T.P. n° 2 - PRM2

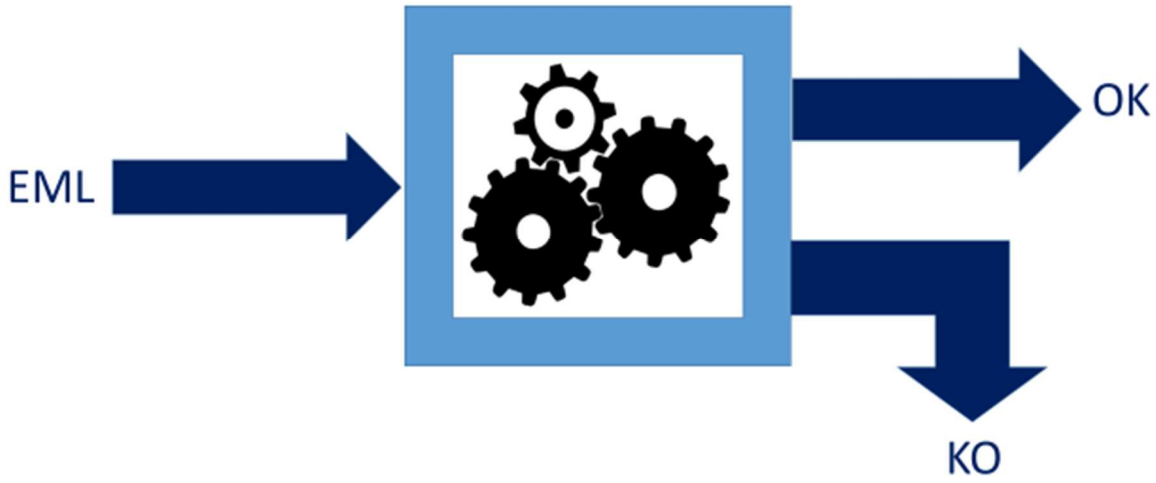
Automate de classement

1 – Introduction

Les grandes entreprises (et les administrations) reçoivent tous les jours de grandes quantités de courriers électroniques. Par exemple, les DRH doivent traiter les candidatures spontanées ; les services commerciaux doivent traiter les demandes de devis ou les bons de commandes ; les S.A.V. doivent traiter les réclamations ; les assureurs doivent traiter les déclarations de sinistres ; etc. Même si, *in fine*, ce sont des opérateurs humains qui traitent les courriers, il faut fréquemment mettre en place en amont des automates afin de trier le courrier volumineux reçu puis de le répartir vers les opérateurs qui en assureront le traitement final.

L'objet de ce TP est de construire un **automate de classement pour courriers électroniques entrants**.

Pour simplifier, on va s'intéresser à un automate à une entrée et deux sorties. En entrée, l'automate traite tous les courriers entrants qui sont stockés dans un répertoire et il les sépare en deux listes : l'une avec tous les courriers reçus qui sont présumés intéressants à traiter et l'autre avec tous les autres courriers qui sont présumés sans intérêt. Les courriers entrants sont des fichiers au format *filename.eml*[1].



Cet automate de classement fonctionne comme un filtre. D'un côté, on laisse passer tout ce qui est présumé intéressant (courrier légitime : ici, dans le TP, des offres d'emploi et de stage) et, de l'autre, on poubellise tout ce qui est présumé sans intérêt (ici, dans le TP, du spam).

2 – Classement avec le modèle vectoriel appliqué aux fichiers de texte

On va utiliser le modèle vectoriel d'indexation des textes. Ce modèle vectoriel a été initialement développé par Gerard Salton[2] [3], un des pionniers de la recherche d'information dans les documents.

[1] Format EML : L'extension .EML (abréviation d'email) caractérise un message électronique qui peut être manipulés par plusieurs logiciels de messagerie (par exemple, Outlook).

[2] G. Salton , A. Wong , C. S. Yang, "A vector space model for automatic indexing" - Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975

[3] Gerard Salton, M. J. McGill, « Introduction to Modern Information Retrieval », 1983

Pour cela, on fait une analyse *full text* du contenu des fichiers EML entrants et on compte les mots qui appartiennent au vocabulaire de référence (ici, offres d'emploi ou de stage).

Le vocabulaire est défini en s'appuyant les hypothèses suivantes :

- Pas de prise en compte de la casse (minuscule = majuscule).
- Pas de prise en compte des accents et signes diacritiques (par exemple, é = è = ê = e)
- Pas de prise en compte des attributs de caractères (*italique* = souligné = **gras**)
- Pas de prise en compte de la ponctuation

Le modèle vectoriel consiste à décrire un texte sur un espace vectoriel **N**-dimensionnel dont les vecteurs de bases sont les **N** mots du vocabulaire de référence. Ainsi, le texte est assimilé à un vecteur **V** de dimension **N** dont chaque composante **V[k]**, pour k=1..N, correspond au nombre d'occurrences du k-ième mot du vocabulaire de référence.

Par exemple, soit le texte **T** = « Le petit chat est mort » et le vocabulaire = {petit, gros, chat, blanc, noir, mort, vivant}. Alors, le cardinal du vocabulaire est **N** = 7 et le vecteur **V** représentant le texte **T** sur ce vocabulaire est **V** = [1, 0, 1, 0, 0, 1, 0]

Dans ce même modèle vectoriel, le texte **T'** = « Un gros chat est plus lent qu'un petit chat » aurait pour représentation vectorielle **V'** = [1, 1, 2, 0, 0, 0, 0].

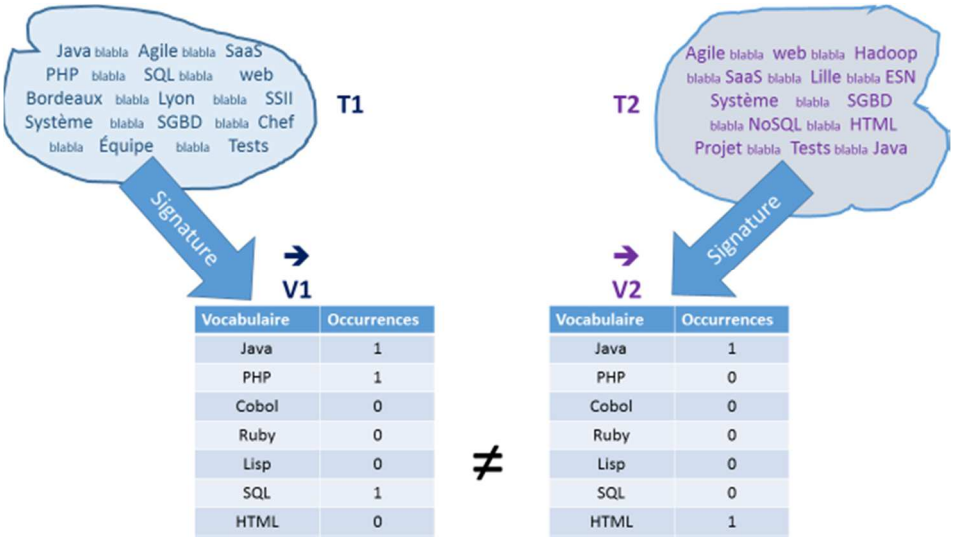
L'ensemble de représentation des textes est un vocabulaire formé avec les mots les plus significatifs du corpus considéré : noms communs, noms propres, adjectifs, etc. Chaque texte **T** est ainsi représenté par un vecteur **V**, dont la dimension correspond au cardinal **N** du vocabulaire. Chaque élément **V[k]** du vecteur **V** correspond au poids associé au mot d'indice k dans le vocabulaire et pour le texte Indexé.

Avec le modèle vectoriel, il est notable que l'ordre d'apparition des mots du vocabulaire dans le texte n'a aucun impact sur la signature.

D'autre part, avec le modèle vectoriel, il n'y a aucun intérêt à incorporer dans le vocabulaire des mots banals (articles, préposition, conjonction de coordination, etc.). Cela ne ferait qu'alourdir la signature sans apporter aucun gain d'information sémantique.

Une fois que chaque texte est associé à sa signature vectorielle sur un vocabulaire de référence, il devient possible de tester des similarités sémantiques entre deux textes.

Par exemple, les deux textes suivants (T1 et T2) sont considérés comme différents parce que leurs signatures respectives (V1 et V2) sont différentes.



Vocabulaire	Occurrences
Java	1
PHP	1
Cobol	0
Ruby	0
Lisp	0
SQL	1
HTML	0

≠

Vocabulaire	Occurrences
Java	1
PHP	0
Cobol	0
Ruby	0
Lisp	0
SQL	0
HTML	1

A contrario, les deux textes suivants (T1 et T2) sont considérés comme similaires parce que leurs signatures respectives (V1 et V2) sont identiques.

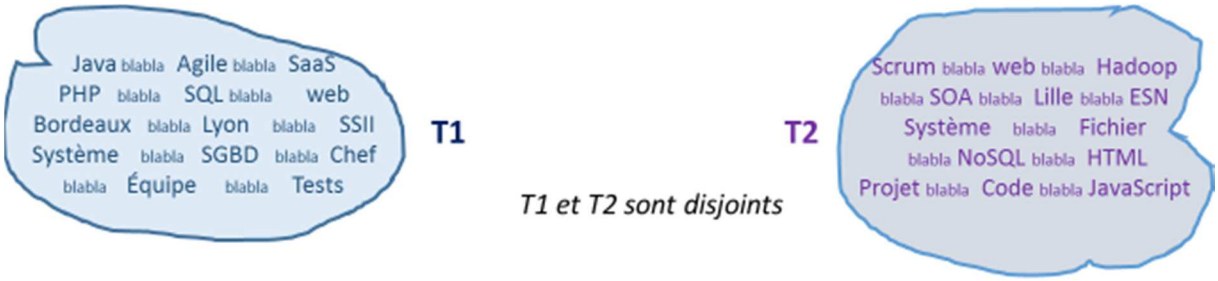
Vocabulaire	Occurrences
Java	1
PHP	0
Cobol	0
Ruby	0
Lisp	0
SQL	1
HTML	1

=

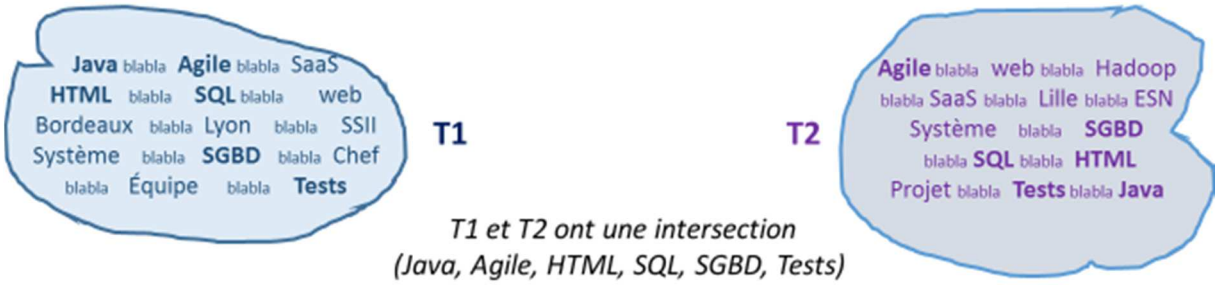
Vocabulaire	Occurrences
Java	1
PHP	0
Cobol	0
Ruby	0
Lisp	0
SQL	1
HTML	1

On note bien que « similaire » n'est pas la même chose que « identique ». Deux textes sont « similaires » si leurs signatures respectives sont « identiques ». Identique, pour une signature, signifie ici que les vecteurs sont strictement égaux.

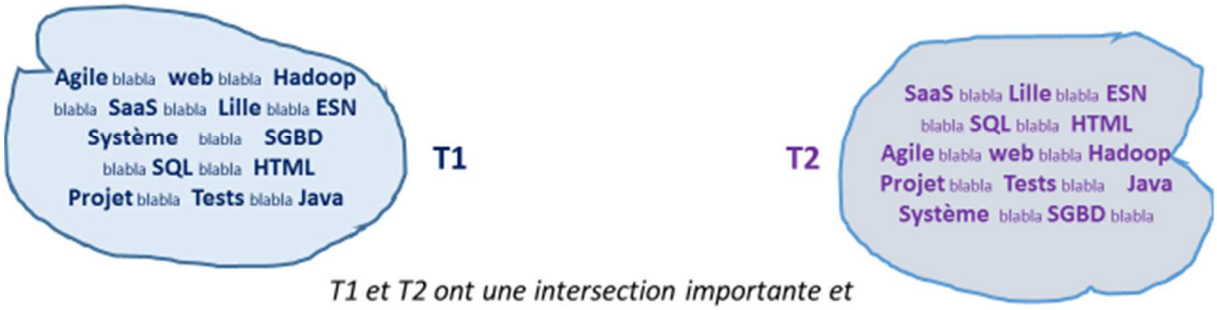
Lorsque le cardinal du vocabulaire augmente, la probabilité que deux signatures soient identiques (pour deux textes différents) chute. Il devient alors utile et nécessaire de raisonner en termes de proximité et non plus en termes d'identité.



T1 et T2 sont disjoints



T1 et T2 ont une intersection
(Java, Agile, HTML, SQL, SGBD, Tests)



T1 et T2 ont une intersection importante et
sont probablement identiques si l'on se fie à leur signature

Si deux textes **T1** et **T2** sont disjoints, leurs signatures **V1** et **V2** sont orthogonales (et réciproquement).

Si deux textes **T1** et **T2** ont une intersection importante, leurs signatures **V1** et **V2** peuvent être colinéaires, voire identiques. En ce dernier cas, les deux textes seront considérés comme « similaires ».

Entre ces deux extrêmes, on a un éventail de situations possibles. On peut caractériser ces différentes situations par la proximité sémantique de deux textes.

Pour mesurer la proximité sémantique de deux textes **T1** et **T2**, on peut utiliser le produit scalaire des signatures **V1** et **V2**.

- Si le vocabulaire de **T1** est un ensemble disjoint du vocabulaire de **T2**, alors les signatures vectorielles (**V1** et **V2**) sont orthogonales et leur produit scalaire **V1.V2** est nul.
- Si le vocabulaire de **T1** est identique au vocabulaire de **T2**, alors les signatures vectorielles (**V1** et **V2**) seront colinéaires et leur produit scalaire **V1.V2** sera maximal et égal à |V1|.|V2|.

- Si le vocabulaire de **T1** et celui de **T2** ont une intersection non nulle, alors les signatures vectorielles (**V1** et **V2**) formeront un angle **Θ** et leur produit scalaire **V1.V2** sera égal à |V1|.|V2| cos **Θ**.

3 – Mise en œuvre du classement

Il s'agit de vérifier la proximité sémantique des fichiers entrants. Pour cela, on mesure la distance entre les fichiers afin de définir deux groupes de fichiers : un groupe de fichiers qu'on va conserver (les offres d'emploi ou de stage) et un groupe de fichier qu'on va poubelliser (les autres).

Dans une géométrie à N dimension, la proximité est usuellement caractérisée par la distance Euclidienne.

$$\mathbf{Dij} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

1) Soit un ensemble de **N** mots du vocabulaire de référence qui constituent les **N** vecteurs de base du **modèle vectoriel**.

2) Soit **To** un texte de référence (pour la sémantique qui nous intéresse : ici, offre de stage ou d'emploi). Alors la signature vectorielle de **To** est **Vo**.

3) Par la suite, pour tout fichier entrant **Ti** :
   - Calculer **Vi** la signature vectorielle de **Ti**
   - Calculer le produit scalaire **Vo.Vi = Σ V[k,o] x V[k,i]**, pour k=1..N
   - Calculer la norme |Vo|.|Vi| afin de pouvoir normaliser le produit scalaire.
   - Renormaliser le produit scalaire de façon à calculer un écart standard entre les deux vecteurs

$$\frac{\mathbf{Vo.Vi}}{|Vo|.|Vi|} = \frac{|Vo|.|Vi| \cos \Theta i}{|Vo|.|Vi|} = \cos \Theta i$$

   - À noter que les **V[k,o]** et **V[k,i]** sont tous positifs ou nuls quels que soient k, o et i. En conséquence, l'écart standard cos **Θ**i varie entre 0 et +1.
   - Si cos **Θ**i est nul, cela signifie que les vecteurs **V[k,o]** et **V[k,i]** sont orthogonaux. Autrement dit, les textes **Ti** et **To** sont disjoints. En ce cas, on peut poubelliser le fichier **Ti** (cas **KO**).
   - Si cos **Θ**i est inférieur à 0,8, cela signifie que les vecteurs **V[k,o]** et **V[k,i]** sont très dissemblables. Autrement dit, le texte **Ti** est sémantiquement très éloigné du texte **To.** En ce cas, on peut poubelliser le fichier **Ti** (cas **KO**).
   - Si cos **Θ**i est supérieur à 0,8 cela signifie que les vecteurs **V[k,o]** et **V[k,i]** sont presque colinéaires. Autrement dit, le texte **Ti** est sémantiquement assez proche du texte de référence **To.** En ce cas, on considère ce texte comme intéressant à traiter (cas **OK**).
   - Si cos **Θ**i est égal à 1, cela signifie que les vecteurs **V[k,o]** et **V[k,i]** sont colinéaires. Autrement dit, le texte **Ti** est sémantiquement similaire au texte **To.** En ce cas, on considère ce texte comme intéressant à traiter (cas **OK**).

Pour le test de similarité entre **To** et **Ti**, nous utilisons l'écart standard cos **Θ**i au seuil de 0,8. Ce seuil dépend de la qualité du texte de référence **To.** Il se peut que l'expérience nous amène à faire évoluer ce seuil : soit pour être plus sévère (augmentation du seuil), soit pour être plus laxiste (diminution du seuil).

À l'usage, on pourra également faire évoluer le texte **To** afin que sa signature soit plus représentative des signatures des fichiers EML que l'on souhaite conserver (cas **OK**).

4 – A faire pour ce TP

Prendre les fichiers EML présents dans le dossier ***EML_a_trier*** et les séparer en deux listes OK et KO.

Rédiger un compte rendu de TP avec :

- les résultats obtenus

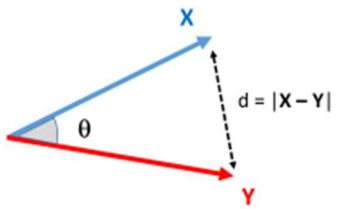
- un explication sur les outils, technologies et méthodes mise en œuvre.

5 – Compléments utiles

5.1 – Relation entre la distance inter-vecteurs et cos Θ

Soit **X** et **Y** deux vecteurs

avec **X** : (x[1], x[2], x[3], . . . , x[i], . . ., x[N]) et **Y** : (y[1], y[2], y[3], . . . , y[i], . . ., y[N]).



Les deux vecteurs **X** et **Y** appartiennent à un espace vectoriel à N dimensions et sont décrits dans un repère orthonormé. Dans le contexte d'utilisation de ces vecteurs, tous les x[i] et tous les y[i] sont positifs ou nuls. En conséquence, on peut supposer que les deux vecteurs X et Y forment un angle **Θ** qui est positif et inférieur à 90°. Autrement dit, cos **Θ** est compris entre 0 et +1.

Le produit scalaire **X**.**Y** = **Σ x[i] . y[i]**, pour i=1..N.

Le carré de la norme de **X** est $|\mathbf{X}|^2$ = **X**.**X** = **Σ x[i] . x[i]**, pour i=1..N.

Le carré de la norme de **Y** est $|\mathbf{Y}|^2$ = **Y**.**Y** = **Σ y[i] . y[i]**, pour i=1..N.

En conséquence,

$$\frac{\mathbf{X.Y}}{|\mathbf{X}|.|\mathbf{Y}|} = \frac{|\mathbf{X}|.|\mathbf{Y}|\cos\Theta}{|\mathbf{X}|.|\mathbf{Y}|} = \cos\Theta$$

Par ailleurs, $\mathbf{X} - \mathbf{Y}$ représente le vecteur-écart entre **X** et **Y**.

avec $\mathbf{X} - \mathbf{Y}$: (x[1]-y[1], x[2]-y[2], x[3]-y[3], . . . , x[i]-y[i], . . ., x[N]-y[N])

En conséquence, la distance d est égale à la norme de $\mathbf{X} - \mathbf{Y}$, soit $|\mathbf{X} - \mathbf{Y}|$.

Autrement dit, $d^2 = |\mathbf{X} - \mathbf{Y}|^2 = (\mathbf{X}-\mathbf{Y}).(\mathbf{X}-\mathbf{Y}) = \mathbf{X.X} - \mathbf{Y.X} - \mathbf{X.Y} + \mathbf{Y.Y} = |\mathbf{X}|^2 + |\mathbf{Y}|^2 - 2\,\mathbf{X.Y}$

D'où $d^2 = |\mathbf{X} - \mathbf{Y}|^2 = (\mathbf{X}-\mathbf{Y}).(\mathbf{X}-\mathbf{Y}) = |\mathbf{X}|^2 + |\mathbf{Y}|^2 - 2\,\mathbf{X.Y}$

$$d^2 = |\mathbf{X} - \mathbf{Y}|^2 = |\mathbf{X}|^2 + |\mathbf{Y}|^2 - 2\,\mathbf{X.Y} = |\mathbf{X}|^2 + |\mathbf{Y}|^2 - 2\,|\mathbf{X}|.|\mathbf{Y}|\cos\Theta$$

Ce qui permet de relier la distance inter-vecteur $d = |\mathbf{X}-\mathbf{Y}|$ et $\cos\Theta$ où Θ est l'angle inter-vecteur**.** Par la suite, chercher à minimiser $|\mathbf{X} - \mathbf{Y}|$ revient à maximiser $\cos\Theta$**.** Ce qui revient à minimiser Θ**.**

$$2\cos\Theta = \frac{|\mathbf{X}|}{|\mathbf{Y}|} + \frac{|\mathbf{Y}|}{|\mathbf{X}|} - \frac{|\mathbf{X}-\mathbf{Y}|^2}{|\mathbf{X}|.|\mathbf{Y}|}$$

5.2 – Détermination d'un texte de référence To.

On cherche à définir un texte de référence qui correspond à un fichier dont la sémantique est en adéquation parfaite avec l'usage qu'on en attend (ici, offre de stage ou d'emploi). Soit **To** ce texte de référence. Alors la signature vectorielle de **To** est **Vo**.

Plutôt que de chercher à construire un texte de référence **To**, il est plus pragmatique de chercher à déterminer une signature de référence **Vo**. En effet, c'est à partir de **Vo** (et non de **To**) que l'on fait les calculs de proximité.

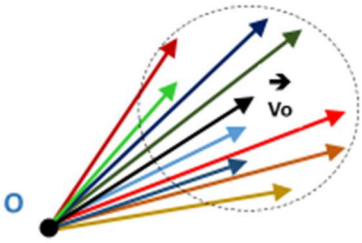
On propose de déterminer **Vo** comme le barycentre d'un échantillon de M vecteurs signatures de M textes considérés comme adéquats.

Ainsi, on part de M textes qui forment un échantillon représentatif des fichiers EML que nous considérons comme intéressants. On calcule les M vecteurs signatures de ces M textes.

Soit **V1, V2, V3, . . . , Vi, . . ., VM** les signatures de ces M textes. On peut alors calculer le barycentre de cet échantillon représentatif.

Soit $\mathbf{Vo} = (1/M)\ \mathbf{\Sigma\ Vi}$, pour i=1..M

On peut alors prendre **Vo** comme signature vectorielle de référence.



Vo est le barycentre des vecteurs-signatures des textes considérés comme adéquats.

5.3 – Détermination de la tolérance autour du texte de référence To.

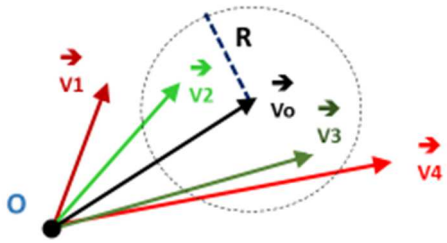
Soit R le rayon de l'hypersphère dont le centre est **Vo** et qui contient tous les **Vi**, pour i=1..M.

On peut donc choisir R tel que : R = max (|**Vi – Vo**|), pour i=1..M

En conséquence, comme tous les **Vi** sont les signatures des M textes formant un échantillon représentatif des fichiers EML considérés comme intéressants, ces **Vi** vérifient tous la propriété suivante : |**Vi – Vo** | ≤ R quel que soit i=1..M.

On définit comme tolérance autour du texte de référence To, l'hypersphère de rayon R et de centre **Vo** qui englobe les M signatures de l'échantillon représentatif des fichiers 'intéressants'.

Cet échantillon peut-être initialement constitué 'à la main'. Ou bien on le constitue statistiquement sur la base d'un historique.



Sur le schéma précédent, le vecteur **Vo** est le barycentre des vecteurs qui représentent les fichiers EML considérés comme 'intéressants'. Les vecteurs **V2** et **V3** sont les signatures de fichiers entrant classés **OK** car ils sont à l'intérieur de l'hypersphère centrée sur **Vo** et de rayon R. En revanche, les vecteurs **V1** et **V4** sont les signatures de textes considérés comme 'sans intérêt'. Les fichiers correspondants seront donc classés **KO**.