World Changers Shaped Here

SMU

A Survey of Ensemble Classification

Outline

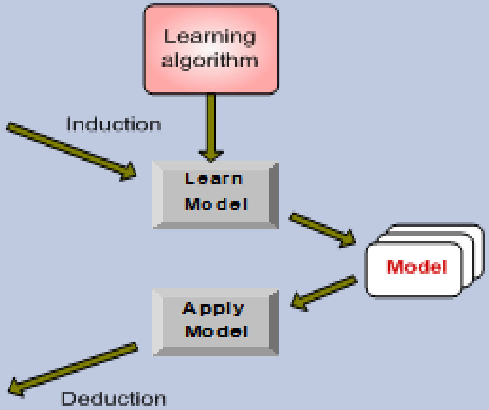
- Definition of Classification and an overview of Base Classifiers
- Ensemble Classification
  - Definition and Rational
  - Properties of Ensemble Classifiers
  - Building Blocks of an Ensemble Classifier
  - Combining Methods
  - Types of Ensemble Classifiers
  - A simple example of building an Ensemble Classifier using R

Classification

- **Definition**: Given a dataset $D=\{t_1,t_2,\ldots,t_n\}$ and a set of classes $C=\{C_1,\ldots,C_m\}$, the Classification Problem is to define a mapping function $f:D \rightarrow C$ where each t_i is assigned to a single class C.

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Source: Tan, Steinbach , and Kumar

An Overview of Common Base Classifiers

- **Logistics Regression** – Classification via extension of the idea of linear regression to situations where outcome variables are categorical.
- **Nearest Neighbor** – Classification of objects via a majority vote of its neighbors, with the object being assigned to the class most common.
- **Decision Tree Induction** – Classification via a divide and conquer approach that creates structured nodes and leafs from the dataset.
- **Rule-based Methods** – Classification by use of an ordered set of rules.
- **Naïve Bayes Methods** – Probabilistic methods of classification based on Bayes Theorem
- **Support Vector Machines** – Use of hyper-planes to separate different instances into their respective classes.

Ensemble Classifiers

Ensemble classification refers to a collection of methods that learn a target function by training a number of individual learners and combining their predictions.

- **Rational:** ‘No Free Lunch’ Theorem
  - Even popular base classifiers will perform poorly on some datasets, where the learning classifier and data distribution do not match well
- **Intuitive Justification:**
  - When combing multiple, independent, and diverse, decisions each of which is at least more accurate than random guessing then random errors cancel each other out, and correct decisions are reinforced

Statistical Justification

- **Binomial Distribution**: The probability of observing x heads in a sample of n independent coin tosses, where in each toss the probability of heads is p, is

$$P(X = x \mid p, n) = \frac{n!}{r!(n-x)!} p^x (1-p)^{n-x}$$

- **Example:**
- Suppose there are 25 independent base classifiers
- Each classifier has error rate, p = 0.35
- The probability that the ensemble classifier make’s a wrong prediction is 0.06

$$\sum_{i=13}^{25} \binom{25}{i} p^i (1-p)^{25-i} = 0.06$$

Source: Tan, Steinbach , and Kumar

Justification by Bias – Variance Decomposition

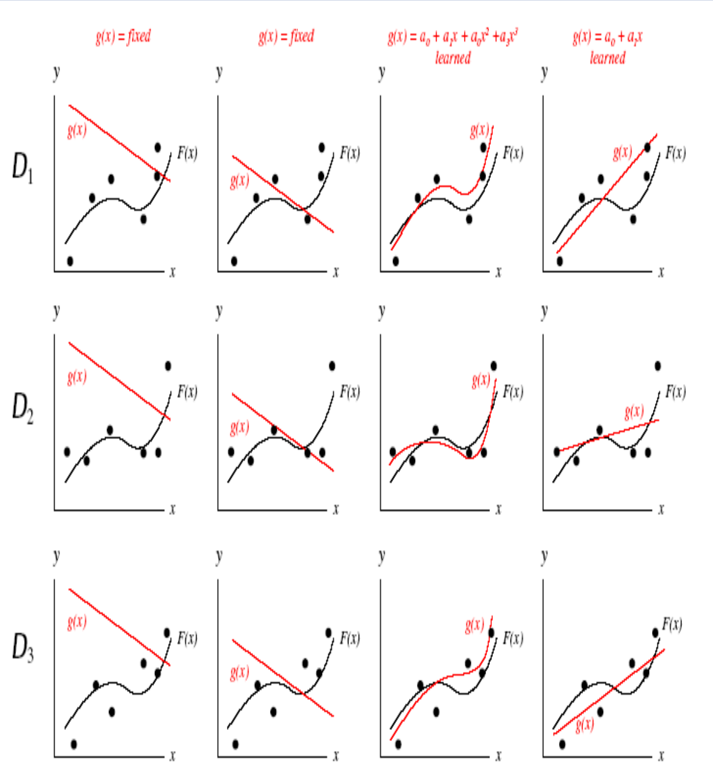
- The expected error ***d*** of a learning algorithm can be decomposed into **Bias**, **Variance** and **Noise**.
- **Bias** measures how closely the average classifier produced by the learning algorithm matches the target function – <u>measures the quality of the match</u>
  - High-bias implies poor match
- **Variance** measures how much the learning algorithm's predictions fluctuate for different training sets (of the same size) – <u>measures the specificity of the match</u>
  - High-variance implies a weak match
- An intrinsic target **noise**, is the minimum error that can be achieved and is that of the Bayes optimal classifier

$$d_{f,\theta}(y,t) = Bias_{\theta} + Variance_{f} + Noise_{t}$$

Bias – Variance Dilemma

- **Flexible Base Classifiers** adapt to training data and have lower bias, but higher variance
  - Fits well to dataset and have low bias, but high variance
- **Inflexible Base Classifiers** have higher bias, but lower variance
  - May not fit well to data: have high bias, but low variance

Hence the need for Ensemble Classifiers



Col 1:
Poor fixed linear model
High bias, zero variance
Col 2:
Slightly better fixed linear model;
Lower (but high) bias, zero variance.
Col 3:
Learned cubic model;
Low bias, moderate variance.
Col 4:
Learned linear model;
Intermediate bias and variance.

Properties of Ensemble Classifiers

- **Diversity of Opinion** – Multiple base classifiers should be available and capable of making classifications on a dataset
- **Independence** – Any Base Classifier's decisions is not influenced by any other Base Classifier.
- **Decentralization** – Base Classifiers can be allowed to specialize on a specific subset of the dataset
- **Aggregation** – Some combining method exist for turning private judgments into a collective decision

Elements of an Ensemble Classifier

A typical ensemble method for classification contains the following building blocks

- **Training Set** – A labeled dataset used to train
- **Base Classifier(s)** – An induction algorithm that obtains a training set and forms a classifier that represents a generalized attribute between input attribute and the target attribute
- **Diversity Generator** – This component is responsible for generating the diverse classifiers
- **Combiner** – The combiner is responsible for combining the classifications of the various classifiers

Diversity Generation

- Diversified classifiers lead to uncorrelated classifications which in turn improve accuracy.
- The most common methods of diversifying are:
  - Manipulating the Training Sample
  - Manipulating the learner
  - Changing the target attribute representation
  - Hybridization

Combing Methods

There are two main methods of combining the Base Classifiers' output – **weighting methods** and **meta-learning methods**

- **Weighting** methods are best if the Base Classifiers have comparable success
- **Meta-learning** methods are suited for cases in which certain classifiers consistently correctly classify or consistently misclassify certain instances

Common Weighting Methods

- **Majority Voting** – Classification of an unlabeled instance is performed according to the class that contains the highest number of votes
- **Performance Weighting** – The weight of each classifier can be set proportionally to its accuracy performance on a validation set.
- **Bayesian Combination** – The weight associated with each classifier is the posterior probability given the training set.
- **Vogging** – To optimize linear combination of base-classifiers so as to aggressively reduce variance while attempting to preserve a prescribed accuracy

Common Meta-combination Methods

Meta-learning is defined as learning form the classifications produced by the learner and from the classification of these classifiers on training data.

- **Stacking** – This method attempts to induce which classifiers are reliable an which are not.
- **Grading** – This method uses ‘graded’ classifications as the meta-level class.

Dependent Framework

In a **dependent framework** the output of a base classifier is used in the construction of the next classifier.

- There are two main approaches for dependent learning:
  - **Incremental Batch Learning** – The classification produced in one iteration is given as ‘prior knowledge’ to the learning algorithm in the following iteration:
  - **Model-guided Instance Selection** – The classifiers that were constructed in the previous iterations are used for manipulating a training set of the iteration. <u>Examples</u>**: Boosting, AdaBoosting.**

Dependent Example: AdaBoost

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
- Initially, all N records are assigned equal weights
- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

AdaBoosting Algorithm

Algorithm AdaBoost.M1

Input :

Sequence of N examples $S = [(x_i, y_i)], i = 1, \cdots, N$ with labels $y_i \in \Omega, \Omega = \{\omega_1, \cdots \omega_C\}$;

Weak learning algorithm **WeakLearn**;

Integer T specifying number of iterations

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

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

1. Select a training data subset S_t, draw from the distribution D_t.
2. Train **WeakLearn** with S_t, receive hypothesis h_t.
3. Calculate the error of

$$h_t : \varepsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$$

If $\varepsilon_t > 1/2$, **abort**

4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$.
5. Update distribution

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

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

Test - Weight Majority Voting: Given an unlabeled instance x.

1. Obtain total vote received by each class

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

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

Source: Rokach

Independent Framework

- In an **independent** framework all classifiers within the ensemble learn independently and their outputs are combined in some fashion.
- The original dataset is transformed into several datasets from which several classifiers are trained
- A combination method is then applied in order to output the final classification
- The independent framework is independent of learning algorithms hence different learners can be used on each data set.
- <u>Examples</u>: **Bagging**, **Random Forest, Mixture of Experts (ME)**

Independent Example: Bagging

- **Bagging** creates an ensemble by training individual classifiers on bootstrap samples of the training set
- Training subsets are randomly drawn - with replacement - from the entire dataset
- For a dataset with N entities, each entity has a probability of $1-(1-1/N)^N$ of being selected at least once in the N samples
- Each re-sampled training set is used to train a different Base Classifier
- Individual classifiers are combined by taking a majority vote of their decisions

Bagging Algorithm

Bagging

input :

Training data S with correct labels $\omega_i \in \Omega = \{\omega_1, \cdots, \omega_C\}$ representi ng C classes

Weak learning algorithm **WeakLearn** ,

Integer T specifying number of iterations .

Percent (of fraction) F to create bootstrapp ed training data

Do $t = 1, \cdots, T$

1.Take a bootstrapp ed replica S_t by randomly drawing percent of S .

2.Call **WeakLearn** with S_t and receive the hypothesis (classifie r) h_t.

3.Add h_t to the ensemble, E .

End

Test : Simple Majority Voting - Given unlabled instance x

1.Evaluate the ensemble on x .

2.Let $v_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \\ 0, & \text{otherwise} \end{cases}$ be the vote given to class by classifier .

3.Obtain t otal vote received by each class $V_j = \sum_{t=1}^{T} v_{t,j}$, $j = 1, \cdots, C$

4.Choose the class that receives the highest to tal vote as the final classifica tion.

Source: Rokach

Other Common Ensembles

- **Random Subspace** – Each Base Classifier uses only a subset of all features for training and testing
- **Class Switching** – Each new training set is obtained by randomly switching the classes of the training examples
- **Rotation Forest** – Bootstrap samples are drawn and principle component analysis PCA is performed
- **Hybrid Adaptive Classifiers** – Base Classifiers compete (adapt) to find ideal classifications within a random subspace
- **Ensemble of Ensembles** – Using other ensembles to create more accurate classifiers

A Simple Example

Tutorial Class Example.R

Background

- Classify the number of cylinders of each vehicle from a dataset containing multiple attributes.
- **Recall** the elements of an ensemble: 1. Training Set, 2. Base learners, 3. Diversity Generator, 4. Combiner

1	Training Set	Vehicle Attributes
2	Base learners	gbm, rpart, treebag
3	Diversification	Hybridization/ensemble of ensembles
4	Combining Method	Performance Weighting
5	Framework	Independent

https://www.youtube.com/watch?v=k7sTiTWWCXM

Source: Manuel Amunategui

Questions

References

- **Manuel Amunategui;** - http://amunategui.github.io/blending-models/, 02-22-2015
- **Tan, Kumar, Steinbach**; Introduction to Data Mining, Pearson, 2013
- **Lior Rockach;** Ensemble-based classifiers, Springer, 2009
- **Amasyali, Ersoy;** Comparison of Single and Ensemble Classifiers in Terms of Accuracy and Execution Time
- **Yu, Liu;** Hybrid Adaptive Classifier Ensemble, IEEE Transaction on Cybernetics, 2015
- **Duangsoithong, Wiindeatt;** Relevance and Redundancy Analysis for Ensemble Classifiers, Springer, 2009
- **Sumana, Santhanam;** An Empirical Comparison of Ensemble and Hybrid Classification, Association of Computer and Electrical Engineers, 2014
- **Thalor, Patil;** Comparison of Ensemble Based Classification Algorithms, IJARCSSE, 2014