N}_j\right) = -\frac{1}{2}\log|\Sigma_i + \Sigma_j| - \frac{1}{2}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T \left(\Sigma_i + \Sigma_j\right)^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right) - \frac{n}{2}\log 2\pi$$

Où $|\Sigma|$ est le déterminant de Σ.

Calculer $\frac{\partial \log K\left(\mathcal{N}_i, \mathcal{N}_j\right)}{\partial \boldsymbol{\mu}_i}$ et $\frac{\partial \log K\left(\mathcal{N}_i, \mathcal{N}_j\right)}{\partial \Sigma_i}$ (pour la première on peut utiliser la dérivée d'un scalaire par rapport à un vecteur, pour la seconde, on passera par la dérivée de la fonction scalaire par rapport aux composantes Σ_{kl} de Σ).

3. Similarité asymétrique : Distance de Kulback-Leibler

La deuxième alternative est la distance de Kulback-Leibler entre deux distributions : $D_{KL}(\mathcal{N}_j || \mathcal{N}_i) = \int_z \left(\log \frac{\mathcal{N}_j(\boldsymbol{z})}{\mathcal{N}_i(\boldsymbol{z})}\right) \mathcal{N}_j(\boldsymbol{z}) d\boldsymbol{z}$, on emploiera comme similarité $S\left(\mathcal{N}_i, \mathcal{N}_j\right) = -D_{KL}(\mathcal{N}_j || \mathcal{N}_i)$

3.1 Montrer :

$$\int_z \left(\log \mathcal{N}_i(\boldsymbol{z})\right) \mathcal{N}_j(\boldsymbol{z}) d\boldsymbol{z} = -\frac{1}{2}\left(n \log 2\pi + \log|\Sigma_i| + Tr\left(\Sigma_i^{-1}\Sigma_j\right) + \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T \Sigma_i^{-1}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)\right)$$

3.2 En déduire :

$$D_{KL}\left(\mathcal{N}_j, \mathcal{N}_i\right) = \frac{1}{2}\left(Tr\left(\Sigma_i^{-1}\Sigma_j\right) + \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T \Sigma_i^{-1}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right) - n - \log \frac{|\Sigma_j|}{|\Sigma_i|}\right)$$

3.3 Calculer $\frac{\partial D_{KL}\left(\mathcal{N}_j, \mathcal{N}_i\right)}{\partial \mu_i}$, $\frac{\partial D_{KL}\left(\mathcal{N}_j, \mathcal{N}_i\right)}{\partial \Sigma_i}$, $\frac{\partial D_{KL}\left(\mathcal{N}_j, \mathcal{N}_i\right)}{\partial \Sigma_j}$

4. Algorithme d'apprentissage. Etant donné un terme w et un terme de contexte c, on va optimiser par gradient stochastique une fonction de coût qui a la forme suivante :

$L(w, c_p, c_n) = \max(0, 1 - S(w, c_p) + S(w, c_n))$ où c_p est un terme de contexte « positif », i.e. qui co-occurre avec w dans le corpus et c_n est un terme de contexte négatif. $S(,)$ dénote l'une des deux similarités introduites ci-dessus et par abus de notation $S(w, c)$ désigne la similarité des représentations correspondantes.

4.1 Expliquer cette fonction de coût.

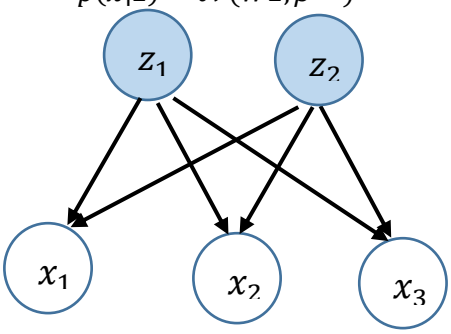
4.2 Donner un algorithme de gradient stochastique pour chacune des deux similarités, qui optimise cette fonction de coût.

Codage sparse binaire

On étudie un modèle génératif illustré sur la figure ci dessous. Une donnée $\boldsymbol{x} \in R^n$ est générée par le modèle suivant :

$$p(\boldsymbol{z}) = \prod_{i=1}^{p} r_i^{z_i}(1 - r_i)^{1-z_i}$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(W\boldsymbol{z}, \beta^{-1})$$



$z_i \in \{0,1\}$ suit une loi de Bernoulli de paramètre r_i, $\boldsymbol{z} \in \{0,1\}^p$, $W \in R^{n \times p}$, β^{-1} est une matrice diagonale $n \times n$ dite matrice de précision (c'est l'inverse de la matrice de covariance). On note que le modèle implique $p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{n} p(x_i|\boldsymbol{z})$ (indépendance conditionnelle) et $p(\boldsymbol{z}) = \prod_{i=1}^{p} p(z_i)$. On veut estimer les paramètres du modèle par maximum de vraisemblance. Pour cela on a besoin de calculer les probabilités a posteriori $p(\boldsymbol{z}|\boldsymbol{x})$ qui n'ont pas d'expression analytique simple. On va utiliser à la place une approximation variationnelle de champ moyen.

1. Optimiser la vraisemblance demande de connaitre les $p(\boldsymbol{z}|\boldsymbol{x})$. Pour s'en convaincre on va regarder la dérivée de $p(\boldsymbol{x})$ par rapport à un paramètre r_i. Montrer :

$$\frac{\partial}{\partial r_i} \log p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{x}) \frac{\frac{\partial}{\partial r_i} p(\boldsymbol{z})}{p(\boldsymbol{z})} = E_{z \sim p(Z|x)}[\frac{\partial}{\partial r_i} \log p(\boldsymbol{z})]$$

2. $p(\boldsymbol{z}|\boldsymbol{x})$ ne peut pas être calculé de façon analytique. L'approximation variationnelle consiste à l'approximer par une distribution simple $q(\boldsymbol{z}|\boldsymbol{x})$ qui permettra d'aboutir à une expression analytique, et à optimiser une borne de la log vraisemblance. Montrer :

$$\log p_\theta(\boldsymbol{x}) = V_L(\theta, \phi; \boldsymbol{x}) + \mathrm{D}_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$$

avec

$$\mathrm{V_L}(\theta, \phi; \mathbf{x}) = \mathrm{E_{q(z|x)}}[\log\ \mathrm{p_\theta}(\boldsymbol{x}, \boldsymbol{z}) - \log \mathrm{q_\phi}(\mathbf{z}|\mathbf{x})]$$

Où θ et ϕ désignent respectivement les paramètres du modèle $(\boldsymbol{r}, W, \beta^{-1})$ et de la distribution q (définis plus bas).

Comme $\mathrm{D}_{KL}(q_{\phi}(\boldsymbol{z}|\boldsymbol{x})||p_{\theta}(\boldsymbol{z}|\boldsymbol{x})\,) \geq 0$, $\mathrm{V}_{\mathrm{L}}(\theta, \phi; \mathbf{x})$ constitue une borne inférieure de $p_{\theta}(\boldsymbol{x})$ et c'est elle que l'on va optimiser.

Montrer que $\mathrm{V}_{\mathrm{L}}(\theta, \phi; \mathbf{x})$ se réécrit sous la forme

$$V_L(\theta, \phi; \boldsymbol{x}) = -D_{KL}(q_{\phi}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) + E_{q_{\phi}(Z|X)}[\log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})]$$

3. Pour obtenir une forme simple de $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$, on va faire l'hypothèse de champ moyen :

$$q_{\phi}(\boldsymbol{z}|\boldsymbol{x}) = \prod_{i=1}^{p} q(z_i|\boldsymbol{x})$$

Puisque $z_i \in \{0,1\}$, $q(z_i|\boldsymbol{x})$ sera modélisé par une distribution de Bernoulli de paramètre $q(z_i = 1|\boldsymbol{x}) = \hat{z}_i$

3.1 Montrer

$$\mathrm{V}_{\mathrm{L}}(\theta, \phi; \mathbf{x}) = E_{z\sim q(z|x)}\left[\sum_{i=1}^{p} \log p(z_i) + \sum_{i=1}^{n} \log p(x_i|\boldsymbol{z}) - \sum_{i=1}^{p} \log q(z_i|\boldsymbol{x})\right]$$

3.2 En introduisant les paramètres des distributions, montrer que cette expression se développe en :

$$\mathrm{V}_{\mathrm{L}}(\theta, \phi; \mathbf{x}) = \sum_{i=1}^{p} \hat{z}_i\,(\log r_i - \log \hat{z}_i) + (1 - \hat{z}_i)(\log(1 - r_i) - \log(1 - \hat{z}_i)) + \frac{1}{2}\sum_{i=1}^{n}\left[\log\frac{\beta_i}{2\pi} - \beta_i\left(x_i^2 - 2x_i W_{i,:}\hat{\mathbf{z}} + \sum_{j=1}^{p}\left(\sum_{k=1}^{p} W_{i,j}W_{i,k}\hat{z}_j\hat{z}_k\right)\right)\right]$$

Où $W_{i,:}$ représente la ligne i de la matrice W et $W_{i,j}$ l'élément i, j

Cette expression montre que la borne a une expression que l'on peut optimiser.

3.3 proposer un algorithme pour apprendre les paramètres θ et ϕ

3.4 A quoi sert ce modèle et comment l'utiliser ?

Formules utiles

A, B et X sont des matrices qui ont les bonnes dimensions pour se multiplier, x est un scalaire, $Tr(A)$ est la trace de A et $|A|$ son déterminant, $\boldsymbol{x}$ est un vecteur, x est un scalaire.

Trace de matrices

$Tr(A+B) = Tr(A) + Tr(B)$, $Tr(AB) = Tr(BA)$, $x^T x = Tr(xx^T)$, $x^T Ax = Tr(xAx^T) = Tr(Axx^T)$

Dérivation matricielle

La dérivée d'un scalaire par rapport à un vecteur est le vecteur des dérivées du scalaire par rapport à chacune des composantes (le gradient habituel) $\left[\frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_n}\right]$. De même la dérivée d'un scalaire par rapport à une matrice est la matrice des dérivées de ce scalaire par rapport à chaque élément de la matrice $\begin{pmatrix} \frac{\partial y}{\partial x_{11}} & \cdots & \frac{\partial y}{\partial x_{p1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1q}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{pmatrix}$.

La dérivée d'un vecteur par rapport à un scalaire est le vecteur des dérivées des composantes par rapport à ce scalaire $\left[\frac{\partial y_1}{\partial x}, \dots, \frac{\partial y_n}{\partial x}\right]^T$. La dérivée d'une matrice par rapport à un scalaire est la matrice des dérivées des composantes $\begin{pmatrix} \frac{\partial y_{11}}{\partial x} & \cdots & \frac{\partial y_{1n}}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_{m1}}{\partial x} & \cdots & \frac{\partial y_{mn}}{\partial x} \end{pmatrix}$.

Dérivées de scalaires

On suppose que le scalaire dépend des coefficients de la matrice ou l'inverse suivant les cas.

$\frac{\partial}{\partial A} Tr(AB) = B^T$, $\frac{\partial}{\partial A} \ln|A| = (A^{-1})^T = (A^T)^{-1}$, $\frac{\partial \ln|A|}{\partial x} = Tr(A^{-1}\frac{\partial A}{\partial x})$, $\frac{\partial Tr(A)}{\partial x} = Tr(\frac{\partial A}{\partial x})$, $\frac{\partial \boldsymbol{x}^T A \boldsymbol{x}}{\partial \boldsymbol{x}} = (A + A^T)\boldsymbol{x}$, $\frac{\partial \boldsymbol{a}^T \boldsymbol{x}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}^T \boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$, $\frac{\partial \boldsymbol{x}^T A^{-1} \boldsymbol{y}}{\partial \boldsymbol{A}} = -(A^{-1})^T \boldsymbol{x}\boldsymbol{y}^T (A^{-1})^T$, $\frac{\partial}{\partial A} Tr(X^T A^{-1} Y) = -(A^{-1} Y X^T A^{-1})^T$

Dérivées de matrices par rapport à un scalaire

On suppose que les coefficients de la matrice dépendent su scalaire.

$\frac{\partial A^{-1}}{\partial x} = -A^{-1}\frac{\partial A}{\partial x}A^{-1}$, $\frac{\partial AB}{\partial x} = \frac{\partial A}{\partial x}B + A\frac{\partial B}{\partial x}$

Espérance et trace

$E_x[A\boldsymbol{x}] = AE_x[\boldsymbol{x}]$, $E_x[Tr(A\boldsymbol{x})] = Tr(AE_x(\boldsymbol{x})]$