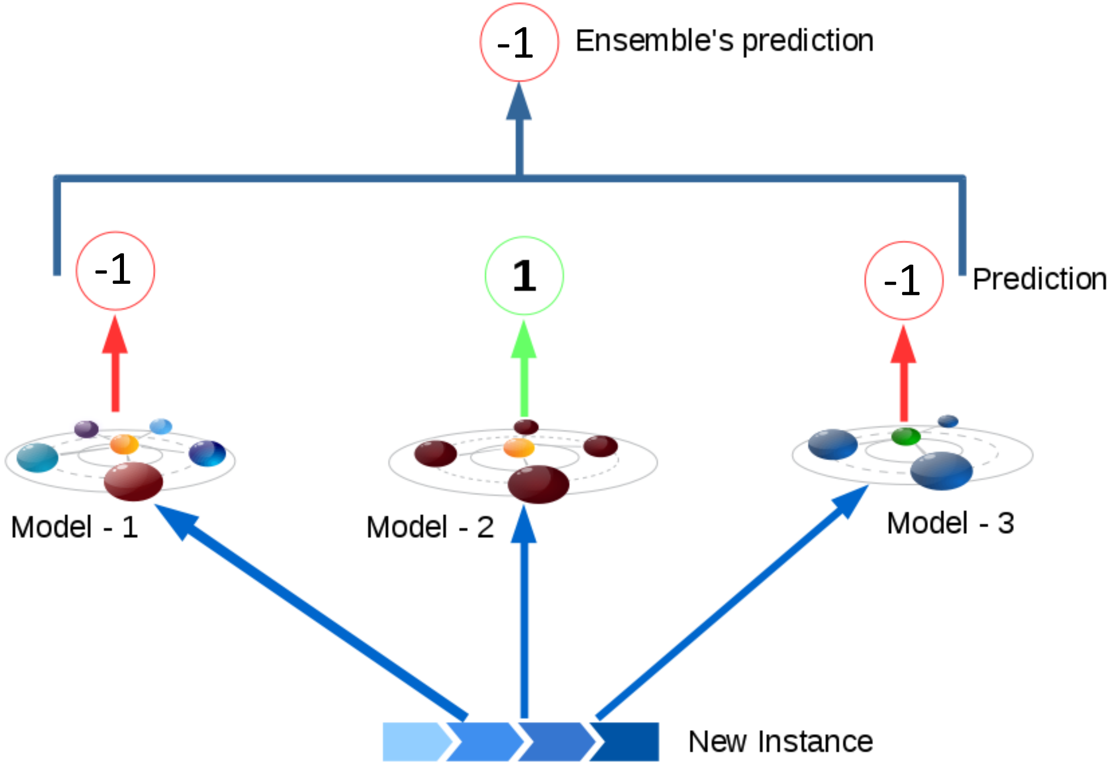
Introduction to Machine Learning

Ensemble Methods, Bagging, and Boosting

Power through diversity

Multiple classifiers: Voting



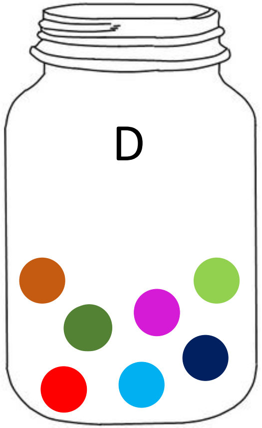
Power through diversity

Let’s try this out

Multiple training sets: Resampling

- Sampling with replacement

D



$\tilde{D}$

Multiple training sets: Bagging

- Start with training data set *D* with *N* examples
- Use sampling with replacement to create *M* datasets $\widetilde{D}_1,..,\widetilde{D}_M$
  - Each has size *N*
  - Train separate classifier on each training set
  - Combine (vote)

Boosting weak learners

- Weak learners become strong through adaptation
- AdaBoost

Algorithm 32 $\textsc{AdaBoost}(\mathcal{W}, \mathcal{D}, K)$

$d^{(0)} \leftarrow \langle \frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N} \rangle$ // Initialize uniform importance to each example

for $k = 1 \dots K$ **do**

  $f^{(k)} \leftarrow \mathcal{W}(\mathcal{D}, d^{(k-1)})$ // Train kth classifier on weighted data

  $\hat{y}_n \leftarrow f^{(k)}(x_n), \forall n$ // Make predictions on training data

  $\hat{\epsilon}^{(k)} \leftarrow \sum_n d_n^{(k-1)}[y_n \neq \hat{y}_n]$ // Compute weighted training error

  $\alpha^{(k)} \leftarrow \frac{1}{2} \log\left(\frac{1-\hat{\epsilon}^{(k)}}{\hat{\epsilon}^{(k)}}\right)$ // Compute "adaptive" parameter

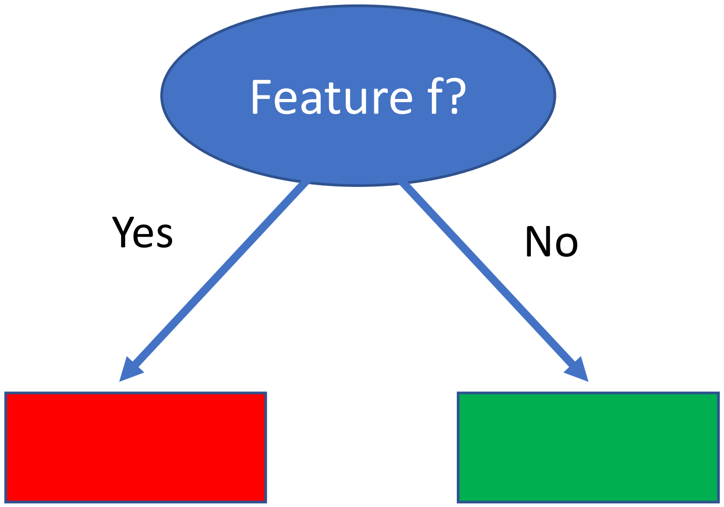
  $d_n^{(k)} \leftarrow \frac{1}{Z} d_n^{(k-1)} \exp[-\alpha^{(k)} y_n \hat{y}_n], \forall n$ // Re-weight examples and normalize

end for

return $f(\hat{x}) = \operatorname{sgn}\left[\sum_k \alpha^{(k)} f^{(k)}(\hat{x})\right]$ // Return (weighted) voted classifier

Let's try this out

Decision Stump? Really?



Random ensembles

- Decision trees are expensive to learn
- If structure was given would be faster

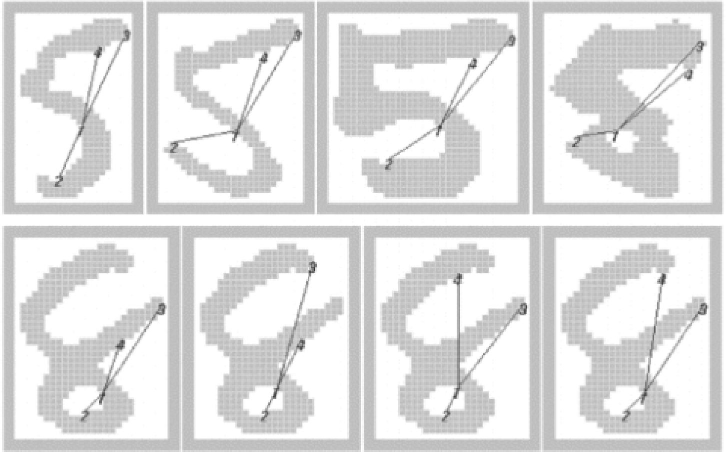
Algorithm 33 RandomForestTrain($\mathcal{D}$, *depth*, K)

for $k = 1 \dots K$ **do**
  $t^{(k)} \leftarrow$ complete binary tree of depth *depth* with random feature splits
  $f^{(k)} \leftarrow$ the function computed by $t^{(k)}$, with leaves filled in by $\mathcal{D}$
end for
return $f(\hat{x}) = \text{sgn}\left[\sum_k f^{(k)}(\hat{x})\right]$ // Return voted classifier

Performance

- Early proponents of random forests: “Joint Induction of Shape Features and Tree Classifiers”, Amit, Geman and Wilder, PAMI 1997

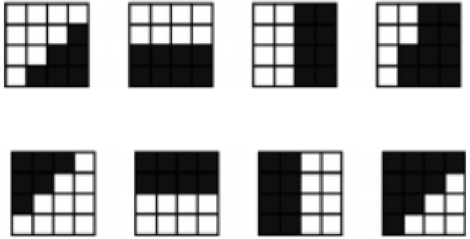
Features: arrangement of tags



Arrangements: 8 angles

tags

Common 4x4 patterns



A subset of all the 62 tags

#Features: 62x62x8 = 30,752

Single tree: **7.0%** error

Random forest of 25 trees: **0.8%** error

Let's try this out