Ensemble Based Sytems in Datamining Application

Janis Schönefeld

Universität Hamburg, MIN-Fakultät

8. Mai 2007

Table of contents

Table of contents

Table of contents

Table of contents

Table of contents

What is an ensemble based system for classification

- An ensemble is a set of classifiers trained for the “same” domain.
- Different types of classifiers may be used in one ensemble.
- Decisions are based on the classifications made by the members of the set using different functions

What is an ensemble based system for classification

- An ensemble is a set of classifiers trained for the “same” domain.
- Different types of classifiers may be used in one ensemble.
- Decisions are based on the classifications made by the members of the set using different functions

What is an ensemble based system for classification

- An ensemble is a set of classifiers trained for the “same” domain.
- Different types of classifiers may be used in one ensemble.
- Decisions are based on the classifications made by the members of the set using different functions

Ensembles are a good choice,

- with respect to computational learning theory
- for statistical reasons (error reduction through generalisation)
- if there are to complex boundaries for linear or circular classifiers
- when there is to little data
- when there is to much data
- when one needs to do data fusion

Ensembles are a good choice,

- with respect to computational learning theory
- for statistical reasons (error reduction through generalisation)
- if there are to complex boundaries for linear or circular classifiers
- when there is to little data
- when there is to much data
- when one needs to do data fusion

Ensembles are a good choice,

- with respect to computational learning theory
- for statistical reasons (error reduction through generalisation)
- if there are to complex boundaries for linear or circular classifiers
- when there is to little data
- when there is to much data
- when one needs to do data fusion

Ensembles are a good choice,

- with respect to computational learning theory
- for statistical reasons (error reduction through generalisation)
- if there are to complex boundaries for linear or circular classifiers
- when there is to little data
- when there is to much data
- when one needs to do data fusion

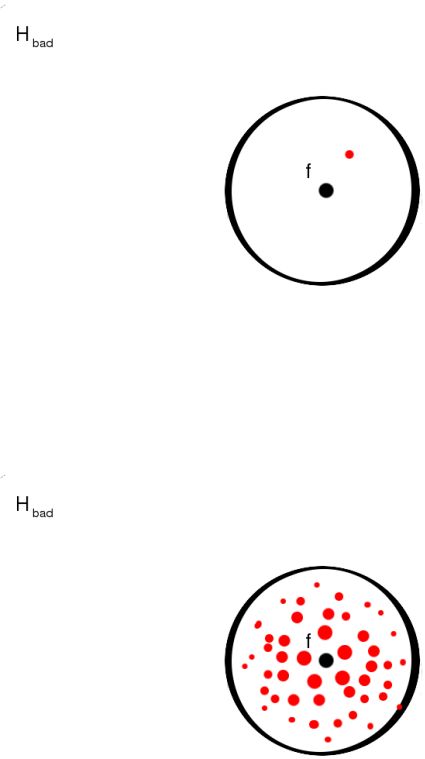
Ensembles are a good choice,

- with respect to computational learning theory
- for statistical reasons (error reduction through generalisation)
- if there are to complex boundaries for linear or circular classifiers
- when there is to little data
- when there is to much data
- when one needs to do data fusion

Ensembles are a good choice,

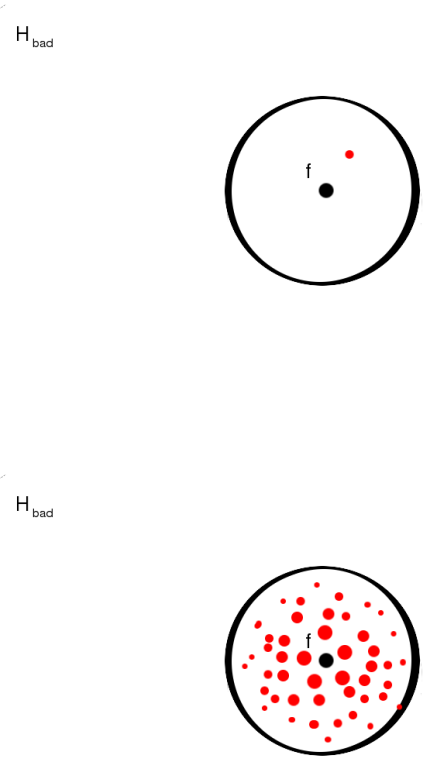
- with respect to computational learning theory
- for statistical reasons (error reduction through generalisation)
- if there are to complex boundaries for linear or circular classifiers
- when there is to little data
- when there is to much data
- when one needs to do data fusion

Computational learning theory



The upper figure shows the typical result of a successful single classifier training. The red dot marks the position of the learned hypothesis, the black dot ,in the center of the e-ball,is the position of the target function. The lower drawing shows the same space for a weighted ensemble. The size of the dots shows their weight in the ensemble. One can imagine that the decisions of the weighted ensemble are centred on f.

Computational learning theory



The upper figure shows the typical result of a successful single classifier training. The red dot marks the position of the learned hypothesis, the black dot ,in the center of the e-ball,is the position of the target function. The lower drawing shows the same space for a weighted ensemble. The size of the dots shows their weight in the ensemble. One can imagine that the decisions of the weighted ensemble are centred on f.

Generalisation

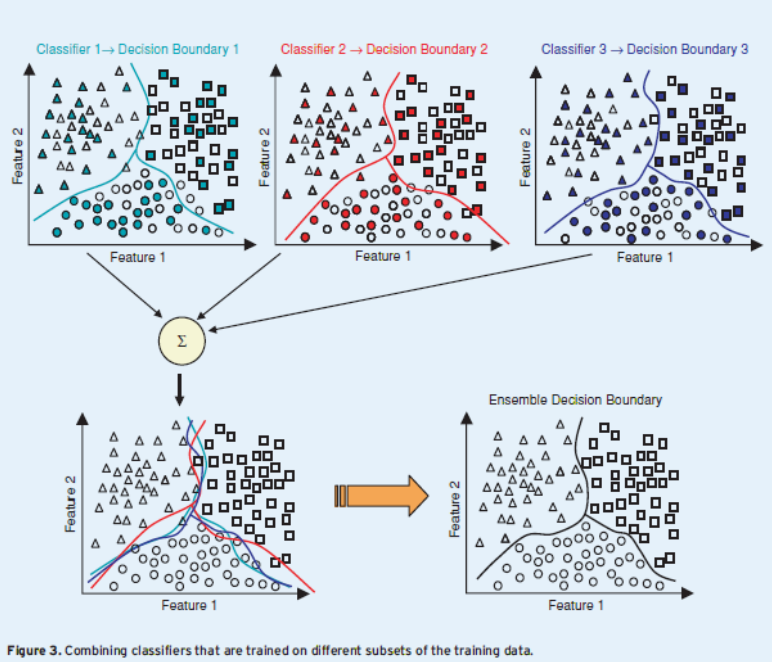


Figure 3. Combining classifiers that are trained on different subsets of the training data.

The figure shows how three classifiers trained with subsets of the training set, get an optimal decision boundary as an ensemble. One can imagine that the boundary gets better and maximises the distance to each class, as the number of classifiers increases.

Abbildung: Image stolen from [4]

Statistical reasons

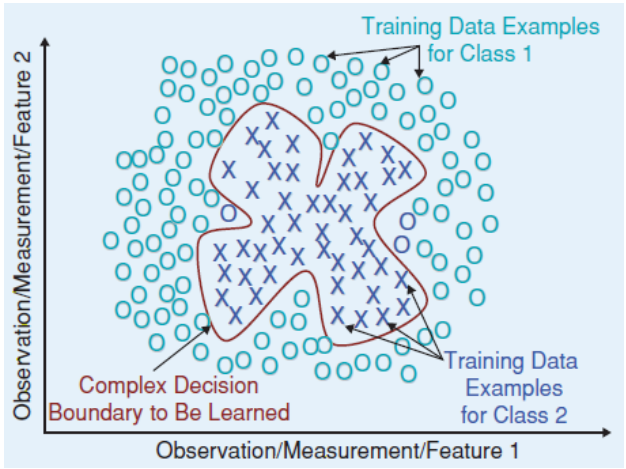
Consider three trees with a classification error of 0.2. Than the chance for a right classification is

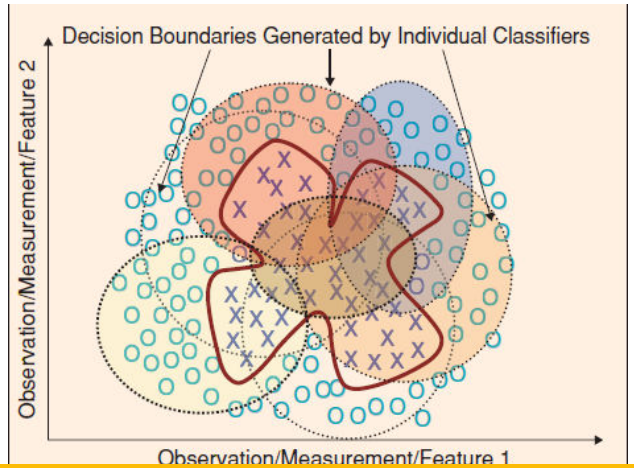
$$0.8^3+0.8\cdot0.8\cdot0.2+0.8\cdot0.2\cdot0.8+0.2\cdot0.8\cdot0.8 = 0.512+0.384 = 0.896 \quad (1)$$

It can be shown [4] that the number of wrong classifications approaches zero , as the number of classifiers increases, given the classifiers have an error rate below 0.5, for the chance of a right decision is:

$$\sum_{k=\lfloor \frac{T}{2} \rfloor+1}^{T} p^k \cdot (1-p)^{T-k} \binom{T}{k} \quad (2)$$

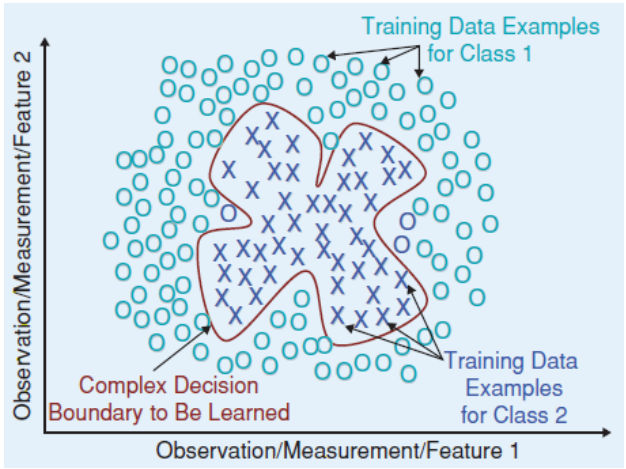
Complex boundaries

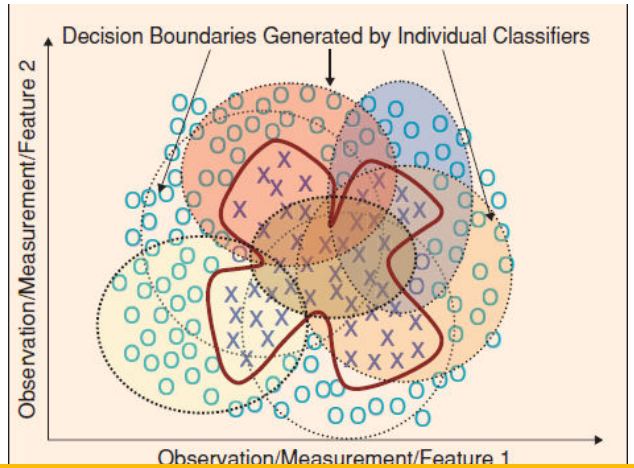




The figures to the left show that an ensemble can follow boundaries a linear or circular classifier cannot. The upper figure shows the class boundary in red. The lower figure shows how an ensemble of classifiers can cover the decision space and follow the class boundary using a majority vote. The Images are stolen from [4].

Complex boundaries





The figures to the left show that an ensemble can follow boundaries a linear or circular classifier cannot. The upper figure shows the class boundary in red. The lower figure shows how an ensemble of classifiers can cover the decision space and follow the class boundary using a majority vote. The Images are stolen from [4].

Data volume

The amount and structure of available training and test data is a key point in machine learning. Some of the problems ensembles can handle are:

- Data volumes might be to big to train on classifier on them
- Data volumes might be to complex to train on classifier on them
- Data might contain a black cat feature, that will have less impact if classifiers use subsets of features
- Not enough data might result in over fitted classifiers, re sampling will counter this effect in an ensemble

Data volume

The amount and structure of available training and test data is a key point in machine learning. Some of the problems ensembles can handle are:

- Data volumes might be to big to train on classifier on them
- Data volumes might be to complex to train on classifier on them
- Data might contain a black cat feature, that will have less impact if classifiers use subsets of features
- Not enough data might result in over fitted classifiers, re sampling will counter this effect in an ensemble

Data volume

The amount and structure of available training and test data is a key point in machine learning. Some of the problems ensembles can handle are:

- Data volumes might be to big to train on classifier on them
- Data volumes might be to complex to train on classifier on them
- Data might contain a black cat feature, that will have less impact if classifiers use subsets of features
- Not enough data might result in over fitted classifiers, re sampling will counter this effect in an ensemble

Data volume

The amount and structure of available training and test data is a key point in machine learning. Some of the problems ensembles can handle are:

- Data volumes might be to big to train on classifier on them
- Data volumes might be to complex to train on classifier on them
- Data might contain a black cat feature, that will have less impact if classifiers use subsets of features
- Not enough data might result in over fitted classifiers, re sampling will counter this effect in an ensemble

Data volume

The amount and structure of available training and test data is a key point in machine learning. Some of the problems ensembles can handle are:

- Data volumes might be to big to train on classifier on them
- Data volumes might be to complex to train on classifier on them
- Data might contain a black cat feature, that will have less impact if classifiers use subsets of features
- Not enough data might result in over fitted classifiers, re sampling will counter this effect in an ensemble

Data fusion

- Data of different sources might not “fit together” in one feature-set
- Different sources might have different Signal to noise ratio
- Different sources might have different classification noise
- Training classifiers for each source and integrate their decision into a single decision can solve this problem

Data fusion

- Data of different sources might not “fit together” in one feature-set
- Different sources might have different Signal to noise ratio
- Different sources might have different classification noise
- Training classifiers for each source and integrate their decision into a single decision can solve this problem

Data fusion

- Data of different sources might not "fit together" in one feature-set
- Different sources might have different Signal to noise ratio
- Different sources might have different classification noise
- Training classifiers for each source and integrate their decision into a single decision can solve this problem

Data fusion

- Data of different sources might not “fit together” in one feature-set
- Different sources might have different Signal to noise ratio
- Different sources might have different classification noise
- Training classifiers for each source and integrate their decision into a single decision can solve this problem

Ensemble decision making

One Task in the design of an ensemble is to determine how the decision of the ensemble is made from the decisions of the individual classifiers. The major method used are:

- Majority vote
- Weighted majority vote
- Behaviour Knowledge Space
- Borda count

Ensemble decision making

One Task in the design of an ensemble is to determine how the decision of the ensemble is made from the decisions of the individual classifiers. The major method used are:

- Majority vote
- Weighted majority vote
- Behaviour Knowledge Space
- Borda count

Ensemble decision making

One Task in the design of an ensemble is to determine how the decision of the ensemble is made from the decisions of the individual classifiers. The major method used are:

- Majority vote
- Weighted majority vote
- Behaviour Knowledge Space
- Borda count

Ensemble decision making

One Task in the design of an ensemble is to determine how the decision of the ensemble is made from the decisions of the individual classifiers. The major method used are:

- Majority vote
- Weighted majority vote
- Behaviour Knowledge Space
- Borda count

Ensemble decision making

One Task in the design of an ensemble is to determine how the decision of the ensemble is made from the decisions of the individual classifiers. The major method used are:

- Majority vote
- Weighted majority vote
- Behaviour Knowledge Space
- Borda count

Majority vote decisions

In majority vote decisions each classifiers d_i votes for one class c , and the class with the most votes wins. Mathematical this means the vote for one class $c \in C$ is

$$v_c = \sum_{i=0}^{n} d_i \tag{3}$$

and the decision is the class with the most votes.

Weighted majority vote decisions

In weighted majority voting each classifier d_i is assigned a weight w_i so that the vote for one class $c \in C$ is now

$$v_c = \sum_{i=0}^{n} d_i w_i \tag{4}$$

Again the class with the most votes wins.

Behaviour knowledge space decisions

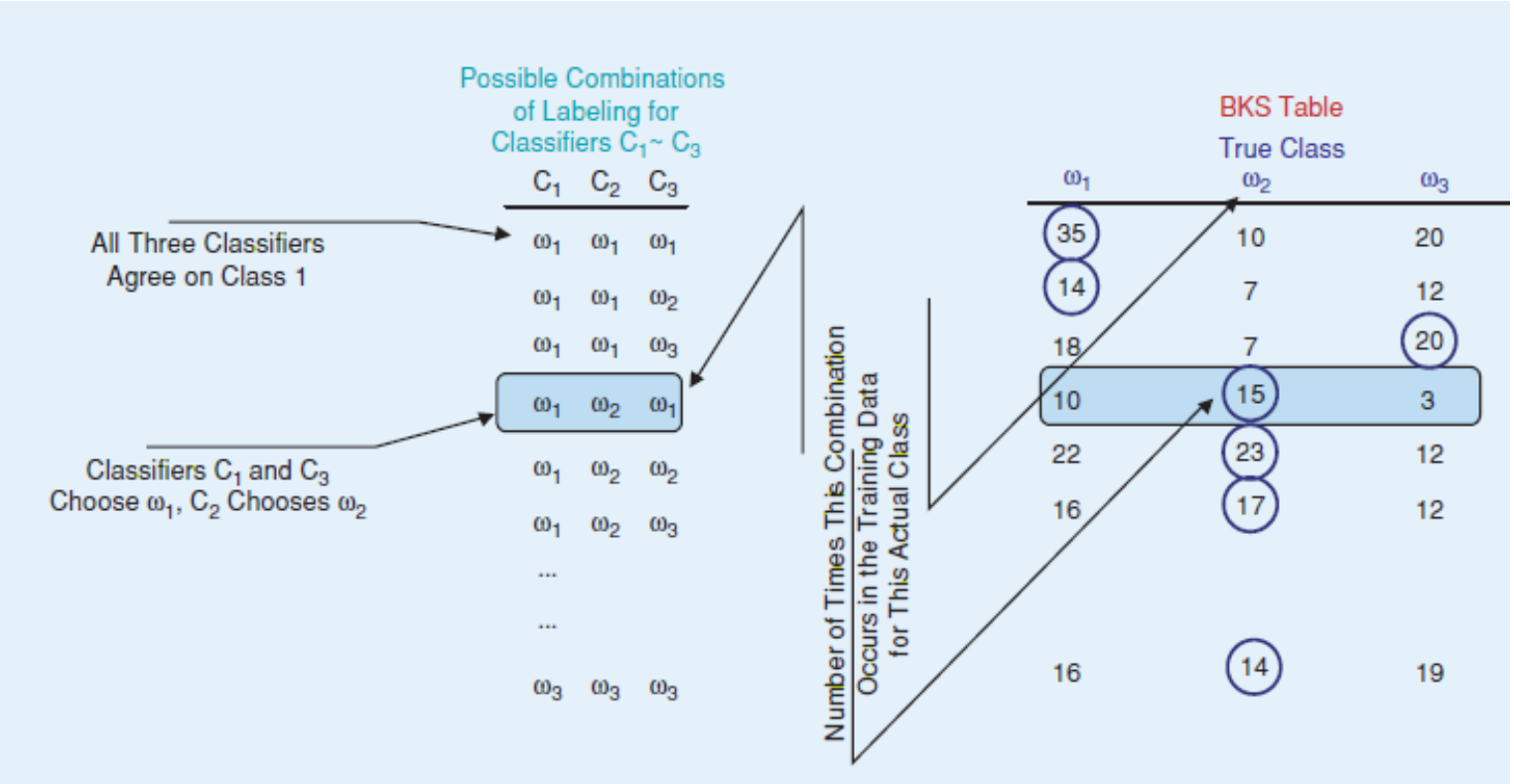


Abbildung: Image stolen from [4]

Borda count decisions

- In Borda Count each classifiers assigns a rank to the sample.
- In this way the sample is classified by an order of classes. The most likely class is the first in order.
- Borda count needs classifiers that allow a ranking of probable classes

Borda count decisions

- In Borda Count each classifiers assigns a rank to the sample.
- In this way the sample is classified by an order of classes. The most likely class is the first in order.
- Borda count needs classifiers that allow a ranking of probable classes

Borda count decisions

- In Borda Count each classifiers assigns a rank to the sample.
- In this way the sample is classified by an order of classes. The most likely class is the first in order.
- Borda count needs classifiers that allow a ranking of probable classes

comparison of voting methods

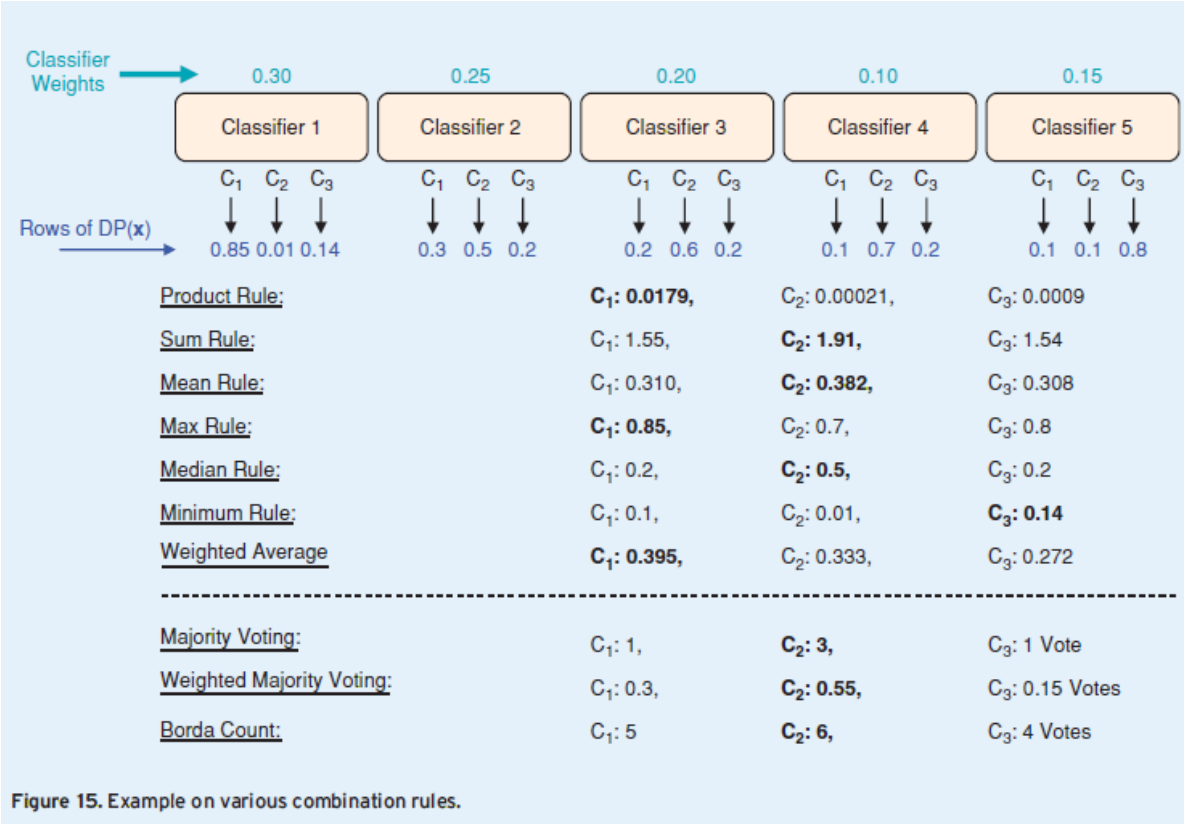


Figure 15. Example on various combination rules.

Abbildung: Image stolen from [4]

Popular training algorithms for ensembles

There are two algorithms for ensemble learning that are very popular. Boosting and bagging. There are other methods of which randomisation is the one compared to boosting and bagging in this talk.

Popular training algorithms for ensembles

There are two algorithms for ensemble learning that are very popular. Boosting and bagging. There are other methods of which randomisation is the one compared to boosting and bagging in this talk.

Popular training algorithms for ensembles

There are two algorithms for ensemble learning that are very popular. Boosting and bagging. There are other methods of which randomisation is the one compared to boosting and bagging in this talk.

Popular training algorithms for ensembles

There are two algorithms for ensemble learning that are very popular. Boosting and bagging. There are other methods of which randomisation is the one compared to boosting and bagging in this talk.

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Boosting

- Uses weighted training sets [6]
- Basic algorithm [6]
  1. Start with an equal weighted training set
  2. Train a classifier on the weighted set
  3. increase the weight of the samples the classifier cannot handle
  4. Do 2 and 3 until the weighted majority voting of the set is not getting more accurate
- Decision is based on weighted majority vote[6]
- Vulnerable to classification noise.[2]

Bootstrap Aggregating - Bagging

- Uses random subsets of training data drawn from the training data with replacement [4].
- Each subsets is used to train one classifier [4]
- Uses same classifier type [4]
- Decision is based on majority vote [4]
- Robust against classification noise[2]

Bootstrap Aggregating - Bagging

- Uses random subsets of training data drawn from the training data with replacement [4].
- Each subsets is used to train one classifier [4]
- Uses same classifier type [4]
- Decision is based on majority vote [4]
- Robust against classification noise[2]

Bootstrap Aggregating - Bagging

- Uses random subsets of training data drawn from the training data with replacement [4].
- Each subsets is used to train one classifier [4]
- Uses same classifier type [4]
- Decision is based on majority vote [4]
- Robust against classification noise[2]

Bootstrap Aggregating - Bagging

- Uses random subsets of training data drawn from the training data with replacement [4].
- Each subsets is used to train one classifier [4]
- Uses same classifier type [4]
- Decision is based on majority vote [4]
- Robust against classification noise[2]

Bootstrap Aggregating - Bagging

- Uses random subsets of training data drawn from the training data with replacement [4].
- Each subsets is used to train one classifier [4]
- Uses same classifier type [4]
- Decision is based on majority vote [4]
- Robust against classification noise[2]

Randomisation

Randomisation is less popular than bagging or boosting but is an interesting approach. Its included for completeness and for comparison to the other ensemble learning algorithms. Randomisation works by randomising the internal training parameters like split points of continuous features, to generate diverse classifiers.

Comparison of performance

The comparison of the performance of boosting, bagging and randomisation shows that,

- Boosting is superior to the other if there is no classification noise [4] [2]
- Bagging is superior to boosting if there is classification noise[2]
- Randomisation performs better than single classifiers but not as good as bagging or boosting[2]

In which areas ensembles are used

Areas in which ensembles are used include [4]

- Incremental learning
- Data fusion
- Feature Selection
- Confidence Estimation

In which areas ensembles are used

Areas in which ensembles are used include [4]

- Incremental learning
- Data fusion
- Feature Selection
- Confidence Estimation

In which areas ensembles are used

Areas in which ensembles are used include [4]

- Incremental learning
- Data fusion
- Feature Selection
- Confidence Estimation

In which areas ensembles are used

Areas in which ensembles are used include [4]

- Incremental learning
- Data fusion
- Feature Selection
- Confidence Estimation

Incremental Learning

- Often data is not available as a complete set but arrives in smaller packets.
- Retraining a classifier on the new data causes a loss of knowledge, and, if the new data is corrupt, wrong decisions
- It is easy to train classifiers for each packet and integrate it into an existing ensemble by weighting it according to its performance on the whole training set

Incremental Learning

- Often data is not available as a complete set but arrives in smaller packets.
- Retraining a classifier on the new data causes a loss of knowledge, and, if the new data is corrupt, wrong decisions
- It is easy to train classifiers for each packet and integrate it into an existing ensemble by weighting it according to its performance on the whole training set

Incremental Learning

- Often data is not available as a complete set but arrives in smaller packets.
- Retraining a classifier on the new data causes a loss of knowledge, and, if the new data is corrupt, wrong decisions
- It is easy to train classifiers for each packet and integrate it into an existing ensemble by weighting it according to its performance on the whole training set

Data fusion

- As mentioned earlier it is a good idea using ensemble learning when one needs to do data fusion, so it is also a typical application
- The next slide shows a video of an ensemble in action

Data fusion

-ss 2280 -endpos 14 -fs Videos/WinningTheDARPAGrand.avi

Feature selection

- Using different subsets of available features to train different classifiers will show which features are really valuable to the classification task.
- Doing so one might find out features that appear to be valuable to classification, but are independent of the classes

Feature selection

- Using different subsets of available features to train different classifiers will show which features are really valuable to the classification task.
- Doing so one might find out features that appear to be valuable to classification, but are independent of the classes

Confidence estimation

- Often it is important not to know the class of a sample, but also the probability that the classification is right.
- Ensembles provide a natural way to solve this problem
- The aggreement of the classifiers in the ensembles can be understand as the confidence in its decision
- Polikar writes [4] that Muhlbaier et al. show that the continuos valued outputs of ensemble classifiers can be used as an estimate of the posterior probability

Confidence estimation

- Often it is important not to know the class of a sample, but also the probability that the classification is right.
- Ensembles provide a natural way to solve this problem
- The aggreement of the classifiers in the ensembles can be understand as the confidence in its decision
- Polikar writes [4] that Muhlbaier et al. show that the continuos valued outputs of ensemble classifiers can be used as an estimate of the posterior probability

Confidence estimation

- Often it is important not to know the class of a sample, but also the probability that the classification is right.
- Ensembles provide a natural way to solve this problem
- The aggreement of the classifiers in the ensembles can be understand as the confidence in its decision
- Polikar writes [4] that Muhlbaier et al. show that the continuos valued outputs of ensemble classifiers can be used as an estimate of the posterior probability

Confidence estimation

- Often it is important not to know the class of a sample, but also the probability that the classification is right.
- Ensembles provide a natural way to solve this problem
- The aggreement of the classifiers in the ensembles can be understand as the confidence in its decision
- Polikar writes [4] that Muhlbaier et al. show that the continuos valued outputs of ensemble classifiers can be used as an estimate of the posterior probability

Questions

- Is a neural network an ensemble?
- Compare the voting/election system of the USA to that of the Bundesrepublik from a mathematical point of view


David Maxwell Chickering, Christopher Meek, and Robert Rounthwaite.
Efficient determination of dynamic split points in a decision tree.
In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 91–98, Washington, DC, USA, 2001. IEEE Computer Society.


Thomas G. Dietterich.
An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization.
Machine Learning, 40(2):139–157, 2000.


James Dougherty, Ron Kohavi, and Mehran Sahami.

Supervised and unsupervised discretization of continuous features.
In *ML-95*, 1995.


Robi Polikar.
Ensemble based systems in decision making.
IEEE Circuits and Systems Magazine, 2006.


David Poole, Alan Mackworth, and Randy Goebel.
Computational intelligence: a logical approach.
Oxford University Press, Oxford, UK, 1997.


Stuart Russell and Peter Norvig.
Artificial Intelligence: A Modern Approach.
Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.


Paul A. Viola and Michael J. Jones.

Robust real-time face detection.
International Journal of Computer Vision, 57(2):137–154, 2004.