
Ensembles

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

Techniques d'ensembles

- Construire un ensemble de classifieurs à partir des données d'entraînement
- Prédire la classe des exemples de test en combinant les prédictions faites par plusieurs classifieurs

Philosophie des techniques d'ensembles

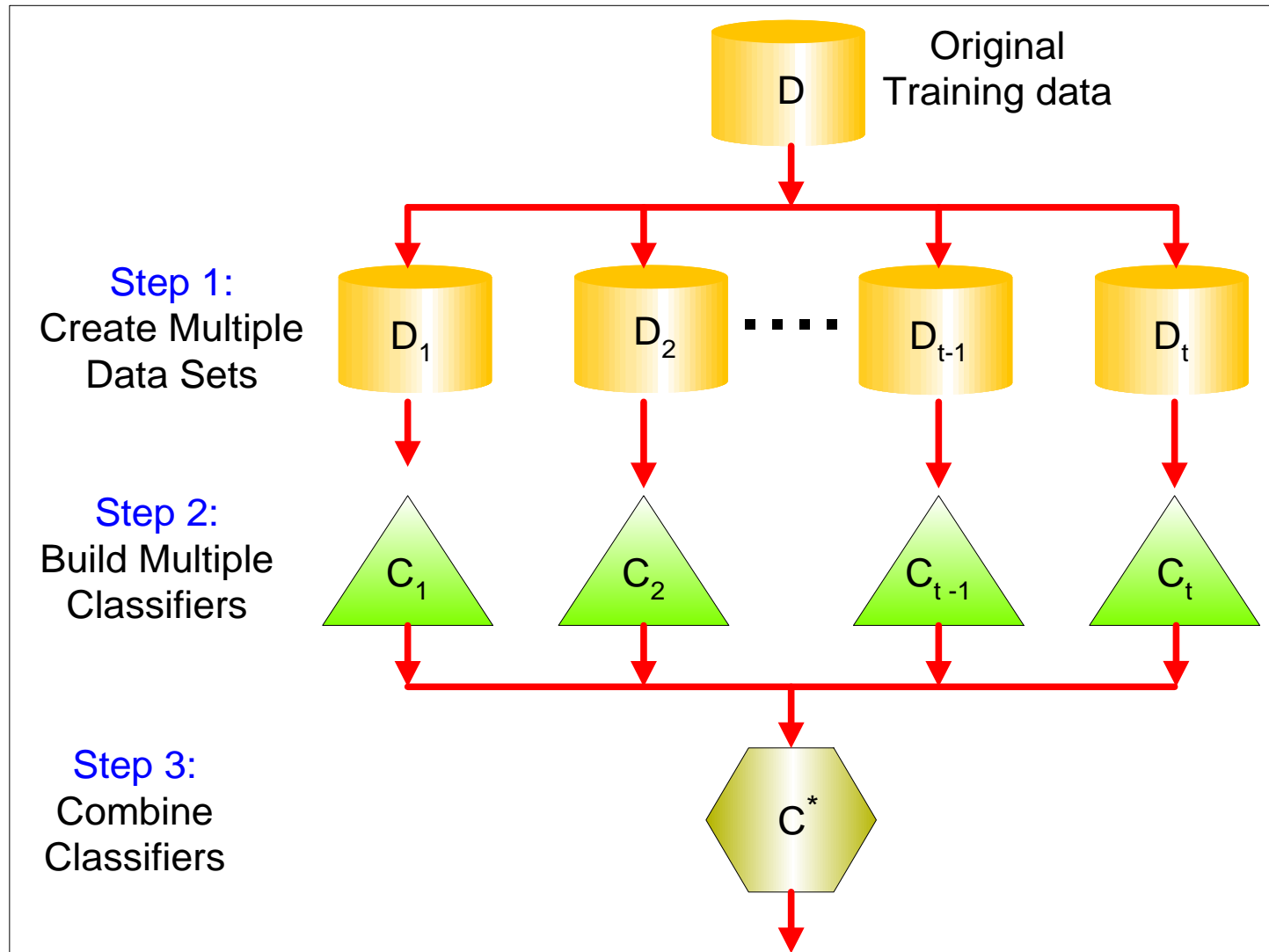
- Construire plusieurs modèles et les combiner
- Ce n'est qu'en calculant la moyenne que nous obtenons la décision finale !
- Il est trop difficile (impossible ?) de construire un modèle unique qui fonctionne au mieux.
- Deux types d'approches :
 - Modèles qui n'utilisent pas le hasard
 - Modèles qui intègrent le hasard

Pourquoi les ensembles fonctionnent ?

- Supposons qu'il y ait 25 classifieurs de base
- Chaque classifieur commet une erreur : $\varepsilon = 0.35$
 - Supposons que les erreurs commises par les classifieurs ne sont pas corrélées
 - Probabilité que l'ensemble fasse une prédiction erronée :
$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$



Approche Générale



Types of Ensemble Methods

- Manipuler la distribution des données
 - Exemple : bagging, boosting
- Manipuler les inputs
 - Exemple: random forests

Bagging

Bagging

- Construction d'ensembles d'entraînement par tirage des exemples avec remise
- Chaque exemple a une probabilité de ne pas être sélectionné de $(1 - 1/n)^n$

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Chaque ensemble d'entraînement est utilisé pour l'apprentissage d'un classifieur différent

L'Algorithme du Bagging

Algorithm 5.6 Bagging Algorithm

- 1: Let k be the number of bootstrap samples.
 - 2: for $i = 1$ to k do
 - 3: Create a bootstrap sample of size n , D_i .
 - 4: Train a base classifier C_i on the bootstrap sample D_i .
 - 5: end for
 - 6: $C^*(x) = \arg \max_y \sum_i \delta(C_i(x) = y)$, $\{\delta(\cdot) = 1$ if its argument is true, and 0 otherwise. $\}$
-

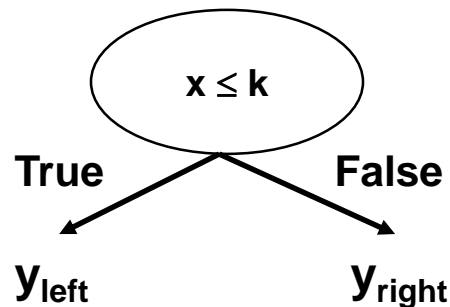
Un Exemple avec Bagging

- Nous considérons un ensemble d'entraînement avec un seul attribut

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

- Le classifieur de base est un arbre de decision avec un seul noeud : le “stump”
 - Le prédicat est : $x \leq k$ versus $x > k$
 - La coupure k est choisie à partir de l'entropie



Un Exemple avec Bagging

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

$x \leq 0.35 \rightarrow y = 1$

$x > 0.35 \rightarrow y = -1$

Un Exemple avec Bagging

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

$x \leq 0.35 \rightarrow y = 1$

$x > 0.35 \rightarrow y = -1$

Bagging Round 2:

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

$x \leq 0.7 \rightarrow y = 1$

$x > 0.7 \rightarrow y = 1$

Bagging Round 3:

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

$x \leq 0.35 \rightarrow y = 1$

$x > 0.35 \rightarrow y = -1$

Bagging Round 4:

x	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

$x \leq 0.3 \rightarrow y = 1$

$x > 0.3 \rightarrow y = -1$

Bagging Round 5:

x	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1
y	1	1	1	-1	-1	-1	-1	1	1	1

$x \leq 0.35 \rightarrow y = 1$

$x > 0.35 \rightarrow y = -1$

Un Exemple avec Bagging

Bagging Round 6:

x	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1
y	1	-1	-1	-1	-1	-1	-1	1	1	1

$x \leq 0.75 \rightarrow y = -1$
 $x > 0.75 \rightarrow y = 1$

Bagging Round 7:

x	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1
y	1	-1	-1	-1	-1	1	1	1	1	1

$x \leq 0.75 \rightarrow y = -1$
 $x > 0.75 \rightarrow y = 1$

Bagging Round 8:

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

$x \leq 0.75 \rightarrow y = -1$
 $x > 0.75 \rightarrow y = 1$

Bagging Round 9:

x	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

$x \leq 0.75 \rightarrow y = -1$
 $x > 0.75 \rightarrow y = 1$

Bagging Round 10:

x	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9
y	1	1	1	1	1	1	1	1	1	1

$x \leq 0.05 \rightarrow y = 1$
 $x > 0.05 \rightarrow y = 1$

Un Exemple avec Bagging

- Résumé des ensembles d'entraînements:

Round	Split Point	Left Class	Right Class
1	0.35	1	-1
2	0.7	1	1
3	0.35	1	-1
4	0.3	1	-1
5	0.35	1	-1
6	0.75	-1	1
7	0.75	-1	1
8	0.75	-1	1
9	0.75	-1	1
10	0.05	1	1

Un Exemple avec Bagging

- Utiliser la majorité des voix pour déterminer la classe déterminée par l'ensemble de classifieurs

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Sign	1	1	1	-1	-1	-1	-1	1	1	1

Predicted
Class

Forêts Aléatoires

Forêts Aléatoires

- Algorithme datant de 1999 (Breiman, "Random Forests").
- Idée : au lieu de faire un arbre, on fait une forêt.
- **Algorithme très puissant, qui règle les problèmes liés à un arbre seul.**
- **Un des algorithmes les plus efficaces sur de nombreux problèmes.**
- Basé sur la **randomisation**.

LE BOOTSTRAP

Idée : on transforme un ensemble d'apprentissage en B ensembles d'apprentissage.

- On part avec n exemples.
- **B fois de suite, on tire au hasard avec remise n exemples parmi les n.**

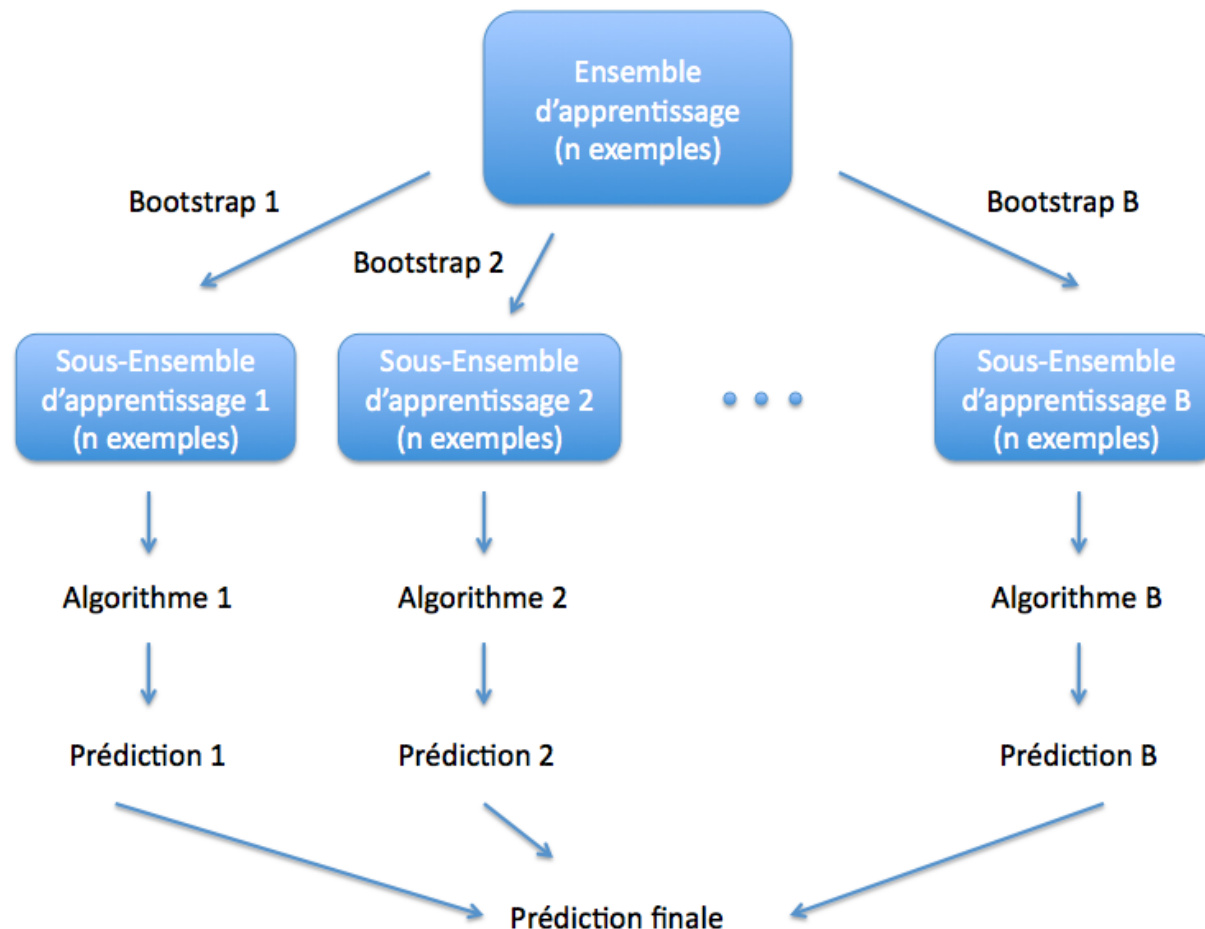
Exemple: $n = 10$. $B = 5$.

- 3, 6, 7, 4, 1, 8, 3, 3, 2, 9
- 2, 8, 5, 0, 0, 8, 6, 1, 0, 4
- 5, 9, 4, 9, 7, 8, 2, 9, 9, 1
- 5, 8, 5, 3, 1, 5, 1, 5, 4, 6
- 3, 3, 5, 0, 1, 6, 3, 2, 0, 4

De cette manière, on décuple la variabilité de l'ensemble d'apprentissage et on évite le **sur-apprentissage**

ALGORITHMES D'ENSEMBLE

Pour augmenter la puissance d'un algorithme, on peut le lancer sur B ensembles d'apprentissage bootstrappés et agréger les résultats :



ALGORITHME

Random Forests

Require: Ensemble d'apprentissage $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, nombre d'arbres B , autres paramètres pour les arbres.

1: **for** $i = 1$ to B **do**

2: Faire un arbre T_i sur le sous-ensemble d'apprentissage i . Le stocker.

3: **end for**

PRÉCISIONS

Afin de limiter le temps de calcul, on introduit un paramètre supplémentaire : à chaque noeud, on choisit au hasard un nombre de variables à regarder. Exemple : si on fait du texte, on a 1M d'attributs. A chaque noeud, on en choisit un petit nombre (quelques centaines) au hasard et on calcule le critère sur ceux-ci seulement.

Plus B est grand plus l'algorithme est efficace (et plus il est long).
Les autres paramètres sont les mêmes que pour les arbres seuls.



Boosting

Boosting

- Une méthode itérative pour modifier de façon adaptative la distribution des données d'entraînement en se concentrant davantage sur les exemples précédemment mal classifiés
- Initialement, tous les N exemples se voient attribuer une pondération égale
 - Contrairement au Bagging, les poids peuvent changer à la fin de chaque cycle de boosting

Boosting

- Les exemples qui sont mal classés verront leur poids augmenter.
- Les exemples qui sont classés correctement verront leur poids diminuer

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- L'exemple 4 est difficile à classer
- Son poids est augmenté, il a donc plus de chances d'être à nouveau choisi lors des tours suivants

ARCING : une Forme simple de Boosting

Arcing de K modèles

Initialisation: Ensemble d'apprentissage E avec n exemples.

Probabilité de tirer un exemple = $1/n$.

- 1 : **for** i = 1 to K **do**
- 2 : Former un ensemble d'apprentissage, en tirant au hasard les exemples avec remise.
- 3 : Entraîner un modèle avec les données créées.
- 4 : Tester le modèle et déterminer $m(k)$, qui est le nombre d'erreurs pour chaque exemple k de l'ensemble d'entraînement.
- 5 : Mettre à jour les probabilités de tirage des exemples selon :
$$p(k) = (1 + m(k)^4) / \text{Somme}_l(1 + m(l)^4)$$
- 6 : **end for**

CONCLUSION

Avantages du Bagging :

Très puissant car basé sur **la sagesse des foules** (plusieurs avis valent mieux qu'un).

La vitesse de calcul peut être réglée en fonction des paramètres.

Parallélisable.

Empiriquement un des meilleurs algorithmes de prédiction.

Inconvénients:

Difficile à interpréter.

Peut être long si le nombre de classifieurs est grand.

CONCLUSION (2)

Avantages du Boosting :

Très puissant car basé sur **la sagesse des foules** (plusieurs avis valent mieux qu'un).

Pour certains jeu de données il peut aboutir à des résultats très supérieurs au Bagging.

Inconvénients:

Difficile à interpreter et non parallélisable.

Si les données sont bruitées, il aura tendance à trop bien apprendre le bruit !

Peut être long si le nombre de classifieurs est grand.