Analyse de sentiments

Frédéric Wantiez – Pierre Vigier

20 mai 2016

1 Description du problème

1.1 Intérêt

1.2 Formalisation

L'analyse de sentiments est un problème d'apprentissage supervisé. Notons V l'ensemble de tous les mots possibles et $V^* = \cup_{n\geq 0} V^n$ l'ensemble des textes sur ce vocabulaire. Soit $x_1, ..., x_N \in V^*$ des textes et $y_1, ..., y_N \in S$ le sentiment associé à chacun des textes. Ces sentiments peuvent être des valeurs dans $S = [0, 1]$ où 0 signifie que le texte est "très négatif" et 1, "très positif". Une variante plus simple est d'avoir les $(y_i)_{i\in 1,...,N}$ dans $S = 0, 1$ où 0 signifie "négatif" et 1, "positif". Notre objectif est de déterminer une fonction f telle que $\forall i \in 1, ..., N, y_i \approx f(x_i)$ et qui devra, de plus, bien généraliser sur des textes jamais vus auparavant. Dans le cas où S est discret, il s'agit d'un problème de classification. Dans le cas continu, il s'agit d'un problème de régression.

Nous allons essentiellement nous concentrer sur le problème de classification. Plusieurs types d'entrée et plusieurs types de classifieurs seront essayés sur le problème. La mesure de performance choisie est la précision $A(y_1, ..., y_N, \hat{y}_1, ..., \hat{y}_N) = \frac{\sum_{i=0}^{N} 1_{y_i=\hat{y}_i}}{N}$. L'objectif est de la maximiser. Elle nous permettra de comparer les performances des différents algorithmes.

2 Données

Il est assez facile de créer un ensemble de données pour entraîner nos algorithmes. En effet, il suffit de trouver un site où l'on peut commenter et mettre des notes sur des produits. La valeur numérique de la note correspond alors au sentiment dégagé par le texte. Cette configuration est présente sur les sites d'e-commerce comme Amazon ou sur les sites de critiques comme IMDB ou Rotten Potatoes.

Nous utilisons l'ensemble de données mis à disposition par Maas et al. [1]. Il s'agit d'un ensemble de 50 000 avis en anglais sur IMDB. À chaque avis est associé un label 0 ou 1 selon que l'avis est positif ou négatif.

2.1 Méthodologie

3 Premières tentatives

La première difficulté lorsque l'on travaille sur des textes est que leur longueur n'est pas fixe. La plupart des classifieurs nécessite des entrées de taille fixe. Il faut alors trouver une représentation de nos textes de taille fixe. Les représentations les plus populaires sont les sacs de mots et les vecteurs de mot. Dans chaque cas, nous allons décrire la représentation puis la tester en l'utilisant avec différents classifieurs. Ceci devrait nous donner une idée de l'efficacité de chaque représentation.

3.1 Sac de mots

Les sacs de mots est une représentation très simple. Numérotons les éléments de V, on a alors $V = w_1, ..., w_M$. Le sac de mots d'un texte $t \in V^*$ est un vecteur b de $\mathbb{R}^M$ tel que $\forall i \in 1, ..., M, b_i = card(\{j, t_j = w_i\})$. Autrement dit, la coordonnée i du sac de mots de t est le nombre d'occurrences du mot w_i dans t.

Les classifieurs bayésiens naïfs et la régression logistique ont comme avantage qu'il est possible d'interpréter le modèle après entraînement. Ainsi dans un classifieur bayésien naïf, on a accès à la probabilité $P(S = 1|w \in t)$. Avec la régression logistique, a chaque mot est associé un poids. On peut interpréter le poids associer à chaque mot comme le sentiment que porte le mot, seul. Dans les figures 1 et 2, on retrouve les mots jugés les plus négatifs et les plus négatifs par la régression logistique.

```
1. disappointment (-2.608113025883491)
2. waste (-2.359417022384978)
3. poorly (-2.3084246907186885)
4. baldwin (-2.19419686645024)
5. worst (-2.0717459858980107)
6. unwatchable (-2.0580492805825648)
7. obnoxious (-1.963499488620894)
8. lacks (-1.9554003844434584)
9. mst (-1.847654496703213)
10. forgettable (-1.8287902091116428)
```

FIGURE 1 – Les 10 mots ayant les poids associés les plus négatifs

```
1. refreshing (2.214710766134065)
2. vengeance (2.1735370232092186)
3. flawless (2.0122553306253668)
4. solo (1.7854059321112288)
5. voight (1.7756433401722442)
6. hooked (1.734909896519615)
7. wonderfully (1.7229447269614802)
8. existed (1.6963294241077245)
9. appreciated (1.5982250466066845)
10. stallone (1.5924422131212714)
```

FIGURE 2 – Les 10 mots ayant les poids associés les plus positifs

3.2 Vecteurs de mots

3.3 Conclusion

4 Prise en compte de l'ordre des mots

5 Conclusion

Références

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.