Université Paris-Sud

Master AIC - Apprentissage

Apprentissage : Estimation du nombre d'utilisateurs d'un shell

Julien Louis, Chloé Mercier, Perceval Wajsbürt

Janvier 2018

L'objectif de ce projet est d'estimer le nombre d'utilisateurs s'étant connectés à un shell UNIX à partir des logs de sessions (environ 2500).

1 Hypothèses

Nous allons traiter notre liste de session shells comme un corpus de textes classique et appliquer des méthodes qui fonctionnent dans le domaine du language naturel (NLP). En effet, une session partage plusieurs traits communs avec un texte de news par exemple :

- **ordre chronologique** : la sémantique est liée à l'ordre des commandes
- **co-occurence de termes** : certaines paires de termes sont plus ou moins fréquentes
- **sens d'une session** : exprimable dans l'espace des compétences (réseau/maths/...)
- **mots vides** : les flags par exemple ne portent pas de sens utile (comme 'à' en français)

Pour expliquer les données, nous ferons donc l'hypothèse du modèle probabiliste suivant :

- chaque utilisateur est représenté par un profil de compétences (ex : 70% réseau, 30% maths)
- chaque compétence est liée à un ensemble de mots (ex : réseau $\rightarrow$ *telnet*, maths $\rightarrow$ *matlab*)
- chaque session, appartenant à un utilisateur est un tirage suivant une loi de Dirichlet dans l'espace des compétences (réseau, maths, ...). Cette loi de Dirichlet a pour paramètres les valeurs d'un profil de session issu d'une loi normale autour du profil de l'utilisateur.

Pour déterminer les hyper-paramètres de notre modèle (nombre de compétences et le nombre d'utilisateurs), nous utiliserons les deux modèles de l'Allocation Latente de Dirichlet (LDA) et Mélange de Gaussiennes (GM).

2 Prétraitement des données

On découpe nos données par session (balises #BOF# and #EOF#), puis par mots pour suivre une approche Bag of Words (notons que cette approche fait perdre l'ordre chronologique).

Pour nettoyer les données, nous appliquons des connaissances spécifiques au problème Unix, et remplaçons certaines classes de termes par un token unique. Entre autres :

- les nombres à plus d'un chiffre sont trop rares pour indiquer un comportement (remplacés par _bignum_)
- les flags (options) n'ont de sens qu'au sein du programme dans lequel ils sont utilisés (_dashedarg_)
- les extensions doivent être détachées de leur fichier pour qu'elles interviennent dans le sens d'une session (remplacés par _extension_cpp_ par exemple)

Enfin, on procède au remplacement des fautes de frappe par un unique token (_typo_), en détectant les suites de mots proches (distance de Levenshtein) et dont le premier terme est très rare : nous pourrions ainsi distinguer les utilisateurs faisant beaucoup de fautes des autres.

3 Réduction en compétences

Modèle utilisé

Chaque session est représentée dans l'espace des mots par un vecteur continu de même taille que le vocabulaire. Ce nombre de dimensions (~ 400) est trop élevé pour rapprocher les sessions entre elles : en effet la granularité est trop forte et nous allons faire une projection dans l'espace réduit des compétences.

Suivant l'hypothèse probabiliste que les termes suivent une loi de Dirichlet, nous allons donc effectuer une LDA. Cependant, ce modèle requiert que l'on précise le nombre de dimensions T de l'espace d'arrivée : nous allons donc procéder à l'entraînement de plusieurs modèles dans un intervalle des valeurs de T.

Critère de sélection

On évalue les modèles en comparant les valeurs de perplexités en validation croisée. Les données sont divisés en 5 parties égales. A chaque itération, il y a en a 4 de train et 1 de validation. On séléctionne le nombre de cluster qui minimise la perplexité sur les données de test.

4 Estimation du nombre d'utilisateurs

Modèle utilisé

Selon notre hypothèse, les utilisateurs sont représentés par un profil dans l'espace des compétences, autour duquel sont tirées les sessions (aussi dans l'espace des compétences) par une loi normale. Il s'agit donc d'un mélange de

Gaussiennes, et nous allons comme pour la LDA entraîner plusieurs modèles avec différentes valeurs de nombre de clusters.

Critère de sélection

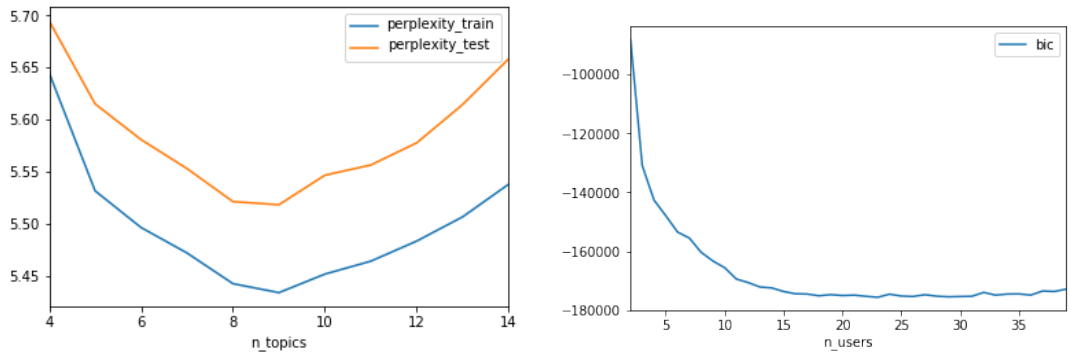
Nous allons utiliser des critères différents pour valider le nombre de cluster choisi :

- **log-vraisemblance** : à quel point les gaussiennes expliquent les observations
- **Calinsky-Harabaz** : un ratio entre les dispersions intra-cluster et inter-cluster, qui donne de bons scores aux clusters bien séparables (hyperplans)
- **silhouette** : utilise la distance entre les exemples au sein d'un cluster et avec le cluster différent le plus proche, et donne de bon scores aux clusters qui ne se recouvrent pas

Les deux derniers critères se concentrent sur la densité, homogénéité et séparation des clusters, ce qui n'est pas notre but ultime : en effet, il est probable que nos utilisateurs aient parfois des sessions très similaires, mais ce sont de bon indicateurs.

Le critère de log-vraisemblance, à maximiser, augmente avec le nombre de compétences choisies car on améliore la capacité du modèle à expliquer les comportements moins réguliers. Nous souhaitons donc détecter à partir de quand ajouter une dimension de compétence n'explique pas mieux nos données, c'est-à-dire observer un coude dans notre courbe. On peut contraindre le modèle en pénalisant le nombre de paramètres qu'il utilise, puisque que l'overfitting se produit pour de grandes quantités de paramètres : on utilise le **Bayesian Information Criterion (BIC)**.

5 Résultats



(a) Perplexité du train set et du test set

(b) BIC en fonction du nombre de clusters

FIGURE 1 – Résultats

On moyenne la perplexité obtenue par validation croisée sur 4 modèles LDA pour chaque nombre de compétences, et on observe que cette perplexité est minimale pour 9 compétences. Nous allons donc fixer ce paramètre pour la suite du modèle.

On sélectionne donc un modèle LDA dont les prédiction de répartitions des topics serviront pour la suite. On moyenne les valeurs de BIC sur les données de test et silhouette, Calhinsky-Harabaz sur l'ensemble des données pour plusieurs Mélanges de Gaussiennes de même nombre de cluster.

Selon le modèle LDA entrainé avec 9 compétences, on obtient différents nombres d'utilisateurs : entre 8 et 12. Les répartitions de topics prédites par la LDA ont en effet une forte variance et impactent fortement la vraisemblance obtenue par les GM. On observe alors un coude à une valeur de 8) 12 clusters en valeur BIC, quasi systématiquement appuyé par des pics sur les valeurs de Calhinsky Harabaz et de silhouette. Cela reste cohérent avec notre modèle. En conclusion, on retient les nombres de 8 à 12 pour les utilisateurs de la machine.

6 Limitation

Le modèle de gaussiennes impose qu'un utilisateur réalise des sessions avec des répartitions de topics proches. Un utilisateur compétent sur deux topics distincts A et B, et effectuant tantôt des sessions sur A, tantôt des sessions sur B, ne peut pas être modélisé par notre système. Ce cas est pourtant sans aucun doute très fréquent.