Méthodes probabilistes pour le TAL
M1 Linguistique Informatique — Marie Candito
Université Paris 7 — 2010/2011

Contrôle des connaissances – 18/03/11 CORRECTION PARTIELLE

1 Cours (5pts)

1.4 (2pts) Estimation par maximum de vraisemblance

Correction : On considère un corpus, plus généralement des données, comme le résultat d'une expérience, elle-même constituée de la réalisation de n épreuves.
Par exemple : un corpus de n mots peut être vu comme le résultat de n tirages de mots, ou bien comme le résultat de n tirages de bigrammes de mots etc… En effet, un modèle probabiliste décompose en général une probabilité complexe en probabilités plus élémentaires, en faisant notamment des hypothèses sur l'indépendance de certains évènements entre eux.
Par exemple, on peut décomposer l'événement « produire une phrase de x mots » en x tirages de mots, et le tirage d'un mot peut ne dépendre que du mot précédent (hypothèse de Markov d'ordre 1) ou des deux mots précédents (hyp. de Markov d'ordre 2) etc…
Ces probabilités plus élémentaires sont appelées les paramètres du modèle.

L'estimation par maximum de vraisemblance consiste à
- considérer la fonction *de vraisemblance* qui pour un corpus fixé (plus généralement des données fixées) prend comme argument des valeurs de paramètres, et leur associe la probabilité du corpus, calculée avec ces valeurs de paramètres
- les paramètres estimés par « maximum de vraisemblance » sont les valeurs qui rendent maximale la fonction de vraisemblance

Intuitivement, il s'agit de fixer les paramètres de telle sorte que ce que l'on connaît (le corpus) soit mathématiquement le plus probable (=ait la plus grande probabilité calculée avec ces paramètres).

Dans le cadre de la question posée,
- les données sont les 10000 phrases
- on les considère comme le résultat de 10000 tirages indépendants de phrases et réponses à la question : la phrase est-elle averbale ou pas. Admettons que l'on note A pour averbale, et V pour verbale, on considère donc une séquence de 10000 A ou V, contenant en tout 25 A
- la probabilité d'une telle séquence, si on note p la probabilité qu'une phrase soit averbale, est $p^{25}(1-p)^{10000-25}$
- ici le **paramètre** du modèle est la probabilité p qu'une phrase soit averbale
- le principe de l'estimation du maximum de vraisemblance consiste à choisir le p qui rende maximal la vraisemblance (notée L pour likelihood)
  - $L_{corpus}(p) = P_p(corpus) = p^{25}(1-p)^{10000-25}$
- Et on peut montrer (par exemple en cherchant le p qui rend nulle la dérivée du log de $L_{corpus}(p)$) que cet argmax est p = 25 / 10000, i.e. la **fréquence relative** de réalisation de l'événement dont p est la probabilité

2 Exercice (2,5pts)

On considère un corpus C, taggé, où l'on a mélangé deux corpus :
- C1 = un corpus de 10000 mots de phrases taggées par un humain,
- C2 = et un corpus de 90000 mots de phrases taggées par un tagger, dont la précision (accuracy) mesurée sur un autre corpus est de 97%.

D'après des expériences antérieures, on estime que l'humain a une probabilité de 0.01 de mal tagger un mot.

On choisit un mot au hasard dans le corpus. Quelle est la probabilité que le mot soit mal taggé ?
Correction : On note :
- H l'événement « le mot appartient à une phrase taggée par un humain »
- $\overline{H}$ et son complémentaire l'événement « le mot appartient à une phrase taggée par un tagger »
- E l'événement « le mot est mal taggé »

Et on peut calculer directement P(H) en utilisant l'équiprobabilité des évènements élémentaires (cf. choix d'une phrase au hasard) :

$$P(H) = \frac{|H|}{|C|} = \frac{10000}{(10000+90000)} = 0{,}1 \quad \text{et donc } P(\overline{H}) = 0{,}9$$

L'énoncé nous donne : $P(E \mid H) = 0{,}01$ et $P(E \mid \overline{H}) = 0{,}97$

Et on cherche **P(E)** => Par la formule des probabilités totales, on obtient :
(intuitivement : pour avoir une erreur on doit « passer » par un des deux évènements : le mot a été taggé par un humain ou bien il a été taggé par le tagger)

$$P(E) = P(E \mid H)P(H) + P(E \mid \overline{H})P(\overline{H}) = 0{,}01 \times 0{,}1 + 0{,}03 \times 0{,}9 = 0{,}028$$

On choisit un mot et il est mal taggé. Quelle est la probabilité qu'il provienne d'une phrase taggée par le tagger ?
Correction : On cherche $P(\overline{H} \mid E)$

que l'on obtient par Bayes : $P(\overline{H} \mid E) = \frac{P(E \mid \overline{H})P(\overline{H})}{P(E)} = \frac{0{,}03 \times 0{,}9}{0{,}028} \approx 0{,}9643$

3 Exercice (2,5pts)

Correction :
On considère l'expérience aléatoire E=« reconnaître 10 mots de suite », qui d'après l'énoncé est la réalisation de 10 épreuves indépendantes M=« reconnaître un mot ». On considère :
- la variable de Bernouilli Y définie pour l'expérience M, qui vaut 1 si le mot est bien reconnu et 0 sinon. On a d'après l'énoncé p=P(Y=1) = 0,6
- la variable X définie pour l'expérience E, qui compte le nombre de cas où Y vaut 1 : il s'agit d'une variable binomiale, de paramètre (10 ; 0,6)

On cherche la probabilité qu'un message soit compris, i.e.
$P(X > 7) = P(X = 8) + P(X = 9) + P(X = 10)$ sachant que pour une variable de paramètres (n ;p) on a $P(X = k) = C_n^k p^k (1-p)^{n-k}$. On obtient avec le logiciel R :

```
> k <- 8:10
> X <- dbinom(k,10,0.6)
> sum(X)
[1] 0.1672898
```

Soit environ 17% de chances pour que le message soit compris.

4 Classificateur naïf bayésien (5pts)

Correction : Les différentes classes sont les langues pour lesquelles on dispose d'un corpus de documents avec langue identifiée. Le principe général consiste à retourner la classe :

$$\hat{c} = \arg\max_c P(c \mid D) = \arg\max_c \frac{P(D \mid c)P(D)}{P(D)} = \arg\max_c P(D \mid c)P(c)$$

Avec un modèle « bag of letters », on décompose la proba de l'événement « D » en la proba conjointe sur toutes les lettres apparaissant dans D, et on considère en outre que chaque lettre est indépendante des autres (! ou disons que l'on considère qu'il est inutile de modéliser les dépendances entre lettres pour capturer l'appartenance à une langue), on obtient alors :

$$\hat{c} = \arg\max_{lg} \; P(\text{lg}) \prod_{i=1}^{N} P(let_i \mid \text{lg})^{C_D(let_i)} = \arg\max_{lg} \; \log(P(\text{lg})) + \log(\prod_{i=1}^{N} P(let_i \mid \text{lg})^{C_D(let_i)})$$

$$= \arg\max_{lg} \; \log(P(\text{lg})) + \sum_{i=1}^{N} C_D(let_i) \log(P(let_i \mid \text{lg}))$$

On estime P(let | lg) par fréquence relative : le nombre d'occurrences de la lettre let dans les documents de langue lg, divisé par le nombre total d'occurrences de lettres dans les documents de langue lg.
On estime P(lg) par le nombre de documents de langue lg divisé par le nombre total de documents.
D'où l'importance du corpus utilisé pour l'estimation : il doit être suffisamment grand pour que les estimations soit fiables, et la répartition des documents sur les différentes langues doit refléter la répartition attendue pour les documents dont la langue sera à identifier.

On pourrait aussi utiliser le nombre de lignes : avoir un seul gros corpus pour chaque langue, et estimer P(lg) comme le nombre de lignes de langue lg divisé par le nb total de lignes.
Ou encore, on peut vouloir faire un détecteur de langues qui définit les P(lg) comme **équiprobables** : on ne préjuge pas de différences de distribution des langues que l'on devra détecter. Les corpus d'apprentissage ne servent qu'à apprendre les P(lettre | langue).
Ci dessous la version du pseudo-code **en utilisant le nb de lignes pour estimer P(lg), et sans aucun lissage, ce qui n'est pas réaliste !** Pbs de lissage : voir cours

Pseudo-code estimation:
L = liste des langues « détectables »
C = liste des corpus (un corpus par langue de L)
_nb_lignes = dictionnaire vide #nbre de lignes pour chaque langue
_occ_obj_in_class = dictionnaire vide # nbre d'occurrences pour chaque couple lettre/langue

Pour chaque langue _lg dans L
  initialiser _nbocc_obj_in_class[lg] comme un dictionnaire vide
  initialiser _nblignes[lg] = 0
  _c = le corpus associé à lg
  Pour chaque _ligne de _c
    incrémenter _nblignes[_lg]
    Pour chaque lettre _lettre dans le corpus _c
      si _lettre n'a pas été déjà rencontrée dans _nbocc_obj_in_class [_lg]
        initialiser _nbocc_obj_in_class [_lg] [_lettre] à 1
      sinon
        incrémenter de 1 _nbocc_obj_in_class [_lg] [_lettre]

  # les P(let | lg)
  Pour chaque _lettre de _nbocc_obj_in_class [_lg]
    _log_prob_let_lg[_let][_lg]
      = log(_nbocc_obj_in_class[_lg])
        – log(somme de toutes les occurrences de lettres dans lg)

 # les P(lg) (après avoir parcouru toutes les lignes de tous les corpus)
Pour chaque langue _lg dans L
  _log_prob_langue = log(_nblignes[_lg]) – log (somme des _nblignes sur toutes les langues)

Pseudo-code prédiction:
T = le texte dont la langue est à prédire
Pour chaque lettre _lettre dans T
  Pour chaque langue _lg dans _log_prob_langue
    si lettre déjà vue dans le corpus associé à _lg
      ajouter _log_prob_let_lg[_lg][_lettre] à score[_lg]
    # rem : ici il faudrait gérer le cas d'une lettre non vue pour _lg, avec un lissage
    sinon
      score[_lg] = non défini et passer à la langue suivante
    ajouter _log_prob_langue à score[_lg]
Retourner la langue _meilleure_lg qui est donne le score maximal

5 Chaîne de Markov (5pts)

Exprimez la probabilité d'une phrase w1…wn dans le cadre d'une **chaîne de Markov d'ordre 1** (=modèle bigramme), sans lissage : donnez d'abord P(w1…wn) en fonction de probabilités plus élémentaires (avec toutes les étapes). Et donnez ensuite l'estimation de ces probabilités élémentaires.
Soit le mini corpus (on ignore la ponctuation et la casse) :
Les lions adorent les antilopes. Les antilopes adorent les herbes.Les herbes chatouillent les lions.

Correction :
On ignore les différences de casse, et la ponctuation. On pose w_0=début de phrase : un token fictif, avec probabilité 1 que w_0 débute toute phrase.
Et on ajoute un token f à chaque fin de phrase : on considère en fait non pas P($w_1 \ldots w_n$) mais

$P(w_1^n f \mid w_0)$

= (règle de multiplication) $P(w_1 \mid w_0)P(w_2 \mid w_0 w_1)P(w_3 \mid w_0 w_1 w_2)...P(w_n \mid w_0^{n-1})P(f \mid w_0^n)$

= (Markov ordre 1) $\left(\prod_{i=1}^{n} P(w_i \mid w_{i-1})\right) P(f \mid w_n)$

Ensuite les $P(w_i|w_{i-1})$ sont estimables par fréquence relative, ce qui maximise la vraisemblance du corpus : la probabilité conjointe de toutes les phrases du corpus, vues comme une série de bigrammes indépendants, est maximale quand les $P(w_i|w_{i-1})$ sont ainsi estimés.

$$\forall w_i \in V \cup \{f\}, \forall w_{i-1} \in V \cup \{d\} \quad P_{MLE}(w_i \mid w_{i-1}) = \frac{C(w_{i-1} w_i)}{\sum_{w \in V \cup \{f\}} C(w_{i-1} w)} = \frac{C(w_{i-1} w_i)}{C(w_{i-1})}$$

Appliquez le modèle bigramme pour calculer la proba de *Les herbes chatouillent les antilopes.*
P(Les herbes chatouillent les antilopes f | d) =
P(les|d) P(herbes|les)P(chatouillent|herbes)P(les|chatouillent)P(antilopes|les)P(f|antilopes)
*=1 * 2/6 * 1/2 * 1 /1 *2/6 *1/2*
=1/36

Expliquez l'utilité du lissage, et donnez le lissage par interpolation.
La probabilité d'une phrase calculée avec ce modèle va être nulle dès lors qu'elle contient un bigramme $mot_1 mot_2$ qui n'est pas dans le corpus d'estimation (cf. l'estimation de P(mot2 | mot1) sera nulle). D'où différentes techniques de **lissage** où l'on affecte une partie de la masse de probabilité des bigrammes connus (plus généralement évènements rencontrés dans le corpus d'estimation) aux bigrammes inconnus.
Lissage par interpolation : voir cours