Cours 3 – Arbres et méthodes ensemblistes

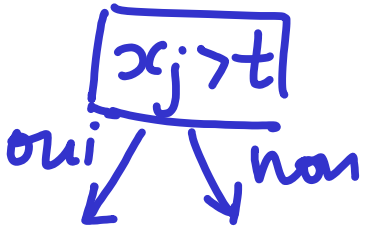
1. Arbres de décision

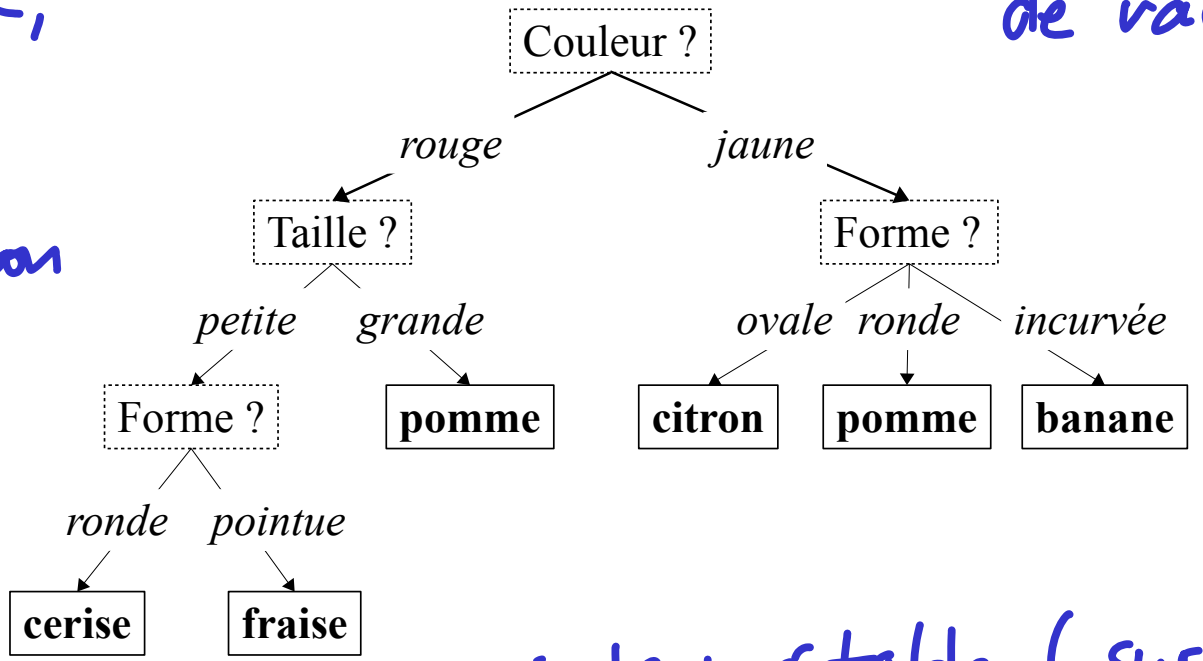
1.1 Exemple

Arbres binaires

* variables catégoriques (pas besoin de one-hot-etc.)
⊕ mélange de types de variables

* si $x_j \in \mathbb{R}$,





* interprétable (surtout si l'arbre est petit).

* multiclasse

* multimode : plusieurs façons d'être une pomme

Classification multiclasse à partir de classification binaire :

méthode générique 1-vs-all

(1 contre tous)

K classes ⇒ K classifieurs binaires

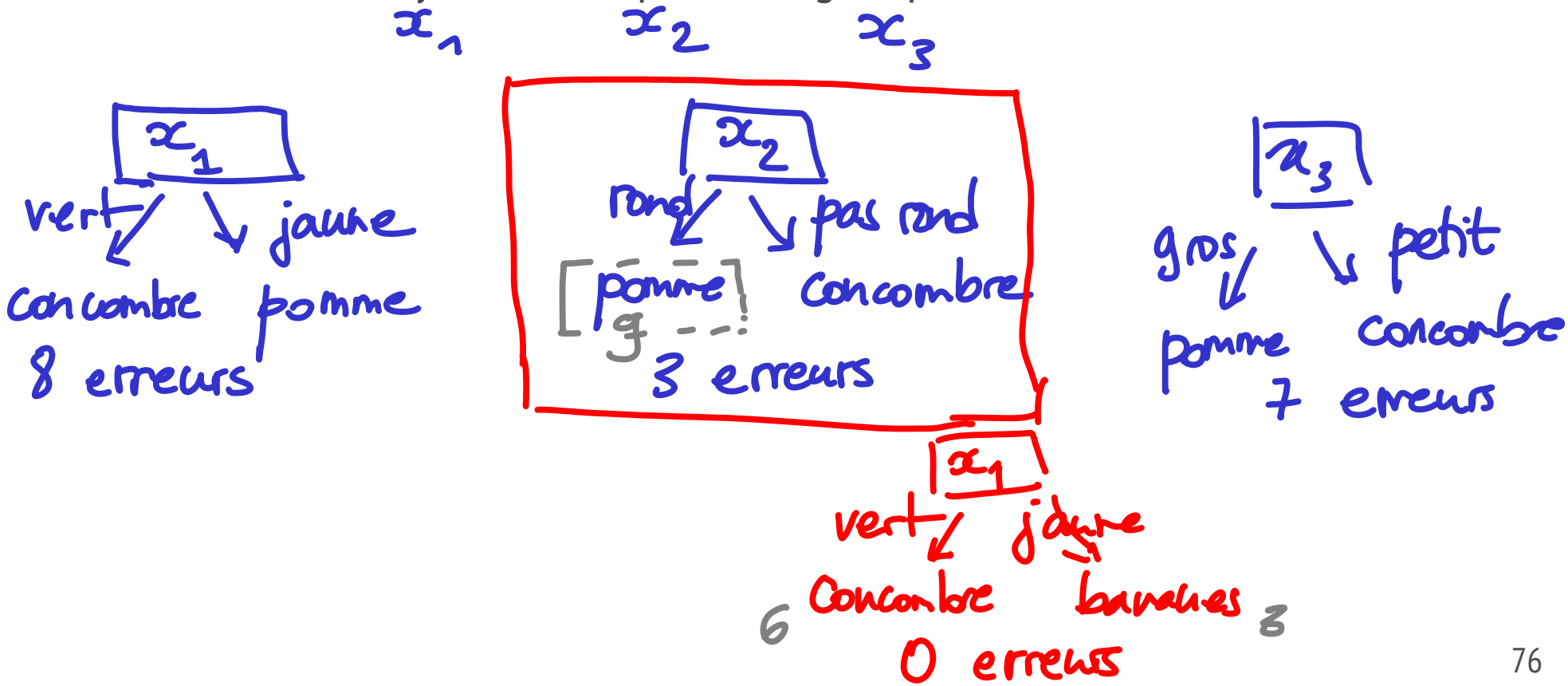
- classe 1 vs {2, 3, ..k}
- classe 2 vs {1, 3, ..., K}
- ...
- classe K vs {1, 2.., K-1}

→ retourner la classe prédite avec le plus grand score

1.2 Construction d'un arbre

Pour chaque noeud : quelle est la meilleure variable ?

- Exemple :
  - 4 petites pommes jaunes, 5 grosses pommes vertes, 3 bananes, 6 mini-concombres
  - attributs : vert/jaune, rond/pas rond, gros/petit



Quand s'arrêter ?

→ quand il n'y a plus d'erreurs
Cb risque de surapprentissage !

Plutôt :

→ à une profondeur fixée

→ ou : à un nb d'observations par feuille fixé à l'avance
(on a trouvé qch de commun à plusieurs exemples)

≡ paramètres de régularisation
déterminés par recherche en grille / validation croisée.

Complexité algorithmique

1) Pour $x_j \in \mathbb{R}$

n observations d'entraînement

$\Rightarrow$ n valeurs $\neq$ de x_j $\Rightarrow (n-1)$ seuils possibles

$n=3$ $\quad x_{1j} = 0.2 \;|\; x_{2j} = 0.4 \;|\; x_{3j} = 1$

seuil 1 seuil 2

2) Pour chaque nouveau noeud : évaluer toutes les variables et éventuellement tous les seuils

$\Rightarrow$ coûteux si: beaucoup de variables continues et de données.

1.3 Avantages et inconvénients

cf slide 1 sur les arbres

1) Le temps de calcul

2) Optimisation heuristique: on ne sait pas résoudre le pb de trouver l'arbre de décision optimal sur les données d'apprentissage

⇒ Souvent mauvaise performance en pratique.

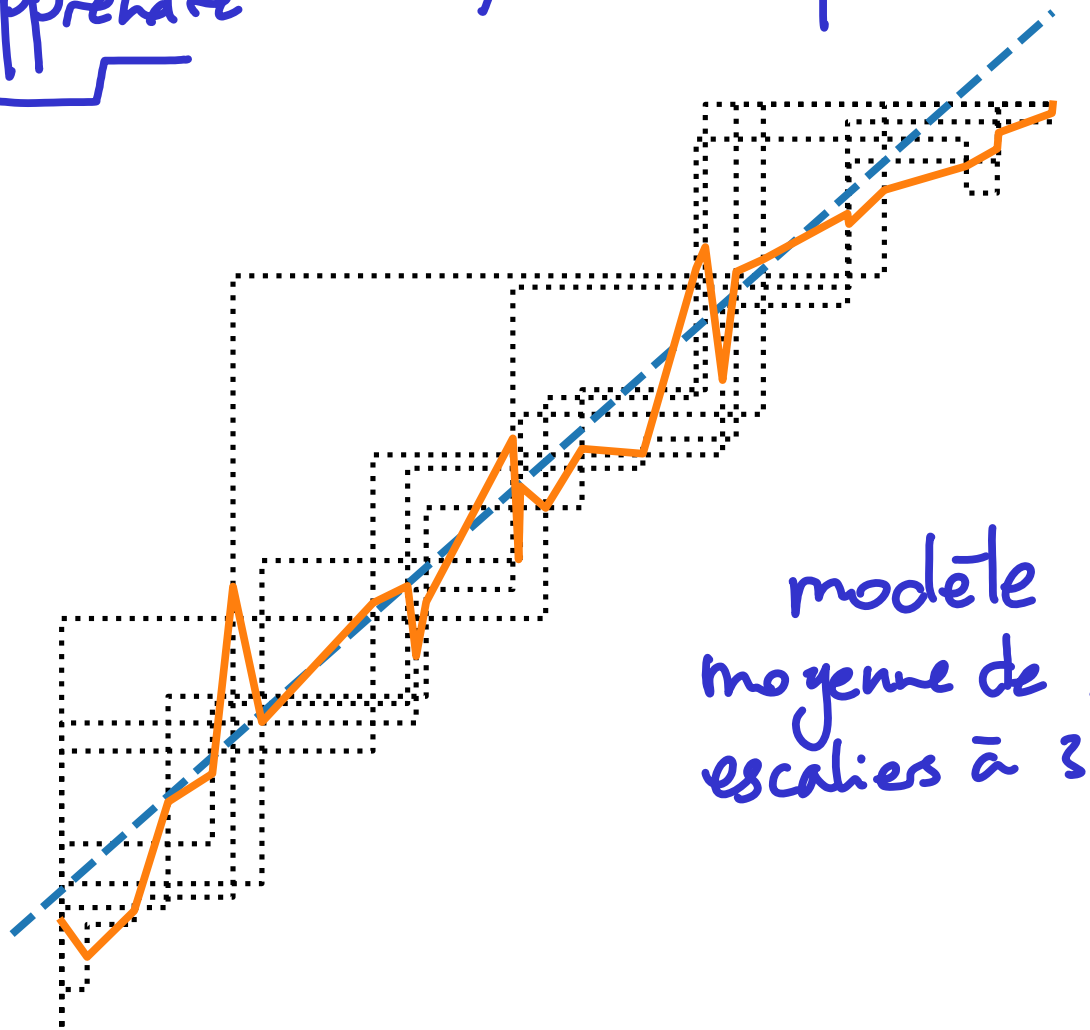
2. Méthodes ensemblistes

2.1 Sagesse des foules

– Combiner des **apprenants faibles**. (weak learners)

↳ modèles plutôt simples, pas très performants

On ne sait apprendre que des



modèle orange = moyenne de plusieurs escaliers à 3 marches

2.2 Bagging : pour générer plusieurs modèles à partir du même jeu de données.

Idée = échantillonner le jeu de données $\mathcal{D}$

hyper-paran. B = nb de modèles à construire

pour $b = 1, \ldots, B$: parallélisable

- construire $\mathcal{D}_b$: échantillon bootstrap de $\mathcal{D}$

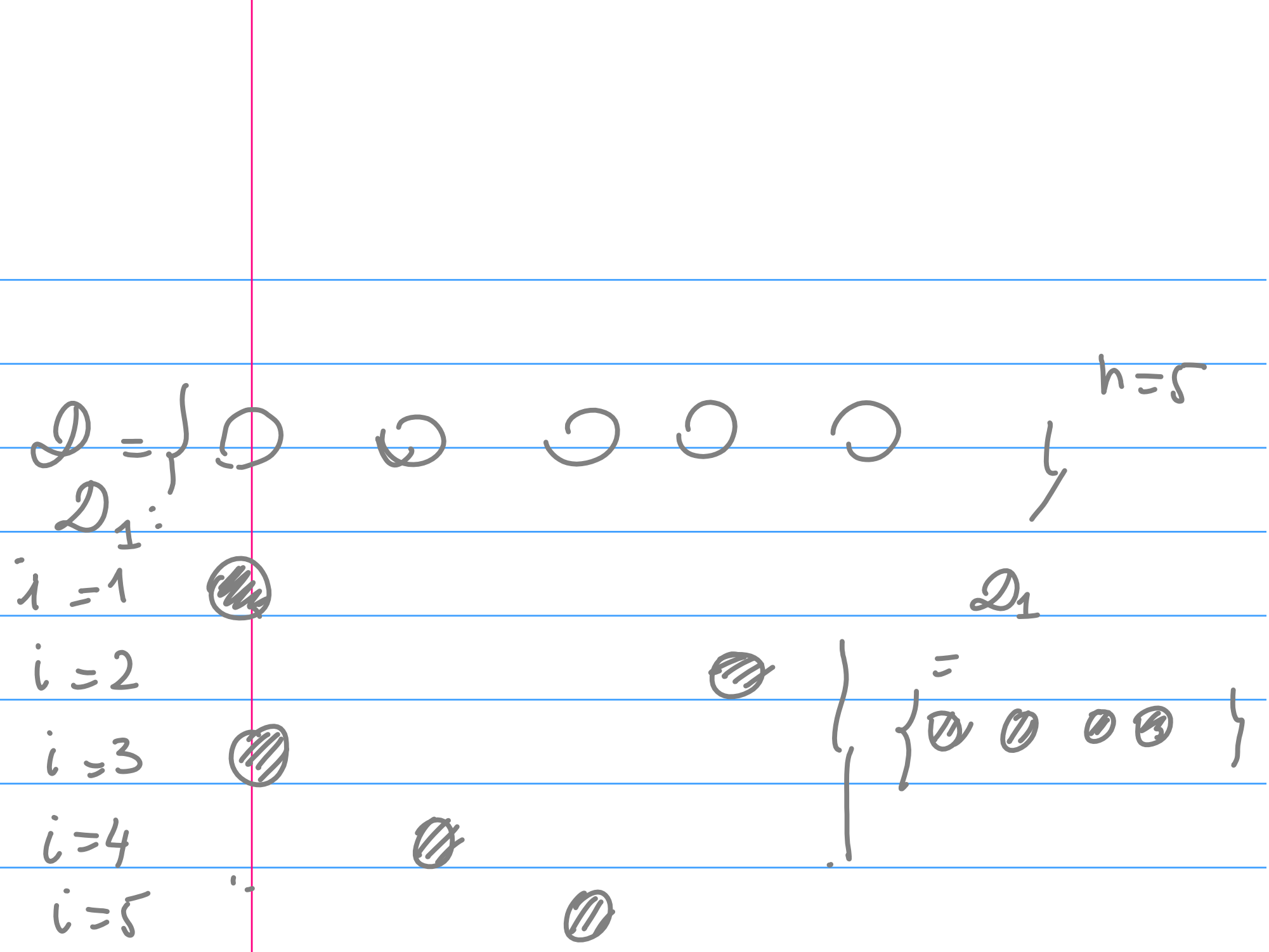
si $\mathcal{D}$ contient n observations → tirer n observations avec remise

- apprend un modèle f_b sur $\mathcal{D}_b$

combiner B modèles :

classification = par vote de la majorité

régression = moyenne des valeurs prédites


n=5
i=1
i=2
i=3
i=4
i=5

2.3 Forêts aléatoires *Random forests*

Bagging : sur des arbres de décision

⊕ chaque arbre est construit uniquement avec une partie des variables

et une partie des données (échantillon bootstrap)

$\simeq \sqrt{p}$ variables

- Créer des arbres ≠ les uns des autres
- réduire le tps d'entraînement de chaque arbre.

2.4 Boosting

$$\frac{1}{n}\sum_{i=1}^{n} w_i L(y_i, f(\vec{x}_i))$$

- À l'itération m
  - Apprendre le modèle f_m qui minimise l'erreur empirique de

$$F_m = \sum_{l=1}^{m} \alpha_l f_l = F_{m-1} + \alpha_m f_m$$

  - L'erreur pour $(\vec{x}_i, y_i)$ est pondérée de sorte à donner plus d'importances aux exemples pour lesquels F_{m-1} se trompe.
- **AdaBoost** (Schapire & Freund 1997) : erreur exponentielle, f_m est un arbre de décision de profondeur 1 (decision stump)
- **Gradient Boosting** (Friedman 2001) : forme générale