Classification - Ensemble Methods

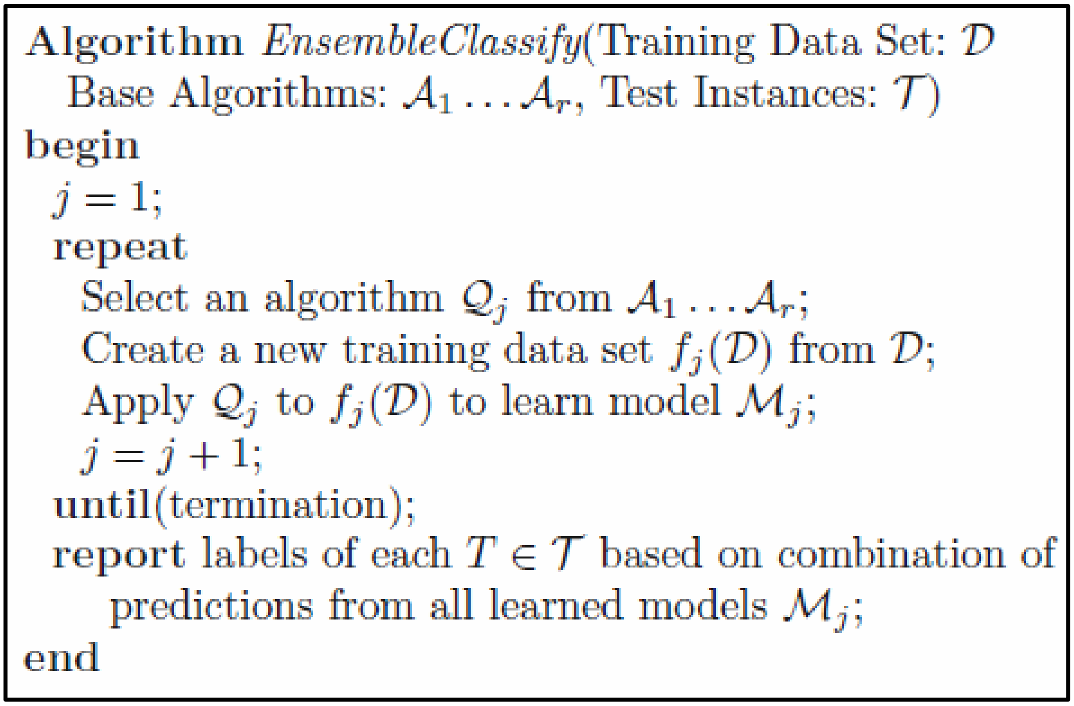
Ensemble methods are motivated by the fact that different classifiers may make different predictions on test instances due to the specific characteristics of the classifier, or their sensitivity to the random artifacts in the training data.

An ensemble method is ***an approach to increase the prediction accuracy by combining the results from multiple classifiers***.

The basic approach of ensemble analysis is to ***apply the base ensemble learners multiple times by using either different models, or by using the same model on different subsets of the training data***.

The ***results from different classifiers are then combined into a single robust prediction***.

There are significant differences in how the individual learners are constructed and combined by various ensemble models.



```
Algorithm $EnsembleClassify$(Training Data Set: $\mathcal{D}$
  Base Algorithms: $\mathcal{A}_1 \ldots \mathcal{A}_r$, Test Instances: $\mathcal{T}$)
begin
  $j = 1$;
  repeat
    Select an algorithm $\mathcal{Q}_j$ from $\mathcal{A}_1 \ldots \mathcal{A}_r$;
    Create a new training data set $f_j(\mathcal{D})$ from $\mathcal{D}$;
    Apply $\mathcal{Q}_j$ to $f_j(\mathcal{D})$ to learn model $\mathcal{M}_j$;
    $j = j + 1$;
  until(termination);
  report labels of each $T \in \mathcal{T}$ based on combination of
      predictions from all learned models $\mathcal{M}_j$;
end
```

The ensemble approach uses ***a set of base classification algorithms $A_1 \ldots A_r$***.

These learners might be completely different algorithms, such as decision trees, SVMs, or the Bayes classifier.

In some types of ensembles, such as ***boosting and bagging, a single learning algorithm is used but with different choices of training data***.

Different learners are used to leverage the greater robustness of different algorithms in different regions of the data. Let the learning algorithm selected in the j^{th} iteration be denoted by Q_j.

It is assumed that Q_j is selected from the base learners.

At this point, a *derivative training data set* $f_j(D)$ from the base training data is selected.

This may be a random sample of the training data, as in bagging, or it may be based on the results of the past execution of ensemble components, as in boosting.

A model M_j is learned in the j^{th} iteration by applying the selected learning algorithm Q_j to $f_j(D)$.

For each test instance T, a prediction is made by combining the results of different models M_j on T.

This combination may be performed in various ways.

Examples include the use of simple averaging, the use of a weighted vote, or the treatment of the model combination process as a learning problem.

The description of the algorithm is very generic, and allows significant flexibility in terms of how the ensemble components may be learned and the combination may be performed.

The two primary types of ensembles are special cases of the description.

1.<u>Data-Centered Ensembles:</u>

- A single base learning algorithm (e.g., an SVM or a decision tree) is used
- The primary variation is in terms of how the derivative data set *fj*(*D*) for the *j*th ensemble component is constructed.

- In this case, ***the input to the algorithm contains only a single learning algorithm A_1***.
- The data set $f_j(D)$ for the j^{th} component of the ensemble may be constructed by sampling the data, focusing on incorrectly classified portions of the training data in previously executed ensemble components, manipulating the features of the data, or manipulating the class labels in the data.

2. <u>Model-Centered Ensembles:</u>

- Different algorithms Q_j are used in each ensemble iteration.
- In these cases, the data set $f_j(D)$ for each ensemble component is the same as the original data set D.
- The ***rationale for these methods is that different models may work better in different regions of the data***, and therefore the combination of the models may be more effective for any given test instance, as long as the specific errors of a classification algorithm are not reflected by the majority of the ensemble components on any particular test instance.

Why Does Ensemble Analysis Work?

The rationale for ensemble analysis can be best understood by examining the different components of the error of a classifier, as discussed in statistical learning theory.

There are ***three primary components to the error of a classifier***:

1. Bias:

Every classifier makes its own modeling assumptions about the nature of the decision boundary between classes.

Example: a linear SVM classifier assumes that the two classes may be separated by a linear decision boundary.

This is, of course, not true in practice.

The decision boundary between the different classes may clearly be non-linear.

Therefore, no (linear) SVM classifier can classify all the possible test instances correctly even if the best possible SVM model is constructed with a very large training data set.

The SVM classifier may seem to be the best possible approximation, it obviously cannot match the correct decision boundary and therefore has an inherent error.

In other words, any given linear SVM model will have an inherent ***bias***.

When a classifier has high bias, it will make consistently incorrect predictions over particular choices of test instances near the incorrectly modeled decision-boundary, even when different samples of the training data are used for the learning process.

2. Variance: Random variations in the choices of the training data will lead to different models.

Example: In this case, the true decision boundary is linear.

A *sufficiently deep* univariate decision tree can approximate a linear boundary quite well with axis-parallel piecewise approximations.

However, *with limited training data*, even when the trees are grown to full depth without pruning, the piecewise approximations will be coarse like the boundaries.

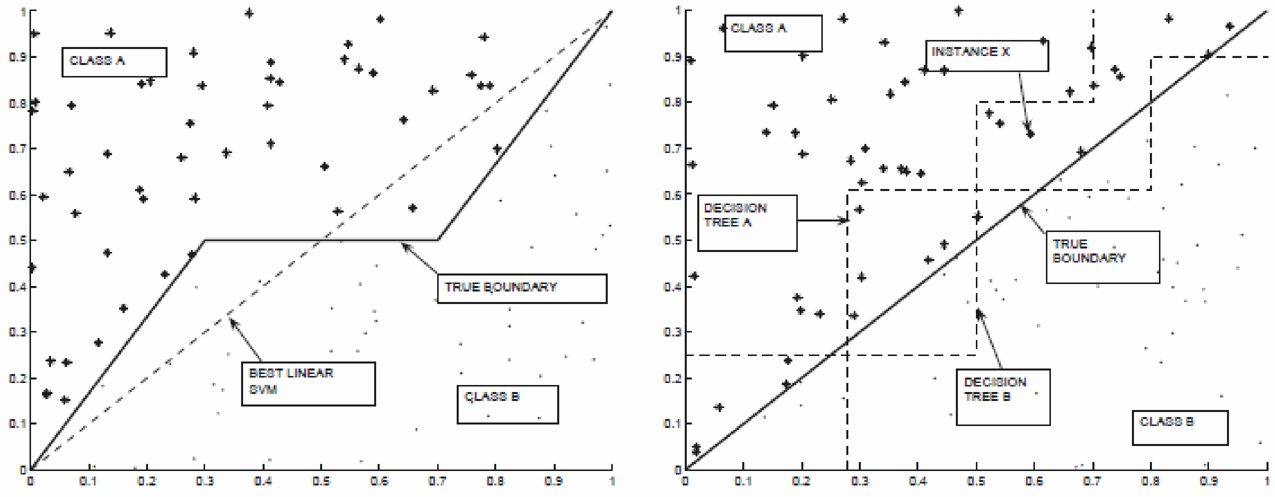
Different choices of training data might lead to different split choices, as a result of which the decision boundaries of trees A and B are very different.

Therefore, (test) instances such as X are *inconsistently classified* by decision trees which were created by different choices of training data sets.

This is a manifestation of model *variance*.

Model variance is closely related to overfitting.

When a classifier has an overfitting tendency, it will make *inconsistent* predictions for the same test instance over different training data sets.



(a) bias (b) variance

Figure 11.5: Impact of bias and variance on classification accuracy

3. Noise:

- The noise refers to the ***intrinsic errors in the target class labeling***.
- Because this is an intrinsic aspect of data quality, ***there is little that one can do to correct it***.
- Therefore, ***the focus of ensemble analysis is generally on reducing bias and variance***.

Note that the design choices of a classifier often reflect a trade-off between the bias and the variance.

Example: (i) Pruning a decision tree results in a more stable classifier and therefore reduces the variance. On the other hand, because the pruned decision tree makes stronger assumptions about the simplicity of the decision boundary than the unpruned tree, the former leads to greater bias.

(ii) Using a larger number of neighbors for a nearest-neighbor classifier will lead to larger bias but lower variance.

In general, simplified assumptions about the decision boundary lead to greater bias but lower variance. On the other hand, complex assumptions reduce bias but are harder to robustly estimate with limited data.

The bias and variance are affected by virtually every design choice of the model, such as the choice of the base algorithm or the choice of model parameters.

Ensemble analysis can often be used to reduce both the bias and variance of the classification process.

In general, different classification models have different sources of bias and variance.

Models that are too simple (such as a linear SVM or shallow decision tree) make too many assumptions about the shape of the decision boundary, and will therefore have high bias.

Models that are too complex (such as a deep decision tree) will overfit the data, and will therefore have high variance.

Sometimes a different parameter setting in the same classifier will favor different parts of the bias-variance trade-off curve.

For example, a small value of *k* in a nearest-neighbor classifier will result in lower bias but higher variance.

Because different kinds of ensembles learners have different impacts on bias and variance, it is important to choose the component classifiers, so as to optimize the impact on the bias-variance trade-off.

An overview of the impact of different models on bias and variance is provided in Table below.

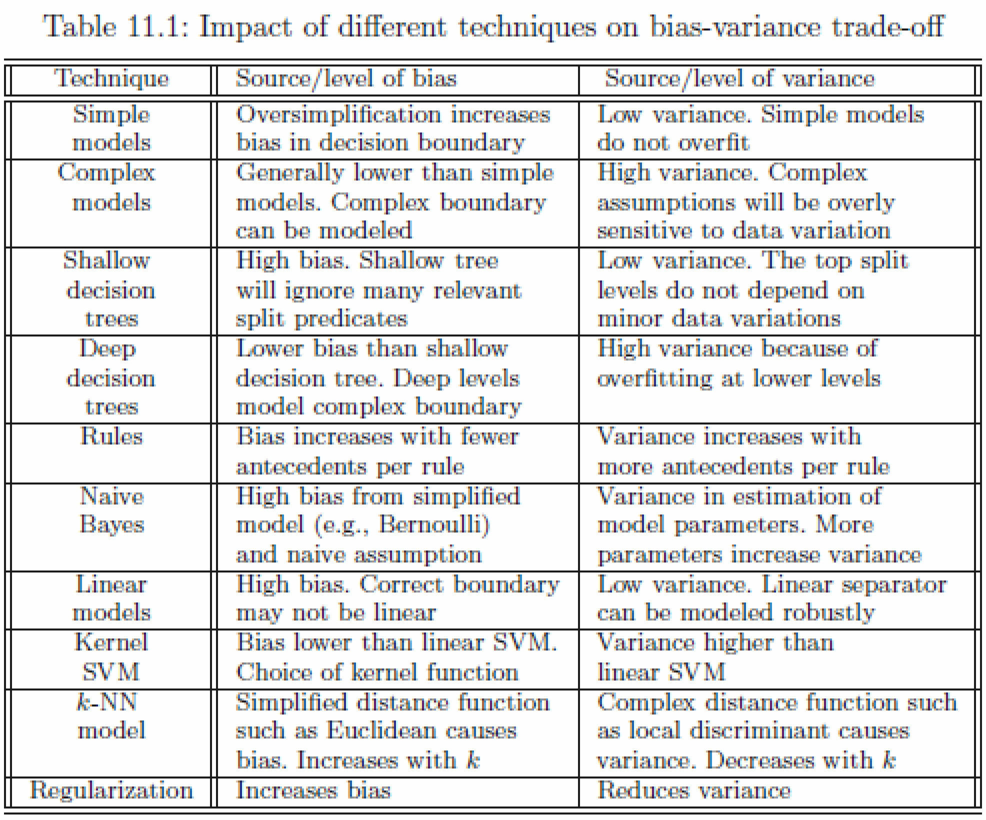


Table 11.1: Impact of different techniques on bias-variance trade-off

Technique	Source/level of bias	Source/level of variance
Simple models	Oversimplification increases bias in decision boundary	Low variance. Simple models do not overfit
Complex models	Generally lower than simple models. Complex boundary can be modeled	High variance. Complex assumptions will be overly sensitive to data variation
Shallow decision trees	High bias. Shallow tree will ignore many relevant split predicates	Low variance. The top split levels do not depend on minor data variations
Deep decision trees	Lower bias than shallow decision tree. Deep levels model complex boundary	High variance because of overfitting at lower levels
Rules	Bias increases with fewer antecedents per rule	Variance increases with more antecedents per rule
Naive Bayes	High bias from simplified model (e.g., Bernoulli) and naive assumption	Variance in estimation of model parameters. More parameters increase variance
Linear models	High bias. Correct boundary may not be linear	Low variance. Linear separator can be modeled robustly
Kernel SVM	Bias lower than linear SVM. Choice of kernel function	Variance higher than linear SVM
k-NN model	Simplified distance function such as Euclidean causes bias. Increases with k	Complex distance function such as local discriminant causes variance. Decreases with k
Regularization	Increases bias	Reduces variance