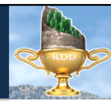
Data Mining



Classifier Ensemble

Adapted from R. Polikar's "Ensemble Based Systems in Decision Making"

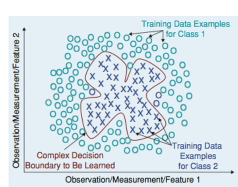
Outline

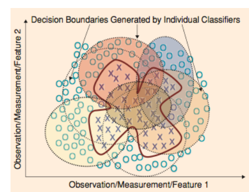
- Why Ensemble?
- Components in an ensemble
- Ensemble Based Systems
  - Bagging
  - Boosting
    - AdaBoost
- How much classification improvements with an ensemble?

Why Ensemble?

- Making important decisions
  - Expert panel, Lifeline (opinion of expert "friends")
- Reasons:
  - Statistical reason
    - Training vs. generalization
    - Algorithm/System vs. data
  - Large volumes of data
  - Too little data
  - Divide and conquer
  - Data fusion

Why Ensemble?

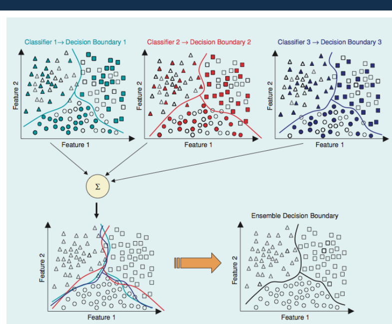




Divide and Conquer

Averaging over an ensemble of classifiers

Components in an Ensemble



- Good candidate base classifiers:
  - Neural Networks
  - Decision Tree
  - KNN
  - …

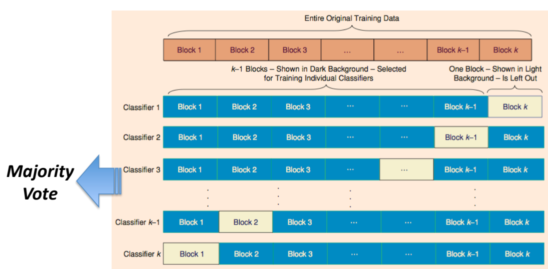
Two components in an Ensemble

- A strategy to create the classifiers
- A strategy to combine the classifiers

Components in an Ensemble

- A strategy to build an ensemble as diverse as possible
  - Disjoint data -- k-fold data splitting
  - Bagging
  - Boosting

K-fold data splitting



Bagging

- Main idea:
  - Form different training data to train each classifier
    - How?
    - Create bootstrapped training data-- randomly pick a certain percent of data from the original data with replacement
  - The classifiers are combined with majority vote

Bagging

Algorithm: Bagging

Input:
- Training data S with correct labels $\omega_i \in \Omega = \{\omega_1,\ldots,\omega_C\}$ representing C classes
- Weak learning algorithm **WeakLearn,**
- Integer T specifying number of iterations.
- Percent (or fraction) F to create bootstrapped training data

Do $t = 1, \ldots, T$
1. Take a bootstrapped replica S_t by randomly drawing F percent of S.
2. Call **WeakLearn** with S_t and receive the hypothesis (classifier) h_t.
3. Add h_t to the ensemble, $\mathbb{E}$.

End

Test: Simple Majority Voting – Given unlabeled instance **x**
1. Evaluate the ensemble $\mathbb{E} = \{h_1, \ldots, h_T\}$ on **x**.
2. Let $$v_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$
   be the vote given to class ω_j by classifier h_t.
3. Obtain total vote received by each class
   $$V_j = \sum_{t=1}^{T} v_{t,j}, \quad j = 1,\ldots,C \quad (9)$$
4. Choose the class that receives the highest total vote as the final classification.

Boosting

- Considered as one of the most important developments in the recent history of machine learning.
- Freund and Schapire, 1997
- Many variations
  - General boosting
  - Adaboost
    - Adaboost.R
    - Adaboost.M1

General Boosting

- Basic ideas: create three weak classifiers:
  - classifier C1 trained with a random subset of the available training data.
  - C2 is trained on a training data only half of which is correctly classified by C1 ,and the other half is misclassified.
  - The third classifier C3 is trained with instances on which C1 and C2 disagree.
  - The three classifiers are combined through a three-way majority vote.
- No replacement allowed

General Boosting

Algorithm: Boosting

Input:

- Training data S of size N with correct labels $\omega_i \in \Omega = \{\omega_1, \omega_2\}$;
- Weak learning algorithm **WeakLearn**.

Training

1. Select $N_1 < N$ patterns without replacement from S to create data subset S_1.
2. Call **WeakLearn** and train with S_1 to create classifier C_1.
3. Create dataset S_2 as the most informative dataset, given C_1, such that half of S_2 is correctly classified by C_2, and the other half is misclassified. To do so:
   a. Flip a fair coin. If Head, select samples from S, and present them to C_1 until the first instance is misclassified. Add this instance to S_2.
   b. If Tail, select samples from S, and present them to C_1 until the first one is correctly classified. Add this instance to S_2.
   c. Continue flipping coins until no more patterns can be added to S_2.
4. Train the second classifier C_2 with S_2.
5. Create S_3 by selecting those instances for which C_1 and C_2 disagree. Train the third classifier C_3 with S_3.

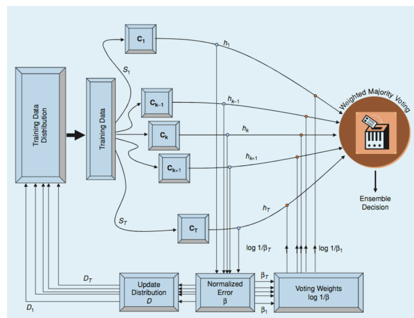
Test – Given a test instance **x**

1. Classify **x** by C_1 and C_2. If they agree on the class, this class is the final classification.
2. If they disagree, choose the class predicted by C_3 as the final classification.

AdaBoost

- Basic ideas:
  - Consecutive classifiers' training data are geared towards increasingly hard-to-classify instances.
    - How?
    - Train each weak classifier using instances drawn from an iteratively updated distribution of the training data.
    - This distribution update ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier.
  - Weighted majority vote

Adaboost



AdaBoost

Algorithm AdaBoost.M1

Input:

- Sequence of N examples $S = [(\mathbf{x}_i, y_i)], i = 1, \cdots, N$ with labels $y_i \in \Omega, \Omega = \{\omega_1, \ldots, \omega_C\}$;
- Weak learning algorithm **WeakLearn**;
- Integer T specifying number of iterations.

Initialize $D_1(i) = \frac{1}{N}, i = 1, \cdots, N$ (11)

Do for $t = 1, 2, \ldots, T$:

1. Select a training data subset S_t, drawn from the distribution D_t.
2. Train **WeakLearn** with S_t, receive hypothesis h_t.
3. Calculate the error of h_t: $\varepsilon_t = \sum_{i: h_t(\mathbf{x}_i) \neq y_i} D_t(i)$. (12)

   If $\varepsilon_t > 1/2$, **abort**.
4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$. (13)
5. Update distribution

$$D_t : D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(\mathbf{x}_i) = y_i \\ 1, & \text{otherwise} \end{cases} \quad (14)$$

   where $Z_t = \sum_i D_t(i)$ is a normalization constant chosen so that D_{t+1} becomes a proper distribution function.

Test – Weighted Majority Voting: Given an unlabeled instance **x**,

1. Obtain total vote received by each class

$$V_j = \sum_{t: h_t(\mathbf{x}) = \omega_j} \log \frac{1}{\beta_t}, j = 1, \ldots, C. \quad (15)$$

2. Choose the class that receives the highest total vote as the final classification.

Boosting

- It has been proven: "the error of this three-classifier ensemble is <u>bounded</u> above, and it is <u>less than</u> the error of the best classifier in the ensemble, provided that each classifier has an error rate that is less than 0.5."

Two components in an Ensemble

- A strategy to create the classifiers
- A strategy to combine the classifiers

Combining Classifiers

- A strategy to combine the output of individual classifiers → amplify correct decisions and cancel out incorrect ones:
  - Majority vote
  - Weighted majority vote
  - Combining numeric outputs
  - Others:
    - Behavior knowledge space, Borda count