RECONNAISSANCE DES FORMES

Projet 4

Etudes et expérimentation de la reconnaissance des textes avec le SOM

réalisé par HO The Nhan et LE Viet Man - promotion 15

1. INTRODUCTION

Les cartes topologies de Kohonen (SOM) est une technique d'apprentissage non-supervisé. En utilisant le SOM, on peut réduire la dimension des données à partir d'une dimension très élevé en 2 ou 3 dimensions. Cette réduction nous permet d'interpréter facilement et instinctivement les résultats. Ce projet a donc pour but d'étudier la reconnaissance des textes avec la méthode des cartes topologiques de Kohonen (SOM) et suite de comparer avec trois méthodes déjà apprises le séparateur à vastes marges (SVM), le réseau rétro-propagation et la classification bayésienne naïve.

En particulier, nous avons d'abord implémenté deux applications : createVocabulaire - permettre de construire un vocabulaire à partir des textes, et exportArff - permettre de créer un fichier arff à partir des textes. Enfin, nous faisons nos expérimentations sur dix groupes de nouvelles trouvés sur http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

Dans ce rapport, nous présenterons d'abord les étapes du reconnaissance des textes dans la partie suivante dans laquelle nous parlerons du pré-traitement des documents et des cartes topologiques de Kohonen. Notre implémentation seront ensuite parlé et nous finirons avec notre expérimentations.

2. RECONNAISSANCE DES TEXTES

Dans la cadre du reconnaissance de textes, les caractéristiques que l'on doit extraire, sont des mots dans les textes. Donc, à partir de textes, on doit d'abord construire un vocabulaire où chaque mots est un caractéristique. Alors, chaque document sera présenté par un vecteur de caractéristiques dont le nombre d'élément est égale à celui dans le vocabulaire. Il s'agit de chaque élément du vecteur est un mot dans le vocabulaire. Ce vecteur sera utilisé dans la classification et ainsi que dans le reconnaissance de textes. Ensuite, on fait la classification des documents en utilisant une technique d'apprentissage. Enfin, après avoir eu des classifications, on peut faire le reconnaissance.

Cette partie va présenter la méthode de traiter des documents pour construire un vocabulaire et ainsi que d'extraire des vecteurs à partir des documents. Ensuite, elle va présenter la technique des cartes topologiques de Kohonen.

2.1. Pré-traitement des documents

Le pré-traitement des données est la phase très important et essentiel dans un classification des documents. La première partie de l'extraction de caractéristiques est pré-traitement du lexique et y compris l'élimination des mots vides (stop words en anglais), la réduction au radical des mots et le calcul des poids des mots. Nous avons utilisé la méthode présentée par [1].

a. Liste d'arrêt

Il s'agit de la première étape de pré-traitement qui va générer une liste de termes qui décrit le document de manière satisfaisante. Le document est analysé à travers pour trouver la liste de tous les mots. L'étape suivante dans cette étape est de réduire la taille de la liste créée par le processus d'analyse, généralement en utilisant des méthodes de l'élimination des mots d'arrêt et de la réduction au radical. La suppression des mots d'arrêt effacera de 20% à 30% du nombre total de mots alors que le processus de réduction au radical réduira le nombre de termes dans le document. Tous les deux processus contribue à améliorer l'efficacité et l'efficience du traitement de textes car ils réduisent la taille du fichier d'indexation.

Les mots d'arrêt sont effacés de chacun de documents en comparant avec la liste d'arrêt. Ce processus réduit le nombre de mots dans le document de manière significative depuis ces mots d'arrêt ne sont pas significatifs pour les mots clés de recherche. Les mots d'arrêt peut être une liste prédéfinie de mots ou qu'ils peuvent compter sur le contexte du corpus. Nous avons utilisé une liste prédéfinie de 725 mots d'arrêt.

b. Réduction au radical

L'étape suivante dans la première phase après l'élimination des mots d'arrêt est la réduction au radical. C'est un processus de normalisation linguistique dans laquelle les formes variantes d'un mot est réduit à une forme commune. Par exemple : le mot, connect a des formes diverses telles que connect, connection, connective, connected, etc. Ce processus réduira toutes ces formes de mots à un seul mot normalisé connect. Nous avons utilisé l'algorithme Porter's English stemmer pour réduire les mots dans chaque document.

c. Représentation de document

Un document est représenté par un ensemble de mots clés ou de termes extraits du document. La collection ou l'union de tous les ensemble de termes est l'ensemble des termes qui représente toute la collection et définit un « espace » de sorte que chaque terme distinct représente une dimension de cet espace. Puisque chaque document est représenté comme un ensemble de termes, l'espace est appelé « l'espace de documents ».

Une matrice de terme-document peut être encodé comme une collection de n documents et m termes. Un élément dans la matrice correspond au poids d'un terme dans le document. La valeur zéro signifie que le terme n'a aucune signification dans le document ou simplement il n'existe pas dans le document. Donc, la collection de tous les documents peut être considéré comme une matrice A de m x n caractéristiques (avec m est le nombre de documents) où l'élément a_{ij} représente la fréquence d'occurrence de terme j dans le document j. Cette façon de représentation de document est appelé la méthode le fréquence de termes. Toutefois, les termes qui ont une grande fréquence ne sont pas nécessaires car ils ne sont pas plus importants ou ont un supérieur discrimination. On peut donc vouloir coder le poids de la matière à l'égard du contexte local, le document ou le corpus. La pondération de termes la plus populaire est la fréquence document inverse, où la fréquence du terme est pesé par rapport au nombre total de fois où le terme apparaît dans le corpus. Il existe une

extension de cette désigné la fréquence fréquence document terme inverse (tf-idf). La formulation de tf-idf est donnée comme suivante :

$$W_{ij} = tf_{i,j} * \log(N / df_i)$$

où W_{ij} est le poids du terme i in document j, $tf_{i,j}$ est le nombre d'occurrence du terme i in document j, N est le nombre total de document et df_i est le nombre de documents contenant le terme i.

d. Réduction de dimensions

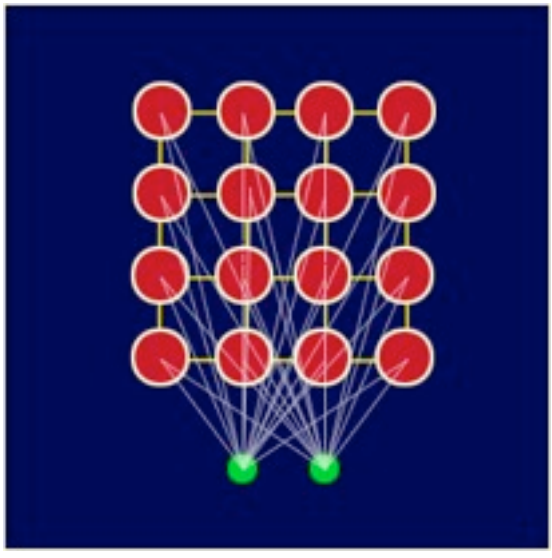
L'espace dans lequel le document a été réduit comprend typiquement des milliers de dimensions ou plus. Compte tenu de la collecte de documents ainsi que la distance matrice associée, nous aimerions trouver un endroit pratique à faible dimension pour effectuer une analyse ultérieure. Cela facilitera certainement le cluster ou la classification. En réduisant la dimension, on peut supprimer le bruit des données et d'appliquer mieux nos méthodes statistiques d'exploration de données pour découvrir des relations subtiles qui peuvent exister entre les documents. Donc, nous avons utilisé la méthode de l'analyse en composantes principales qui utilise les vecteurs propres à partir de matrices covariance ou corrélation pour réduire la dimension. PCA est utilisé pour la réduction de la dimension dans un ensemble de données en récupérant les caractéristiques de l'ensemble de données qui contribue le plus de sa variance, en restant des composantes principales d'ordre inférieur et en ignorant ceux d'ordre supérieur. Ces éléments d'ordre inférieur contiennent souvent les "plus importants" aspect des données.

2.2. Cartes topologiques de Kohonen (SOM)

La méthode des cartes topologiques de Kohonen est introduite pour la première fois par T.Kohonen en 1981. Les cartes topologiques auto-organisatrices font partie de la famille des modèles à « apprentissage non supervisé » par opposition aux perceptions multicouches, c'est-à-dire dans la première approche, ils seront utilisés dans un but descriptif. Les données à analyser sont des observations dont on cherche à comprendre la structure. Il n'y a pas de but précis à atteindre, ni de réponse souhaitée.

a. Architecture du réseau de Kohonen

Le réseau SOM le plus utilisé est le SOM de deux dimensions. Ce réseau est crée à partir d'un treillis des nœuds, chaque nœud est connecté totalement à la couche entrée. La figure à côté présente un réseau Kohonen de 4x4 nœuds.



Chaque nœud a une position (des coordonnées x et y) et contient un vecteur de poids de même dimensions que les vecteurs entrés. C'est-à-dire, si chaque des données d'apprentissage sont les vecteurs V de n dimensions $V_1, V_2, ..., V_n$; alors, chaque nœud correspond à un vecteur

de poids W de n dimensions $W_1, W_2, ..., W_n$.

b. Apprentissage

L'apprentissage utilisé par le réseau Kohonen est l'apprentissage compétitif. Pour chaque exemple d'apprentissage, ses distances Euclidiens vers tous les vecteurs de poids sont calculées. Le neurone qui a le vecteur de poids le plus similaire au vecteur d'entrée est appelé le gagnant ou Best Matching Unit (BMU). Chaque fois qu'on trouve un neurone gagnant, les poids de ce neurone et des neurones voisins sont ajustés vers le vecteur entré. La formule pour mettre à jours un neurone avec le vecteur de poids W(t) est :

$$W(t + 1) = W(t) + \Theta (v, t)\, \alpha(t)(D(t) - W(t))$$

Avec α(t) est le coefficient d'apprentissage décroissant invariablement et D(t) est le vecteur entré. La fonction de voisinage Θ(v, t) dépend la distance entre le gagnant et le neurone v.

L'apprentissage par un réseau de Kohonen suit les étapes suivantes :

- Les poids de chaque nœud de la couche sortie sont initialisés pour que 0<w<1. On peut les initialise aléatoirement ou utilisant les stratégies d'initialisation différentes.
- Choisir un vecteur dans l'ensemble d'entraînement pour présenter comme vecteur entré.
- Calculer la distance entre le vecteur de poids de chaque nœud et le vecteur entré. Le nœud gagnant est le nœud avec la distance minimal, c'est-à-dire le vecteur de poids est le plus similaire au vecteur entré. Une des méthodes pour calculer la distance est la distance Euclidien.
- Le rayon du voisinage du nœud gagnant est calculé. Au début, ce rayon est souvent large, il est diminué après chaque pas. Un nœud quelconque qui est dans ce rayon par rapport au vecteur gagnant est considéré comme un voisin du nœud gagnant.
- Les poids de chaque nœud voisin sont ajustés pour qu'il est plus similaire le vecteur entré.
- Répéter le pas 2 pour N itérations.

c. Classification

Pour classifier par le réseau Kohonen, on suit les étapes suivantes :

- Tout d'abord, on diviser l'ensemble de données en deux ensembles, un pour l'apprentissage et l'autre pour faire le test.
- Initialiser le réseau Kohonen (la taille du réseau ou bien le nombre des neurones, les poids de chaque neurone).
- Faire l'apprentissage sur l'ensemble d'entraînement (sans tenir compte au champ de classe) pour ajuster les poids des neurones.
- Utilisant encore une fois l'ensemble d'entraînement mais tenir compte au champ de classe pour faire l'étiquetage des neurones. Pour chaque exemple de données, on calcule le nœud

gagnant, et attribuer la valeur du champ de classe comme étiquette de ce nœud, si un nœud est correspond à plusieurs étiquette, on garde l'étiquette correspond le plus nombre de fois.

- En fin, on classifier pour les données dans l'ensemble de test. Pour chaque ensemble de données, on calcule le nœud gagnant et l'étiquette de ce nœud est la classe de cet ensemble.

3. IMPLEMENTATION

Nous avons implémenté deux programmes : CreateVocabulaire et exportArff. Tous les deux programmes est sans interface.

CreateVocabulaire sert à créer un vocabulaire à partir d'un répertoire des emails. L'utilisateur donne un chemin du répertoire qui stocke des emails et un fichier de liste d'arrêts. A la sortie, on a un fichier qui stocke des mots du vocabulaire. Ce programme possède la commande suivante :

```
./CreateVocabulaire [option] <path>
```

où :

<path> : le chemin du répertoire des emails

[option] :

- h : l'information de l'aide
- s <filename> : le fichier du vocabulaire
- l <filename> : le fichier qui stocke la liste d'arrête.

Exemple : créer un vocabulaire à partir des emails dans la base d'apprentissage :

```
./CreateVocabulaire -l stoplist /10newsgroup/train
```

exportArff sert à exporter les données en la forme de ARFF de Weka. Chaque ligne dans le fichier ARFF est un vecteur de caractéristiques d'un email. L'utilisateur donne un chemin du répertoire des emails et un vocabulaire. A la sortie, on a un fichier ARFF de Weka. Ce programme possède la commande suivante :

```
./exportArff [option] <path>
```

où :

<path> : le chemin du répertoire des emails

[option] :

- h : l'information de l'aide
- v <filename> : le fichier du vocabulaire
- s <filename> : le fichier arff

Exemple : créer un fichier Arff d'apprentissage à partir des emails dans la base d'apprentissage :

```
./exportArff -v voca /10newsgroup/train
```

créer un fichier Arff de test à partir des emails dans la base de test :

```
./exportArff -v voca /10newsgroup/test
```

Nous avons utilisé la librairie QtCreator pour programmer ces deux programmes. Ensuite, nous avons utilisé la librairie OpenCV pour programmer l'étape de réduction de dimension en utilisant la méthode ACP. Pour faire la classification, nous avons utilisé le logiciel Weka.

4. EXPERIMENTATION

Dans ce projet, la base de données testée est dix groupes de nouvelles trouvées sur http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups. On divise la base en deux bases, la base d'apprentissage et la base de test.

1. comp.graphics
2. comp.sys.ibm.pc.hardware
3. rec.autos
4. rec.sport.baseball
5. rec.sport.hockey
6. sci.electronics
7. sci.med
8. sci.space
9. talk.politics.misc
10. talk.religion.misc

Nous avons testé avec trois base de vocabulaires que nous avons crée à partir de 10 bases au-dessus :

- **Premier** : comprend 2118 mots - nous avons lassé la plus part des mots dans 10 base
- **Deuxième** : comprend 1172 mots - nous avons effacé plusieurs mots dans la base « stop words »
- **Troisième** : comprend 545 mots - nous avons effacé plusieurs mots dans la base « stop words » et effacé par la main les mots qui n'ont pas le sens.

Chaque email dans ces bases est un individu. Chaque individu a un vecteur de descripteur. Le nombre d'attributs de ce descripteur est le nombre de mots dans la base de vocabulaires.

Après avoir construit la base de descripteurs sous la forme ARFF, nous avons testé l'algorithme SOM du WEKA avec ces bases.

L'algorithme SOM du WEKA possède des paramètres suivants. Si l'on veut augmenter les résultats de classification, on doit choisir les bons paramètres.

- *Rate* : rate d'apprentissage
- *WH* : taille du carte KOHONEN

- *Iteration* : l'itération de l'algorithme
- *Vote* : choisir les voisins aléatoirement ou trié.
- *Topologie* : Topologie de chaque nœud dans carte (rectangulaire ou hexagonale)
- *Number of neighbour* : nombre de voisins dans l'algorithme KOHONEN
- *Seed* : coefficient de la fonction d'apprentissage

Nos résultats obtenus sont suivants :

Rate	HW	Iteration	Vote	Topo	Neighbour	Seed	Result
0.3	8	1440	FAUX	hexa	8	1	20.6%
0.5	-	-	-	-	-	-	26.9%
0.7	-	-	-	-	-	-	22.5%
-	-	-	VRAI	-	-	-	26.9%
-	15	-	-	-	-	-	34.8%
-	30	-	-	-	-	-	37.5%
-	-	150	-	-	2	-	44.6%
-	-	100	-	-	-	-	45.2%
0.5	**35**	**100**	**VRAI**	**hexa**	**2**	**32**	**54.83%**

Les résultats avec le premier vocabulaire

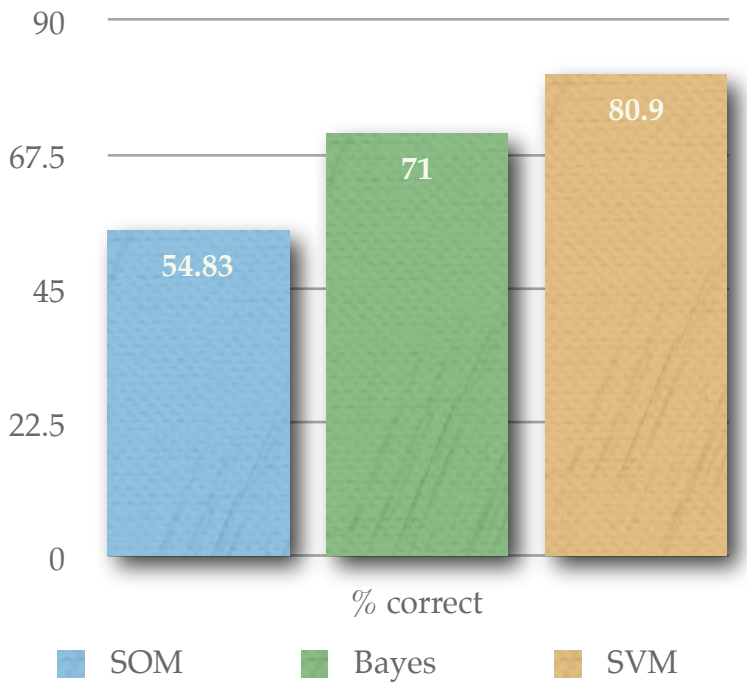
Rate	HW	Iteration	Vote	Topo	Neighbour	Seed	Result
0.5	30	100	VRAI	hexa	2	1	46.1%
-	-	-	-	rec	-	-	45.5%
-	-	-	-	-	4	-	28%
-	-	-	-	hexa	8	-	25%
0.3	-	-	-	-	2	-	46%
0.5	35	35	-	-	-	-	48%
-	-	-	-	-	-	16	50.8%

Rate	HW	Iteration	Vote	Topo	Neighbour	Seed	Result
0.5	**35**	**35**	**VRAI**	**hexa**	**2**	**32**	**53%**

Les résultats avec le deuxième vocabulaire

Rate	HW	Iteration	Vote	Topo	Neighbour	Seed	Result
0.3	8	1440	FAUX	hexa	8	1	26.47%
-	30	-	-	-	-	-	39%
0.5	**-**	**-**	**-**	**-**	**2**	**-**	**40.33%**
1	-	-	-	-	-	-	38%
0.7	-	-	-	-	-	-	38.8%
-	40	-	-	-	-	-	37.7%

Les résultats avec le deuxième vocabulaire



La comparaison entre trois méthodes : SOM, Bayes et SVM

Nos résultats sont un peu similaires avec le résultat de l'article, mais encore trop bas. Notre meilleur résultat obtenu est 54.83% avec le vocabulaire de 2118 mots. Cela se passe parce que il y a des faux dans l'algorithme Porter stemming. Cet algorithme efface toujours les mots vrais, tels que eye -> ey, apple -> appl. Nous trouvons que les mauvaises classes sont hockey, politique, médecin et

ibm.hardware. Nous pouvons augmenter le résultat en construisant un vocabulaire où contient des mots plus appropriés avec des groupes de test.

De plus, nous trouvons que les meilleurs paramètres sont Rate = 0.5, la taille de la carte est 10-15, la topologie est hexagonale et le nombre de voisins est 2 ou 4.

5. CONCLUSION

Les cartes topologies de Kohonen (SOM) est une bonne technique qui une compromis entre la visualisation et la classification. Selon mes tests, il semble que cet algorithme n'est pas convenable avec les données de test. La méthode Bayésinne et la méthode SVM donne les meilleurs résultats.

6. REFERENCES

1. ChandraShekar B.H., Shoba G., *Classification Of Documents Using Kohonen's Self-Organizing Map*, International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009