In The Name of Allah

Digital Media Laboratory
Sharif University of Technology

Statistical Pattern Recognition

Combining Classifiers

Hamid R. Rabiee
Jafar Muhammadi, Alireza Ghasemi

Spring 2012
http://ce.sharif.edu/courses/90-91/2/ce725-1/

Agenda

- **Combining Classifiers**
  - **Empirical view**
  - **Theoretical view**
- **Resampling**
  - **Bagging**
  - **Boosting**
    - Adaboost

Combining Classifiers (Empirical View)

- **Just like different features capturing different properties of a pattern, different classifiers also capture different structures and relationships of these patterns in the feature space.**
- **An empirical comparison of different classifiers can help us choose one of them as the best classifier for the problem at hand.**

Combining Classifiers (Empirical View)

- **However, although most of the classifiers may have similar error rates, sets of patterns misclassified by different classifiers do not necessarily overlap.**
- **Not relying on a single decision but rather combining the advantages of different classifiers is intuitively promising to improve the overall accuracy of classification.**
- **Such combinations are variously called *combined classifiers*, *ensemble classifiers*, *mixture-of-expert models*, or *pooled classifiers*.**

Combining Classifiers (Empirical View)

- **Some reasons for combining multiple classifiers to solve a given classification problem can be stated as follows:**
  - **Access to different classifiers, each developed in a different context and for an entirely different representation/description of the same problem.**
  - **Availability of multiple training sets, each collected at a different time or in a different environment, even may use different features.**
  - **Local performances of different classifiers where each classifier may have its own region in the feature space where it performs the best.**
  - **Different performances due to different initializations and randomness inherent in the training procedure.**

Combining Classifiers (Theoretical View)

- **At a single data point the quadratic error of the ensemble $(f_{ens}-d)^2$ is less than or equal to the average quadratic error of individuals $(f_i-d)^2$:**

$$(f_{ens}-d)^2 = \sum_i w_i (f_i-d)^2 - \sum_i w_i (f_i - f_{ens})^2$$

$$\text{Where: } f_{ens} = \sum_i w_i f_i$$

  - **The first term is the weighted average error of individuals.**
  - **The second term is the diversity term, measuring the amount of variability among the ensemble member answers for this pattern.**

Combining Classifiers (Theoretical View)

- **It tells us that taking the combination of several predictors would be better on average over several patterns, than a method which selected one of the predictors at random.**
- **We need to get the right balance between diversity (the diversity term) and individual accuracy (the average error term), in order to achieve lowest overall ensemble error.**
- **All successful ensemble methods encourage diversity to some extent.**

Combining Classifiers

- **In summary, we may have different feature sets, training sets, classification methods, and training sessions, all resulting in a set of classifiers whose outputs may be combined.**
- **Combination architectures can be grouped as:**
  - ***Parallel*: all classifiers are invoked independently and then their results are combined by a combiner.**
  - ***Serial (cascading)*: individual classifiers are invoked in a linear sequence where the number of possible classes for a given pattern is gradually reduced.**
  - ***Hierarchical (tree)*: individual classifiers are combined into a structure, which is similar to that of a decision tree, where the nodes are associated with the classifiers.**

Combining Classifiers

- **Selecting and training of individual classifiers:**
  - Combination of classifiers is especially useful if the individual classifiers are largely independent.
  - This can be explicitly forced by using different training sets, different features and different classifiers.
- **Combiner:**
  - Some combiners are static, with no training required, while others are trainable.
  - Some are adaptive where the decisions of individual classifiers are evaluated (weighed) depending on the input pattern, whereas non-adaptive ones treat all input patterns the same.
  - Different combiners use different types of output from individual classifiers: confidence, rank, or abstract.

Combining Classifiers

- **Examples of classifier combination schemes are:**
  - **Majority voting (each classifier makes a binary decision (vote) about each class and the final decision is made in favor of the class with the largest number of votes),**
  - **Sum, product, maximum, minimum and median of the posterior probabilities computed by individual classifiers,**
  - **Class ranking (each class receives m ranks from m classifiers, the highest (minimum) of these ranks is the final score for that class),**
  - **Weighted combination of classifiers.**
- **We will study different combination schemes using a Bayesian framework and resampling.**

Resampling

- **Resampling is well-known method for generating training data and evaluating the accuracy of different classifiers.**
- **It can also be used to build classifier ensembles.**
- **We will study:**
  - ***bagging*, where multiple classifiers are built by bootstrapping the original training set, and**
  - ***boosting*, where a sequence of classifiers is built by training each classifier using data sampled from a distribution derived from the empirical misclassification rate of the previous classifier.**

Bagging

- ***Bagging (bootstrap aggregating)*** **uses multiple versions of the training set, each created by bootstrapping the original training data.**
- **Each of these bootstrap data sets is used to train a different component classifier.**
- **The final classification decision is based on the vote of each component classifier.**
- **Traditionally, the component classifiers are of the same general form (e.g., all neural networks, all decision trees, etc.) where their differences are in the final parameter values due to their different sets of training patterns.**

Bagging

- A classifier/learning algorithm is informally called unstable if small changes in the training data lead to significantly different classifiers and relatively large changes in accuracy.
- Decision trees and neural networks are examples of unstable classifiers where a slight change in training patterns can result in radically different classifiers.
- In general, bagging improves recognition for unstable classifiers because it effectively averages over such discontinuities.

Boosting

- In boosting, each training pattern receives a weight that determines its probability of being selected for the training set for an individual component classifier.
- If a training pattern is accurately classified, its chance of being used again in a subsequent component classifier is reduced.
- Conversely, if the pattern is not accurately classified, its chance of being used again is increased.
- The final classification decision is based on the weighted sum of the votes of the component classifiers where the weight for each classifier is a function of its accuracy.

Adaboost

- **The popular *AdaBoost (adaptive boosting)* algorithm allows continuous adding of classifiers until some desired low training error has been achieved.**
  - **Let $\alpha^t(x_i)$ denote the weight of pattern x_i at trial t, where $\alpha^1(x_i) = 1/n$ for every x_i.**
  - **At each trial t=1,...,T , a classifier C^t is constructed from the given patterns under the distribution α^t where $\alpha^t(x_i)$ reflects occurrence probability of x_i.**
    - The error ε^t of this classifier is also measured with respect to the weights, and consists of the sum of the weights of the patterns that it misclassifies.
      - **If ε^t is greater than 0.5, the trials terminate and T is set to t−1.**
      - **Conversely, if C^t correctly classifies all patterns so that ε^t is zero, the trials also terminate and T becomes t.**
      - **Otherwise, the weights α^{t+1} for the next trial are generated by multiplying the weights of patterns that C^t classifies correctly by the factor $\beta^t = \varepsilon^t/(1-\varepsilon^t)$ and then are renormalized so that $\Sigma^n_{i=1}\ \alpha^t(x_i) = 1$.**
  - **The boosted classifier C* is obtained by summing the votes of the classifiers $C^1,...,C^T$, where the vote for classifier C^t is also weighted by $\log(1/\beta^t)$.**

Adaboost

- Provided that ε^t is always less than 0.5, it was shown that the error rate of C^* on the given patterns under the original uniform distribution α^1 approaches zero exponentially quickly as T increases.
- A succession of weak classifiers $\{C^t\}$ can thus be boosted to a strong classifier C^* that is at least as accurate as, and usually much more accurate than, the best weak classifier on the training data.
- However, note that there is no guarantee of the generalization performance of a bagged or boosted classifier on unseen patterns.

Any Question?

End of Lecture 16

Thank you!

Spring 2012

http://ce.sharif.edu/courses/90-91/2/ce725-1/