Exercice à faire (sur papier)

Pour les deux premiers exercices, vous aurez besoin d'utiliser la règle de Bayes. Il découle de l'égalité suivante:

$$P(X = x, Y = y) = P(X = x)P(Y = y|X = x) = P(Y = y)P(X = x|Y = y).$$

De cette égalité, il est possible de déduire la règle de Bayes:

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{P(X = x)}.$$

De plus souvenons-nous que:

$$P(X = x) = \sum_{y \in \omega_Y} P(X = x, Y = y) = \sum_{y \in \omega_Y} P(X = x|Y = y)P(Y = y).$$

Critique de film

Pour aller voir un film, dois-je me baser sur l'avis de mon critique préféré (ou ami facebook)

- Notons A mon avis et B l'avis du critique sous forme de VA, par exemple A=1 si mon avis est positif, et A=0 sinon).
- Ce critique est fiable à 95% :

```
* 95% des films que j'ai aimé sont recommandés par le
  critique et
* 95% des films que je n'ai pas aimés sont déconseillés par le crit
ique
```

- J'aime seulement 1% des films qui sortent La critique de mon ami arrive, elle est positive **Quel est la probabilité que j'aime ce film ?**

Les cookies

Supposons 2 bols de cookies:

- le bol 1 contient 30 cookies à la vanille et 10 au chocolat
- le bol 2 en contient 20 de chaque Supposons que l'on choisisse un bol au hasard et à l'aveugle, et qu'on y pioche un cookie. Il est à la vanille. **Quelle est la probabilité qu'il vienne du bol 1 ?**

Estimation des paramètres pour une gaussienne

Soit X une VA continue que l'on suppose suivre la loi normale. On dispose d'un ensemble de N observations $x_{1:N}$ et nous souhaitons estimer les paramètres de la gaussienne à partir de ces données. Donner les valeurs des deux paramètres en fonction des données. Pour cela, il faut suivre les étapes données en cours:

1. écrire la vraisemblance
2. passer en log
3. dériver pour annuler et trouver les paramètres selon le max. de vraisemblance

TC4-TP1 (2016)

Remarque préalable

Les TP sont en python. Il est important d'en garder la trace convenablement: soit sous la forme d'un script ou d'un notebook. Néanmoins il faut réfléchir aux structures de données à mettre en oeuvre et penser à créer des fonctions, facilement ré-utilisables et bien paramètrées.

Les toolkits utiles sont en général installés dans: /partage/public/allauzen/python2.7/site-packages

Tout comme ce TP est accessible en local dans /partage/public/allauzen/TC-AIC/

python et ipython notebook

Les TPs seront fait en python et avec les notebook ipython. Donc pour bien commencer, regarder les tutoriels suivant:

- Pour python et numpy: http://cs231n.github.io/python-numpy-tutorial/ (http://cs231n.github.io/python-numpy-tutorial/)
- Pour ipython : http://cs231n.github.io/ipython-tutorial/ (http://cs231n.github.io/ipython-tutorial/)

Pour bien faire ce TP, le plus simple est d'utiliser sa version notebook, soit le fichier de départ TC4-tp1.ipynb

Pour lancer une session, créer un répertoire puis y copier le fichier *ipynb*. Placer vous dans ce répertoire executer: ipython notebook

Sélectionner le fichier que vous souhaiter à savoir *TC4-tp1.ipynb*. C'est parti.

Les données : le Brown corpus

L'objectif de cette première partie est de charger les données et de regarder (statistiquement) ce qu'elles contiennent. Le corpus utilisé contient du texte étiqueté en partie du discours, ou *POS tags* pour *Part Of Speech*: à chaque mot d'une phrase est associé une classe grammaticale. Ainsi une séquence de mot doit être associée à une séquence de tags (de même longueur). Le corpus est organisé comme un ensemble de phrases. À chaque mot est associé une cartégorie grammaticale et donc chaque phrase est une séquence de mots à laquelle est associée une séquence de catégorie.

Supposons qu'un mot est la réalisation d'une variable aléatoire notée *X* et que son étiquette est la réalisation de la variable aléatoire *Y*.

Pour le chargé, nous allons utiliser le toolkit *NLTK (Natural Language ToolKit)*

In [2]:

```
import nltk
data = nltk.corpus.brown.tagged_sents(tagset='universal')
print data[0]
```

```
[(u'The', u'DET'), (u'Fulton', u'NOUN'), (u'County', u'NOUN'), (u'Gr
and', u'ADJ'), (u'Jury', u'NOUN'), (u'said', u'VERB'), (u'Friday',
 u'NOUN'), (u'an', u'DET'), (u'investigation', u'NOUN'), (u'of', u'A
DP'), (u"Atlanta's", u'NOUN'), (u'recent', u'ADJ'), (u'primary', u'N
OUN'), (u'election', u'NOUN'), (u'produced', u'VERB'), (u'``',
 u'.'), (u'no', u'DET'), (u'evidence', u'NOUN'), (u"''", u'.'), (u't
hat', u'ADP'), (u'any', u'DET'), (u'irregularities', u'NOUN'), (u'to
ok', u'VERB'), (u'place', u'NOUN'), (u'.', u'.')]
```

Voici donc la première phrase du corpus, vue comme une séquence de couple *(mot,label)*. La première chose à faire est de diviser les données en 3 parties: apprentissage (*train*), développement (*dev*) et évaluation (*test*). Une répartition est en gros 80%, 10%, 10% respectivement.

S'échauffer en python

- De quel type est la variable data ? Et pour data[0] ?
- Combien y-a-t-il de phrases dans le corpus ?
- Combien y-a-t-il de mots (nombre d'occurence, par exemple il y a 25 mots dans la première phrase) ?
- Quel est la taille du vocabulaire (la liste des mots qui apparaissent au moins une fois dans le corpus) ? Pour répondre à cette question, une solution est de construire un set en parcourant le corpus.
- Quel est la liste des tags utilisés ? Combien y-en-t-il ?
- Construire une map qui stocke l'association ("mot",compte du mot) et estimer ces comptes sur tous le corpus.

In [11]:

```
# affichage du type
print type(data)
print type(data[0])
# ...
```

```
<class 'nltk.corpus.reader.util.ConcatenatedCorpusView'>
<type 'list'>
```

Questions

1. Comment faire une séparation aléatoire des données qui respectent (à peu près) la répartition proposée ?
2. Réaliser ce partage.
3. Sur les données d'apprentissage, contruire l'espace de réalisation concernant les étiquettes, calculer la distribution et la représenter avec matplotlib. Estimer et représenter la distribution mesurée sur les données de *dev* également.
4. Pour les mots, construire également l'espace de réalisation et calculer le compte de chaque mot. Puis représenter les comptes de comptes (soit combien de mots apparaissent une fois, deux fois, ...).
5. Représenter l'histogramme correspondant.

Probabilités

1. Nous allons limiter le vocabulaire aux mots apparaissant 10 fois ou plus. Les autres mots sont tous remplacés par la même forme *unk*
2. Estimer la distribution *P(Y|X)* sur les données d'apprentissage
3. Sur les données de test, effectuer la prédiction des etiquettes les plus probables et comparer votre prédiction aux etiquettes de référence et calculer le taux d'erreur.

In [ ]: