Sélection de variable : Semi-supervisée

Tout comme la partie supervisée, nous retrouvons les trois mêmes familles d'approches de sélection de variable pour l'apprentissage semi-supervisé. Nous verrons néanmoins que les techniques de sélection reposent sur des principes différents et qu'il est possible de créer des sous-catégories pour chaque approche de sélection que sont les filtres, les wrapper (symbiose) et embedded (intégrés).

Les méthodes de filtre :

Tout comme l'apprentissage supervisé, les méthodes de filtrage semi-supervisées ont pour objectif de supprimer les variables « non-signifiantes » afin de réduire l'espace de données. Ces méthodes sont indépendantes du classifieur utilisé empêchant ainsi un surajustement des données. Dans le cadre de l'approche semi-supervisée, différentes sous-familles d'approche par filtrage existent.

Théorie des graphes spectraux

Présentée en 2007 par Z.Zhao et H.Liu [13], l'algorithme *sSelect* est la première approche semi-supervisée se basant sur l'analyse spectrale permettant de résoudre le problème du « plus petit échantillon étiqueté ». En effet, il n'est pas rare dans l'approche semi-supervisée que la proportion de données étiquetées soit bien plus faible que la partie étiquetée. Ainsi, les méthodes de sélection classique utilisées en mode supervisé ne peuvent fonctionner correctement.

Pour pallier ce problème, la méthode proposée ici exploite à la fois les données labélisées et non-labelisées à travers une méthode de régularisation des données. Concrètement, les auteurs expliquent transformer les vecteurs de variable en indicateur de cluster. Cet indicateur est ensuite évalué suivant deux critères : la séparabilité et la cohérence entres les données étiquetées et non-étiquetées.

Score Laplacien

Utilisé initialement dans l'approche non-supervisée, le score Laplacien est introduit en 2005 par X.He, D.Cai et P.Niyogi [7]. Cette technique consiste à sélectionner les variables pertinentes préservant au mieux la structure locale et produisant de grandes valeurs de variances. Très performante pour les données non-étiquetées, plusieurs chercheurs ont depuis 15 ans essayé d'adapter cette méthode à l'approche semi-supervisée.

Nous pouvons par exemple citer l'algorithme *GSFS* (Graph-Based Semi-supervised Feature Selection) introduit en 2011 par Cheng and al. [6] renforçant l'aspect de préservation de localité dans le choix des variables. Les chercheurs ont pour cela introduit une nouvelle métrique appelée *S-Laplacien* dérivée du score Laplacien classique [7]. Ce nouveau score ajoute une considération pour chaque variable à préserver la structure géométrique globale. Ils définissent également un degré de gain d'information *CIG* afin de limiter la redondance des données dans leur algorithme de sélection de variable *GSFS*.

L'algorithme *LSDF* (Zhao et al) [14] fondé sur l'analyse discriminante sensible à la localité est un autre exemple de sélection de variable semi-supervisée utilisant le score laplacien. Cette méthode se décompose en trois temps : construire un graphe intra-classes combinant les

informations des données étiquetées et non étiquetées, construire un graphe inter-classes reliant les données avec différents labels puis calculer à partir de ces constructions un graphe Laplacien utilisant une formule déterminée par les chercheurs.

A noter que l'ensemble de ces algorithmes cités font partie de la plus grande famille des sélections de variables basées sur des graphes.

Critère de Fisher

Très largement utilisée dans le cadre de l'approche supervisée, plusieurs chercheurs ont depuis adapté cette technique à l'approche semi-supervisée. Concrètement, l'ensemble de ces techniques utilise la partie étiquetée des données pour les séparer en différentes classes et utilise la partie non-labellisée pour conserver la structure locale. Ces méthodes combinent donc les avantages du critère de Fisher pour la partie supervisée des données avec différents critères plus appropriés aux données non-labellisées.

C'est par exemple le cas de Yang et al. [12] à travers leur algorithme *Semi_Fisher Score* qui combine un critère de variance utilisé dans le cadre du test de Fisher et un critère de préservation de la localité proche du score Laplacien présenté auparavant.

Score de contrainte

Cette technique s'appuie sur la notion de contraintes entre les données non évoquées jusqu'à présent. Aussi bien dans l'approche supervisée que semi-supervisée, les contraintes représentent une connaissance initiale supplémentaire afin de mieux décrire la variable cible. Elle ajoute des conditions, critères entre les données souvent plus faciles à obtenir dans les jeux de données que l'étiquetage complet. Plusieurs types de contraintes existent. Néanmoins, la majorité des algorithmes semi-supervisés repose sur les contraintes « par paires ».

Les contraintes par paires précisent pour une paire de données spécifiques si elles appartiennent, ou non, à la même classe. On nomme alors ces contraintes « must-link » (M) pour des classes identiques et « not-link » (C) pour des classes différentes. C'est à partir de ces contraintes qu'est défini un « score de contrainte » propre à chaque algorithme de sélection de variables. Cela se traduit principalement par la définition d'une formule mathématique cherchant à maximiser le nombre de contraintes satisfaites.

Introduite en 2008 par Zhang et al en 2008 [15] dans le cadre de l'approche supervisée, les méthodes reposant sur le score de contrainte peuvent facilement s'étendre au cadre semi-supervisé. C'est par exemple le cas de l'algorithme proposé par Zhao et al [14] qui défini un score utilisant à la fois les contraintes par paires et les plus proches voisins non étiquetés des échantillons. Il a d'ailleurs été montré par des résultats expérimentaux que cette approche obtient de meilleures performances que les approches utilisant le critère de Fisher.

Une amélioration à ces algorithmes a ensuite été proposé en 2011 [9] par Kalakech et al. Les chercheurs ont défini leur score de contraintes comme le simple produit entre le score Laplacien (prenant en compte les données non-labelisées) et le score de contrainte défini par Zhang [15] prenant en compte les données étiquetées.

Les méthodes de symbiose:

Les méthodes de symbiose en semi-supervisé peuvent être divisées en deux catégories : les méthodes basées sur un seul classifieur et les méthodes basées sur un ensemble de classifieurs.

Un seul classifieur

Ren et Al [10] ont introduit en 2008 le Forward Semi-Supervised Feature Selection (*FW-SemiFS*). L'algorithme utilise l'algorithme de *sélection de variables séquentiel avant* (*SFFS*) avec l'approche wrapper. En pratique, cette méthode consiste à sélectionner les n premières variables afin de construire un classifieur qui sera utilisé pour prédire les étiquettes des données non étiquetées. Ensuite, les données non étiquetées sélectionnées aléatoirement avec leurs étiquettes prédites sont combinées avec les données déjà étiquetées pour former un nouvel ensemble d'apprentissage.

Dans un second temps, le nouvel ensemble de données d'apprentissage est utilisé pour sélectionner les variables en fonction de la SFFS et de l'apprenant. Le processus de sélection aléatoire et de sélection des variables est répété un certain nombre de fois et un certain nombre de groupes de variables sont sélectionnés. La fréquence de chaque caractéristique dans les groupes de variables est calculée, et celle qui a la plus grande fréquence est ajoutée pour former un nouveau sous-ensemble de variables. Ce processus se répète jusqu'à ce que la taille du sous-ensemble de variables atteigne un nombre prédéfini.

Input: L, U, $sizeFS$, $samplingRate$, $samplingTimes$, $maxIterations$, $startfn$, $fnstep$
Output: $resultfs$

```
1  Perform feature selection on L using SFFS, select startfn features to
   form the current feature subset currentfs;
2  ReducedL ← L * currentfs;
3  ReducedU ← U * currentfs;
4  for iteration ← 1 to maxIterations do
5      Predicted ← classifier(ReducedL, ReducedU);
6      for rand ← 1 to samplingTimes do
7          Randomly select samplingRate% of instances from Predicted,
           and add it into L to form a new dataset NewDataset;
8          Perform feature selection on NewDataset using SFFS, select
           fnstep features to form feature subset fs[rand];
9      end
10     Count the frequency of every feature in fs, add the most frequent
       and not in currentfs feature into currentfs;
11     ReducedL ← L * currentfs;
12     ReducedU ← U * currentfs;
13     if SIZE(currentfs) == sizeFS then break;
14 end
15 resultfs ← currentfs;
```

Fig. 1. Forward semi-supervised feature selection (FW-SemiFS)

Algorithme 1 FW-SemiFS

Néanmoins comme l'évoque Han et al [8] dans leur article de 2011, la méthode FW-SemiFS ne tient pas compte de la confiance des données prédites non étiquetées, mais évalue plutôt la pertinence des variables en fonction de leur fréquence. Ces fréquences sont obtenues par une sélection itérative supervisée de variables séquentielles avant (SFFS). Cependant, le temps de calcul important associé à la SFFS itérative est préjudiciable à FW-SemiFS. De plus, cette méthode d'évaluation de la pertinence élimine le principal avantage de la sélection de variables de type wrapper : la possibilité d'évaluer le pouvoir discriminant d'une combinaison de variables [8].

Ensemble de classifieur

D'autre part, il existe des méthodes entraînant plusieurs classifieurs avant de combiner leurs résultats en sortie.

En effet dans l'article de Barkia et al [2], les chercheurs proposent une méthode d'évaluation de l'importance des variables semi-supervisées, appelée SSFI. Cette méthode combine le co-training et le random forest [5] avec une nouvelle mesure de l'importance des variables basées sur la permutation tout en utilisant les données étiquetées et non étiquetées. SSFI combine à la fois des stratégies de ré-échantillonnage des données (bagging) et de sous-espace aléatoire pour générer un apprenant d'ensemble à l'aide d'un algorithme de type co-training. La combinaison de ces deux stratégies pour produire un ensemble de classifieurs conduit à l'exploration de points de vue distincts sur les relations inter-modèles. Une fois que chaque membre de l'ensemble est obtenu, une extension de la mesure d'importance de permutation RF [5], utilisant les données étiquetées et non étiquetées ensemble, est proposée pour mesurer la pertinence de la variable. Un classement de toutes les variables est finalement obtenu par rapport à leurs pertinences dans tous les classifieurs semi-supervisés obtenus.

Algorithm 1 $SSFI(L, U, F, K, N, n, maxiter, BaseLearn)$

Require:
set of labeled training examples (L), set of unlabeled training examples (U), input space ($F = \{f_1, \ldots, f_p\}$), number of classes (K), committee size (N), sample size (n), maximum number of iterations ($maxiter$) and base learning algorithm ($BaseLearn$)

Get the class prior probabilities, $\{Pr_k\}_{k=1}^{K}$
Set the class growth rate, $n_k = n \times Pr_k$ where $k = 1, \ldots, K$

Initial committee construction H
$H = \emptyset$
for $i = 1 : N$ **do**
  RSM^i = randomly draw m features from F
  L^i_{bag} = bootstrap sample from L projected onto RSM^i
  U^i_{bag} = bootstrap sample from U projected onto RSM^i
  $L^i_{oob} = L \backslash L^i_{bag}$, $U^i_{oob} = U \backslash U^i_{bag}$
  $h^i = BaseLearn(L^i_{bag})$
  $H = H \cup h^i$
end for

Committee refinement using SSL ensemble method
$t = 1$
repeat
  for each $h^i \in H$ **do**
    π^i =*SelectConfidentExamples*$(i, H, U^i_{bag}, \{n_k\}_{k=1}^{K})$
    $L^i_{bag} = L^i_{bag} \cup \pi^i$, $U^i_{bag} = U^i_{bag} \backslash \pi^i$
    $h^i = BaseLearn(L^i_{bag})$
  end for
  $t = t + 1$
until ($t > maxiter$ **OR** *no committee member changes*)

Feature relevance estimate
$imp = 0$
for each $h^i \in H$ **do**
  $[O^i_{data}, O^i_{label}, O^i_{conf}]$ =*BuildOOBMatrix*$(i, H, L^i_{oob}, U^i_{oob}, K)$
  for each $f \in RSM^i$ **do**
    randomly permute the values of f over the O^i_{dara} examples to form O^i_{perm}
    for each $x \in O^i_{perm}$ **do**
      if $(h^i(x) \neq O^i_{label}(x))$ **then**
        $imp(f) = imp(f) + O^i_{conf}(x)$
      end if
    end for
  end for
end for
rank the features f according to $imp(f)$
return F and imp

Algorithme 2 SSFI

De plus, Bellal et al [4] présente en 2012 une méthode similaire à SSFI appelée méthode de classement des variables guidée par l'apprentissage d'ensemble semi-supervisé (SEFR), l'algorithme classe les variables à travers un ensemble de modèles, dans lequel la pertinence d'une variable est évaluée par sa précision prédictive en utilisant des données étiquetées et non étiquetées.

Data: $\mathcal{D}, L, U, F = \{f_1, \ldots, f_p\}, nbags, BaseLearn, \phi$
Result: F,imp
$Imp = 0$;
for $i = 1$:$nbags$ **do**
  L_{bag}= bootstrap sample from L;
  U_{bag}= bootstrap sample from U;
  $L_{oob} = L \backslash L_{bag}$; $U_{oob} = U \backslash U_{bag}$;
  randomly draw $\sqrt{|F|}$ features from F to form F';
  /* Labeling by self-training */
  while $U_{bag} \neq \emptyset$ **do**
    $U' = selectMostConfident(\mathcal{D}(L_{bag} \cup U_{bag}, F'), L_{bag}, U_{bag}, \phi)$;
    if $U' = \emptyset$ **then**
      break;
    end
    $L_{bag} = L_{bag} \cup U'$;
    $U_{bag} = U_{bag} \backslash U'$;
  end
  /* Feature importance measures */
  apply ϕ to $\mathcal{D}(L_{oob} \cup U_{oob}, F')$;
  select the well classified samples in L_{oob} to form L';
  $U' = selectMostConfident(\mathcal{D}(L_{oob} \cup U_{oob}, F'), L_{oob}, U_{oob}, \phi)$;
  define y as the predicted labels of $L' \cup U'$;
  for each $f \in F'$ **do**
    randomly permute f in $\mathcal{D}$ to form $perm\mathcal{D}$;
    apply ϕ to $perm\mathcal{D}(L\prime \cup U\prime, F\prime)$;
    define y_p as the predicted labels of $L' \cup U'$;
    Increase $imp(f)$ by the number of mismatches between y and y_p;
  end
end
rank the features f according to $imp(f)$ and return both F and imp;

Algorithme 3 SEFR

Les méthodes intégrées

Sur le même principe que les méthodes filtres ou de symbioses, les méthodes intégrées semi-supervisées s'appuient sur une combinaison d'algorithmes reconnus en mode supervisé et/ou non supervisé. Les avantages sont donc très semblables, à savoir l'intégration du processus de sélection de variables directement dans la phase d'apprentissage du classifieur optimisant au mieux cette sélection. Ces méthodes sont plus rapides que les approches wrapper mais moins que l'approche filtrage. L'inconvénient majeurs de cette technique est en revanche la spécificité de chaque sous-ensemble de variables sélectionnées. En effet, les variables choisies étant spécifique à un classifieur, il faut pour chaque modification de classifieur réexécuter toute la phase de sélection.

Parmi les nombreux algorithmes intégrés existant, nous évoquerons principalement les algorithmes s'appuyant sur les classifieurs linéaires tel que SVM. (Support-Vector Machine). Ces algorithmes utilisent pour cela différentes méthodes de sélection comme la régularisation multiple (*Manifold Regularization*), les normes de régularisation l1 et/ou l2 ou l'élimination récursive de variables.

FS-Manifold

Dans leur article de 2009, Xu et al [11] ont d'ailleurs proposé une nouvelle méthode de sélection semi-supervisée discriminative s'appuyant sur l'idée de la régularisation multiple introduit par Belkin et al [3]. Ce nouvel algorithme sélectionne les variables en maximisant la marge de classification entre les différentes classes et en exploitant parallèlement la structure locale des données labélisées et non-labelisées. D'après les chercheurs, cette méthode permet de trouver des variables plus discriminées. Leurs résultats expérimentaux ont aussi montré des performances supérieures aux algorithmes classiques de filtrage tel que le critère de Fisher. Cela est principalement lié à la grande optimisation de leur méthode qui peut se formaliser à travers un problème d'optimisation concave-convexe à résoudre.

Elimination récursive de variable (Backward elimination)

Autre approche de sélection semi-supervisées les algorithmes *S3VM-RFE* et *S3VM-FS* [1] (*Ang et Al*) s'appuient sur la méthode supervisée SVM-RFE. Cette méthode est très ressemblante au principe de backward elimination détaillé précédemment. Elle consiste à classer les variables dans l'ordre décroissant en fonction de leur poids puis à supprimer celles les moins bien classer. Les variables restantes s'entrainent ensuite récursivement avec le classifieur SVM jusqu'à ce que l'ensemble des variables soient classées.

S3VM-RFE est une première évolution semi-supervisée de l'algorithme supervisé SVM-RFE. Il prend en compte à la fois les données étiquetées et celles non-étiquetées dans son processus d'apprentissage. Les données labellisées sont formées avec un SVM supervisé alors qu'un algorithme SVM non-supervisé est appliqué aux données non-labellisées. Le vecteur de pondération des variables, faisant office de fonction de score, est ensuite conçu en combinant les deux SVM utilisés. La variable avec le plus petit score est éliminée puis l'algorithme continue son apprentissage jusqu'à ce que l'ensemble des variables soient classées.

Une autre variante de cet algorithme est la méthode *S3VM-FS* utilisant un nombre de variables à conserver prédéfini.

Algorithm 1. S^{3}VM-RFE (X_0, y):

Input

Training examples, $X_0 = [x_1, x_2, ?, x_l, x_{(l+1)}, ?, x_u]^T$

Class labels, $y = [y_1, y_2, ?, y_l, y_{(l+1)}, ?, y_u]^T$,

where $[y_1, ?y_l] \in -1, +1$ and $[y_{(l+1)}, ?, y_u] = 0$

Initialize

Subset of surviving features, $s = [1, 2, ?, n]$

Feature ranked list, $r = [\ ]$

Repeat until $s = [\ ]$

Start

1. Restrict training examples to good feature indices

$$labelx = X_0(:, s);$$
$$unlabelledx = X_0(:, s);$$

2. Train the classifier

$\propto_{label}$= SVM-train (X, y); %Supervised SVM

$\propto_{unlabelled}$= SVM-train (X); %Unsupervised one-class SVM

3. Compute the weight vector of dimension length(s), for all i

$$w_i^l = \sum_k \propto_{(label_k)} y_k x_k$$
$$w_i^u = \sum_k \propto_{(unlabelled_k)} y_k x_k$$

4. Determine the sum of w_i^l and w_i^u,

$$W_i = w_i^l + w_i^u;$$

5. Compute the ranking criteria, $c_i = (W_i)^2$;

6. Find the feature with smallest ranking criterion, $f = argmin(c)$;

7. Update feature ranked list, $r = [s(f), r]$;

8. Eliminate the feature with smallest ranking criterion

$$s = s(1 : f - 1, f + 1 : length(s));$$

9. Repeat until all the features are ranked.

Output

Ranked feature set r.

Algorithme 4 S^{3}VM-RFE

Références

[1] Ang, J. C., Haron, H., & Hamed, H. N. A. (2015, May). *Semi-supervised SVM-based feature selection for cancer classification using microarray gene expression data*. In Current Approaches in Applied Artificial Intelligence: 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2015, Seoul, South Korea, June 10-12, 2015, Proceedings (pp. 468-477). Cham: Springer International Publishing.

[2] Barkia, H., Elghazel, H., & Aussem, A. (2011, December). *Semi-supervised feature importance evaluation with ensemble learning*. 2011 IEEE 11th International Conference on Data Mining (pp. 31-40). IEEE.

[3] Belkin, M., Niyogi, P., Sindhwani, V. (2006) *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples*. Journal of Machine Learning Research, 7:2399–2434.

[4] Bellal, F., Elghazel, H., & Aussem, A. (2012). *A semi-supervised feature ranking method with ensemble learning*. Pattern Recognition Letters, 33(10), 1426-1433.

[5] Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

[6] Cheng, H., Deng, W., Fu, C., Wang, Y., & Qin, Z. (2011). *Graph-based semi-supervised feature selection with application to automatic spam image identification*, Computer Science for Environmental Engineering and EcoInformatics: International Workshop, CSEEE 2011, Kunming, China, July 29-31, 2011, Proceedings, Part II (pp. 259-264). Springer Berlin Heidelberg.

[7] He, X., Cai, D., Niyogi, P. , (2005). *Laplacian score for feature selection*, Proceedings of the 18th International Conference on Neural Information Processing Systems (NIPS'05). MIT Press, Cambridge, MA, USA, 507–514.

[8] Han, Y., Park, K., & Lee, Y. K. (2011). *Confident wrapper-type semi-supervised feature selection using an ensemble classifier*. 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC) (pp. 4581-4586). IEEE.

[9] Kalakech, M., Biela, P., Macaire, L., & Hamad, D. (2011*). Constraint scores for semi-supervised feature selection:A comparative study*. Pattern Recognition Letters, *32*(5), 656-665.

[10] Ren, J., Qiu, Z., Fan, W., Cheng, H., & Yu, P. S. (2008). *Forward semi-supervised feature selection*. Advances in Knowledge Discovery and Data Mining: 12th Pacific-Asia Conference, PAKDD 2008 Osaka, Japan, May 20-23, 2008 Proceedings 12 (pp. 970-976). Springer Berlin Heidelberg.

[11] Xu, Z., King, I., Lyu, M. R. T., & Jin, R. (2010). *Discriminative semi-supervised feature selection via manifold regularization*. IEEE Transactions on Neural networks, 21(7), 1033-1047.

[12] Yang, M., Chen, Y. J., Ji, G. L. (2010). *Semi_Fisher Score: A semi-supervised method for feature selection*. 2010 International Conference on Machine Learning and Cybernetics (Vol. 1, pp. 527-532). IEEE.

[13] Zhao, Z., Liu, H. (2007). *Semi-supervised feature selection via spectral analysis*, Proceedings of the 2007 SIAM international conference on data mining (pp. 641-646). Society for Industrial and Applied Mathematics.

[14] Zhao, J., Lu, K., & He, X. (2008). Locality sensitive semi-supervised feature lection. Neurocomputing, 71(10-12), 1842-1849.

[15] Zhang, D., Chen, S., & Zhou, Z. H. (2008). *Constraint score: A new filter method for feature selection with pairwise constraints*, Pattern Recognition, 41(5), 1440-1451.