Ensemble Methods for Noise Elimination in Classification Problems

Sofie Verbaeten *Anneleen Van Assche*

Report CW 358, June 2003

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Ensemble Methods for Noise Elimination in Classification Problems

Sofie Verbaeten[*] *Anneleen Van Assche*

Report CW 358, June 2003

Department of Computer Science, K.U.Leuven

Abstract

Ensemble methods combine a set of classifiers to construct a new classifier that is (often) more accurate than any of its component classifiers. In this paper, we use ensemble methods to identify noisy training examples. More precisely, we consider the problem of mislabeled training examples in classification tasks, and address this problem by pre-processing the training set, i.e. by identifying and removing outliers from the training set. We study a number of filter techniques that are based on well-known ensemble methods like cross-validated committees, bagging and boosting. We evaluate these techniques in an Inductive Logic Programming setting and use a first order decision tree algorithm to construct the ensembles.

Keywords : Machine Learning, Data pre-processing, Classification, Ensemble Methods.
CR Subject Classification : I.2.6.

[*]Sofie Verbaeten is a Postdoctoral Fellow of the Fund for Scientific Research - Flanders (Belgium)(F.W.O. - Vlaanderen).

1 Introduction

In many applications of machine learning the data to learn from is imperfect. Different kinds of imperfect information exist, and several classifications are given in the literature (see e.g. [3], [11], [14]). In [11] the following types of imperfect data for Inductive Logic Programming tasks are discussed: 1) noise, that is, random errors in training examples and background knowledge, 2) too sparse training examples, from which it is difficult to reliably detect correlations, 3) inappropriate or insufficient background knowledge, and 4) missing argument values in the training examples. As pointed out in [3], one can also have, in contrast to noise or random errors, systematic errors in the data (caused e.g. by incorrect calibration of an item of measuring equipment so that it is reading consistently low). In this paper, we consider the problem of noise or random errors in training examples for classification problems.

One of the problems created by learning from noisy data is overfitting, that is, the induction of an overly specific hypothesis which fits the (noisy) training data well but performs poor on the entire distribution of examples. Classical noise-handling mechanisms are based on appropriate search heuristics and stopping criteria used in the hypothesis construction, or on some form of post-processing of hypotheses. These techniques modify the learning algorithm itself to make it more noise-tolerant. Another approach, which we explore in this paper, is to pre-process the input data before learning. This approach consists of filtering the training examples (hopefully removing the noisy examples), and applying a learning algorithm on the reduced training set. As pointed out in [8], this separation of noise detection and hypothesis formation has the advantage that noisy examples do not influence the hypothesis construction, making the induced hypothesis less complex and more accurate.

In classification problems, noise in the training examples can be caused by erroneous argument values and/or erroneous classifications. Quinlan [15] demonstrated that, for high levels of noise, removing noise from attribute information decreases the predictive accuracy of the resulting classifier if the same attribute noise is present when the classifier is subsequently used. This is not the case for noise in the classification of examples. In this paper, we consider the problem of random labelling errors in training examples for classification problems.

Many of the methods for filtering training data are in fact removing outliers from the training data. An outlier is a case that does not follow the same model as the rest of the data. As such, an outlier does not only include erroneous data but also surprising correct data.[1] For instance in [4], the basic idea is to use a set of classifiers (induced by a number of possibly different learning methods) formed from part of the training data to test whether instances in the remaining part of the training data are mislabeled. By taking a consensus or majority vote of these classifiers, it is decided whether or not to remove a particular instance. The noise detection algorithm of [8] is based on the observation that the elimination of noisy examples, in contrast to the elimination of examples for which the target theory is correct, reduces the CLCH value of the training set (CLCH stands for the Complexity of the Least Complex correct Hypothesis). The noise detection algorithm is called the Saturation filter since it employs the CLCH measure to test whether the training set is saturated, i.e. whether, given a selected hypothesis language, the data set contains a sufficient number of examples to induce a stable and reliable target theory. In [9] two combinations of the Saturation filter and the filters proposed in [4] are proposed: the combined Classification-Saturation filter and the Consensus Saturation filter. In [10] robust decision trees are presented. Robust decision trees take the idea of pruning one step further: training examples which are locally uninformative and harmful, that is, examples which are misclassified by the pruned tree, are also globally uninformative. Therefore, after pruning a decision tree, the training examples which are misclassified should be removed from the training set and the tree needs to be rebuilt using this reduced set. This process is repeated until no more training examples are removed. In [18], we presented filter techniques for Inductive Logic Programming (ILP) that are based on the idea of [4]. We also applied the robust decision tree technique of [10] to the ILP setting. We already obtained some good results with the filters proposed in [18].

In this paper, we further explore a number of other, new techniques. We propose filter techniques that are based on well-known ensemble methods [5], namely cross-validated committees, bagging and boosting. We present two approaches: (1) filtering based on (unweighted) voting of classifiers that are built on different subsets of the training set (obtained by either cross-validation or bagging), (2)

[1] One work addressing the problem of distinguishing noise from exceptions is [17], we will not cover this topic here.

filtering based on removing training examples that obtain high weights in the boosting process. We introduce these filter techniques in the next section, and evaluate them in an ILP setting in section 3. We conclude and discuss topics for future research in section 4. In the appendix, more detailed results of the experiments are included.

2 Filter Algorithms

The filters that we present in subsections 2.2 and 2.3 make use of a learning algorithm for classification (subsection 2.1).

2.1 Base Classification Algorithm

Our filters make use of a learning algorithm for classification. With L we denote this base classification algorithm.

In the experiments, we evaluate the different filters in an ILP setting, and use Tilde [1] as the base learning algorithm L. Tilde (Top-down Induction of Logical Decision Trees) is an ILP extension of the C4.5 decision tree algorithm [16]. Instead of using attribute-value tests in the nodes of the tree, logical queries are used. As in C4.5, Tilde builds the decision tree in a top-down way, starting with the empty tree and all training examples. In each node, Tilde generates all possible tests and computes a heuristic value, in our experiments information gain ratio, for each of these tests. The test which scores best is placed in the node. The examples in the current node are then sorted down the tree: the examples passing the test are propagated to the left, the other examples to the right. The procedure is repeated for the left and right subtree. A node is turned into a leaf when the examples it covers are of a single class. After a tree is constructed, a post-pruning algorithm is used. The post-pruning algorithm that we will use here is the C4.5 post-pruning method which is based on an estimate of the error on unseen cases.

Note that the logical queries in the nodes of a first order decision tree may contain logical variables. These variables may be shared among different nodes under the restriction that a variable that is introduced in a node (that is, it does not occur in higher nodes) must not occur in the right branch (the "no"-branch) of that node. The reason for this restriction follows from the semantics of a first order decision tree. A variable that is introduced in a node is existentially quantified within the conjunction of that node. When this conjunction fails (and we thus enter the right subtree) no further reference to that variable is needed.

Tilde will be used in all filters described below.

2.2 Voting Filters

Voting filters are (as many other filter methods) based on the idea of removing outliers from a training set: an instance is removed if it can not be classified correctly by all, or the majority of, the classifiers built on parts of the training set. A motivation for using ensembles for filtering is pointed out in [4]: when we assume that some instances in the data have been mislabeled and that the label errors are independent of the particular model being fit to the data, collecting information from different models will provide a better method for detecting mislabeled instances than collecting information from a single model. As noted in many articles (see e.g. [5]), constructing ensembles of classifiers by manipulating the training examples works especially well for unstable learning algorithms. Decision tree algorithms, like Tilde, are unstable. Therefore, we expect that ensembles of decision trees will act well as a filter for noisy data sets.

The general scheme of our voting filters is as follows:

1. L induces n classifiers on different subsets of the training set,
2. these n classifiers give labels to every example in the training set,
3. the filter compares the original class of each example with the n labels it has, and decides whether or not to remove the example.

A variation of instances of this general scheme exists depending on the way these n classifiers are induced, the value of n and the decision procedure in step 3.

Concerning step 3, we consider two possibilities: (1) a *consensus* filter (C filter), where a training example is removed only if all the n labels it has differ from its class; (2) a *majority vote* filter (M filter), where a training example is removed if the majority of the labels it has differ from its class.

Concerning step 1, we present two approaches for building these n classifiers. In the first approach, the training set is partitioned in n subsets of (approximately) equal size. L is trained n times, each time leaving out one of the subsets from the training set. This results in n classifiers. Such a filter is called a *cross-validated committees* filter (X filter). In the second approach, n bootstrap replicates are taken from the training set, and L learns on these n sets. Such a filter is called a *bagging* filter (Ba filter). In [6], a motivation is found for using bagging as a filter. In that paper, it is experimentally shown that bagged C4.5 gains advantage over C4.5 when noise is added to the training sets. More precisely, it is observed that for most data sets, noise improves the diversity of bagging, which permits it to perform better.

We performed experiments with cross-validated committees - consensus (XC) filters, cross-validated committees - majority vote (XM) filters, bagging consensus (BaC) and bagging majority vote (BaM) filters. The parameter n was set to 5, 9, 15, and 25.

2.3 Boosting Filters

Boosting is a technique to construct ensembles (see e.g. [5]). Like bagging, it operates by taking a base learning algorithm and invoking it many times with different training sets. The boosting algorithm maintains a set of weights over the original training set and adjusts these weights after each classifier is learned by the base learning algorithm. The weights of examples that are misclassified by the induced classifier are increased and the weights of those that are correctly classified are decreased. After each round of the boosting process a new training set is constructed from the original data set. There are several ways to handle this. The simplest way is by having a base learning algorithm that can deal with weighted examples in the training set. Another way, that we will follow here, is by some sort of resampling. This means that examples from the original data set may occur multiple times in the new training set proportional to their weights. Hence examples with higher weights will occur more often in the training set. Consequently, the base learning algorithm is forced to concentrate more on the "difficult" examples or outliers. These outliers are exceptional examples or noisy examples.

Boosting is known to perform poorly with respect to noise. According to [6] a plausible explanation for the poor response of boosting to noise is that mislabeled training examples will tend to receive very high weights in the boosting process. Hence, after a few iterations, most of the training examples with high weights will be mislabeled examples. This gives a good motivation to use boosting as a noise filter. The idea is to use Adaboost [7] and to remove, after a number of rounds (n), the examples with the highest weights.

The general scheme for the boosting filter (Bo filter) is as follows (cfr. Adaboost [7]) (more details are given below):

1. The weights of the examples are initialised by $1/N$ (with N the size of the data set),
2. The training set is constructed: examples from the original data set are resampled and may occur multiple times in the new training set according to their weights.[2] We put however a maximum on the size of the training set, so examples with smallest weights may no longer occur in the new training set,
3. Tilde induces a classifier on the training set,
4. The weighted error e of the classifier is calculated as the sum of the weights of the misclassified examples. The weights of correctly classified examples are multiplied by β where $\beta = e/(1-e)$. All the weights are rescaled so they sum to 1,

[2]The first time the training set is constructed, it is equal to the original data set, since the weights of the examples are all the same.

5. Steps 2, 3 and 4 are repeated n times or until the weighted error e exceeds 0.5 (in which case the process is halted),

6. Finally, after n rounds, the examples with the highest weights are considered noise and are removed from the data set.

Within this general scheme several parameters need to be set. Resampling in step 2 means that examples with higher weights appear more often in the training set than those with smaller weights, as a result the training set will grow. We put a maximum on the size of the training set. The working of the filter might however be problematic when the maximum size of the training set is set too small. Hence this is an important parameter. For a data set with 392 examples (the Bongard data set, see subsection 3.1) we used 1000 and 3000 as a maximum size. We also performed experiments where the maximum size was set to the same size as the original data set, but then, for the majority of the cases, the error exceeds 0.5 already in the 4th or 5th round. For the other data sets (the trains data sets and the KRK data sets, see subsection 3.1) we used 3000 as the maximum size.

A second important parameter is the number of rounds of the boosting process (value of n in steps 5 and 6). Increasing the number of rounds increases the diversity of weights. We chose to test the boosting (Bo) filter with 3, 5, 10 and 17 rounds.

Finally, in step 6 the examples are sorted according to their weights and a certain percentage of examples with highest weights are removed. However the boosting filter has no idea what this percentage might be. We chose to give the exact percentage of noise as input to the filter.[3] This might be a disadvantage since in most cases the percentage of noise in the data is unknown, but on the other hand when it is indeed known, this filter might have an advantage over the other filters. Although the exact percentage of noise is given, the filter always filters more than that percentage. This is due to the fact that different examples can have the same weights, and if one example is removed, all other examples with the same weight are also removed.[4]

3 Experiments

We evaluate the different filters in an ILP setting. As opposed to propositional or attribute-value learning systems that use a single table to represent the data set, ILP systems use a first order representation. This makes ILP very suitable for dealing with complex data structures.

We first describe the data sets that we used in the experiments. Then we explain how the experiments were carried out, and finally we discuss the results. For the detailed results, we refer to the appendix.

3.1 Data Sets and Noise Introduction

We want to evaluate how well the different filter techniques perform on data sets with different amounts of classification noise. We therefore considered noise-free ILP data sets, and artificially introduced different levels of classification noise.

We considered the following noise-free data sets: an (artificial) Bongard data set [2] (392 examples), three (artificial) eastbound/westbound trains data sets [12] (200, 400, and 800 examples), and two (non-artificial) KRK data set for learning illegal positions in a chess endgame [11] (200 and 400 examples).[5] These are all 2-class problems.

We introduced different levels of classification noise in the data sets. A noise level of $x\%$ means that for a randomly chosen subset of $x\%$ of the training examples, the class-value of these examples was flipped.[6] We introduced noise levels of 0%, 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40%.

[3] In a later stage, we will try to estimate this.
[4] Another possibility is to remove less examples than is given as input; we chose the greedy approach here.
[5] See appendix for more details about the different data sets.
[6] Positive examples are made negative and vice versa.

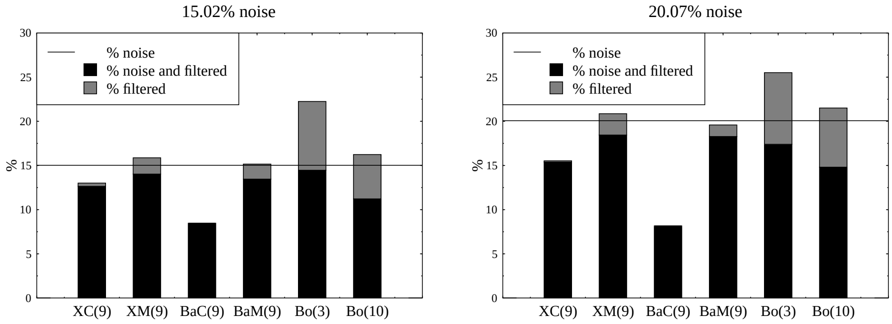


Figure 1: A comparison of the precision of some of the filters on the Bongard data set with 15% and 20% of noise. The different voting filters included in the figure each make use of 9 classifiers. The boosting filter with 3 and 10 rounds is shown.

3.2 Experimental Method

All the filters are based on Tilde. Tilde was also used to evaluate the performance of the different filtering techniques.

In order to obtain a more reliable estimate of the performance of the filters, all experiments were carried out in 10-fold cross-validation and the results were averaged over the 10 folds. For each of the 10 runs, the data set was divided in a training set (9 parts) and a test set (remaining 1 part). The training set was then corrupted by introducing classification errors using noise levels of 0%, 5%, 10%, 15%, 20%, 25%, 30%, 35% and 40%. Each of the above described filter techniques was then run on the (noisy[7]) training set. After filtering the training set, Tilde was used to learn a decision tree on the reduced training set. This decision tree was then validated on the (noise-free) test set. Results were obtained by taking the mean of the results of the 10 runs. For each of the 9 noise levels and each of the 10 runs, we also run Tilde directly on the (unfiltered) training set.

In the next subsections, we report results concerning filter precision, tree size and accuracy.

3.3 Filter Precision

We first evaluate how well the different filters perform. For each data set, each of the 9 noise levels and each of the filters, we report the following (see appendix): the percentage of examples that are removed, the percentage of examples that are removed and are actually noisy, the percentage of noisy examples in the filtered data sets, and an estimated probability of making an error of type 1 and of type 2. We also include, per data set, some bar-graphs that show, for each noise level and each filter, the percentage of filtered as well as the percentage of filtered and noisy examples.

A type 1 error (E_1) is made when a correct example is filtered. The probability of making a type 1 error is estimated as $P(E_1) = \frac{\sharp(\mathcal{F} \cap \mathcal{C})}{\sharp \mathcal{C}}$, where $\mathcal{F}$ is the set of filtered examples and $\mathcal{C}$ is the set of correct (non-noisy) examples. A type 2 error (E_2) is made when a noisy example is not removed from the training set. The probability of making a type 2 error is estimated as $P(E_2) = \frac{\sharp(\overline{\mathcal{F}} \cap \mathcal{N})}{\sharp \mathcal{N}}$, where $\overline{\mathcal{F}}$ is the complement of the set $\mathcal{F}$ (i.e. the examples which are not removed) and $\mathcal{N}$ is the set of noisy examples.

In Fig. 1, we show the precision of some of our filters on the Bongard data set with 15% and 20% of noise. In the next paragraphs we give for each filter only the general conclusions drawn from the results on the different data sets, as given in the appendix.

[7]For each noise level and each of the 10 training sets, the classification errors were introduced only once (in a random way), and the different filter techniques were run on the same noisy training sets.

3.3.1 Cross-Validated Committees - Consensus filters

As can be seen from Fig. 1 the XC(n) filter is a rather conservative filter: it does not remove so many examples (so there still remain quite some noisy examples in the filtered sets), but the probability of removing a correct example ($P(E_1)$) is rather low.

The influence of the parameter n in the XC(n) filter is as follows: the higher n,

- the higher the percentage of filtered examples,
- the higher the percentage of filtered and noisy examples,
- the lower the percentage of noisy examples in the filtered training set,
- the higher the probability of making an error of type 1,
- the lower the probability of making an error of type 2.

This is especially the case when there is not too much noise (up to 20%).

The fact that the XC(n) filter removes more examples as n increases, can be explained as follows. When n increases, the training sets on which the n classifiers are built become more similar, and as a consequence also the classifiers are less diverse, hence the classifiers will obtain a consensus in more cases.

3.3.2 Cross-Validated Committees - Majority Vote filters

The XM(n) filter is more aggressive than the XC(n) filter[8]: XM(n) removes more examples than XC(n); XM(n) removes more noisy examples, but also more non-noisy examples than XC(n). The precision of the XM(n) filter is quite good.

The influence of the parameter n in the XM(n) filter is not that big as in the XC(n) filter.

3.3.3 Bagging - Consensus filters

The BaC(n) filter is a very conservative filter: it removes few examples (so there remain still many noisy examples in the filtered sets), but the probability of removing a correct example ($P(E_1)$) is very low. One might choose to use a BaC(n) filter when data is sparse.

The influence of the parameter n in the BaC(n) filter is as follows. The higher n,

- the lower the percentage of filtered examples,
- the lower the percentage of filtered and noisy examples,
- the higher the percentage of noisy examples in the filtered training set,
- the lower the probability of making an error of type 1,
- the higher the probability of making an error of type 2.

Note that this behaviour is exactly the opposite of the behaviour of the XC(n) filters. We might explain this as follows. As we noted before, when n increases, the classifiers that are built using an n-fold cross-validation become more similar, so they will obtain a consensus in more cases. With Bagging, very diverse classifiers are built (especially when there is a lot of noise, see [6]). When the number (n) of diverse classifiers increases, it becomes more difficult to obtain a consensus, hence less examples will be removed.

[8]This is trivial, since we take a majority vote instead of a consensus vote to remove examples.

3.3.4 Bagging - Majority Vote filters

The precision of the BaM(n) filters is quite good.

It seems that the higher the parameter n in the BaM(n) filter,

- the lower the percentage of filtered examples,
- the higher the percentage of filtered and noisy examples,
- the lower the percentage of noisy examples in the filtered training set,
- the lower the probability of making an error of type 1,
- the lower the probability of making an error of type 2.

So it seems that the BaM(n) filter improves its precision when n, the number of classifiers, is increased. Although this observation is violated in some cases (mostly in data sets with a small number of examples), the general tendency is as such in all the data sets.

3.3.5 Boosting Filter

The general tendency is as follows: the more rounds in the boosting process

- the lower the percentage filtered examples,
- the lower the percentage of filtered and noisy examples,
- the higher the percentage of noisy examples in the filtered training set,
- the lower the probability of making an error of type 1,
- the higher the probability of making an error of type 2,

Despite the fact that the exact percentage of noisy examples is given as input to the Bo(n) filters, these filters will remove more examples. This is because there might be examples that have the same weight. Since this is especially the case in the first rounds of the boosting process, we observe that, the higher n, the lower the number of filtered examples. At the same time we observe that, the higher n, the less noisy examples are filtered.

For the Bongard data set, both the probability of making an error of type 1 and 2 seem to be higher when the training set has a maximum size of 3000 examples than in the case of a maximum size of 1000 examples.

As can be seen from the data in the tables in the appendix, for a noise level of 15% and more, the precision of Bo(3) with maximum size 3000 is the same as the precision of Bo(3) with maximum size 1000. In the context of a higher percentage of noise there is less initial resampling. This causes the size of the training set never to exceed 1000, what makes the results the same. For more than 3 rounds this doesn't hold anymore.

3.3.6 Influence of the size of the training set on the filter precision

Finally, by looking at the results on the three trains data sets, we observe that the more training examples we have at our disposal, the more precise the filters are. See for example for 15% of noise the results of the voting filters on the three trains data sets in Fig. 2.

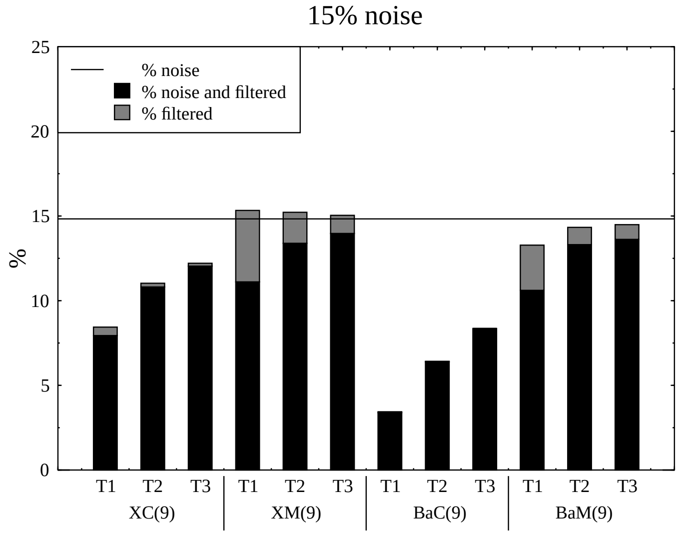


Figure 2: Filter precision of XC(9), XM(9), BaC(9) and BaM(9) on the three trains data sets with 15% noise. T1 has 200 examples, T2 400 and T3 800 examples. It is clear that the precision of the filters increases with the size of the data set.

3.4 Tree Size

A decision tree that is built from a noisy training set might be overly complex due to overfitting of this set. Therefore, it can be expected that the sizes of the trees induced from a filtered (and hopefully non-noisy) training set are smaller than the sizes of the trees induced from the non-filtered, noisy training set.

In Fig. 3, we show the number of nodes in the decision trees induced from (un)filtered Bongard training sets. More precisely, we report the results for the XM(9), BaM(9) and Bo(10) filters.

For noise levels up to 15% it is indeed the case that the sizes of the trees induced from a filtered training set are smaller than the sizes of the trees induced from the unfiltered set. For higher noise levels however there is, for many of the cases, no decrease in tree size if a filtered training set is used. One plausible explanation is that, for high noise levels, the filters still leave some amount of noise in the training sets. Also, we should note that Tilde with pruning is used (both in the filter algorithms and for inducing decision trees from the (un)filtered training sets), so the effect of overfitting is already largely reduced.

In [13] it is empirically shown that for many data sets there is a nearly linear relationship between training set size and tree size, even after accuracy has ceased to increase. This suggests that all data reduction techniques will see some decrease in tree size simply because they are reducing the size of the training set. Therefore, each data reduction method should be compared with random data reduction. Only then one can have an idea of how much of the reduction in tree size is directly attributable to how a data reduction method selects instances to remove.

In [13] this analysis was done for Robust C4.5 [10] and it was shown that 41.67% of the decrease in tree size is attributable to reduction in training set size. The remainder is due to the removal of uninformative examples. We plan to make this analysis also for our filters.

3.5 Accuracy

Decision trees built on a non-noisy training set will (in general) be more accurate (on a separate test set) than trees induced from a noisy training set. We compare the accuracies of the trees induced from the filtered sets (on the non-noisy test sets) with the accuracies (also on the non-noisy test sets) of the

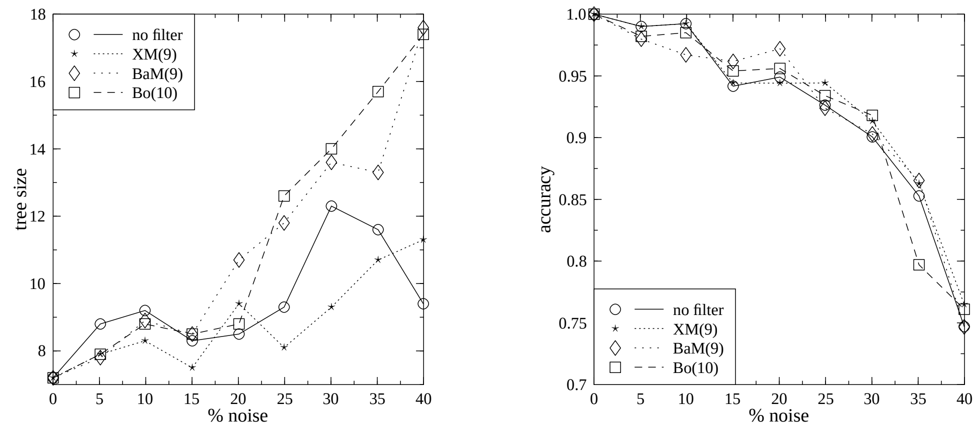


Figure 3: Results concerning tree size (left) and accuracy (right) on the Bongard data set.

trees induced from the unfiltered, noisy sets.

In Fig. 3, we show our results for the XM(9), BaM(9) and Bo(10) filters on the Bongard data set. For this data set and also for the trains data sets, we observe that for noise levels up to 10%, Tilde still performs well on an unfiltered training set. For higher noise levels, it seems better to first filter the training set. No one filter outperforms the other filters in this respect, although the Bo filters perform in general the worst. For the KRK data sets, it is better to also filter the training set for low noise levels.

3.5.1 Voting versus Filtering

A hypothesis of interest is whether a majority vote ensemble classifier can be used instead of filtering, or whether the best method is to first filter the training set and then use a majority vote ensemble classifier. This hypothesis was tested in [4]. There, two majority vote ensemble classifiers were formed: one from the filtered and one from the unfiltered data. The resulting classifiers were then used to classify the uncorrupted test data. The experiments in [4] show that the majority vote classifier performed better than the individual classifiers, but that it cannot replace filtering when data are noisy. It is concluded that the best approach is to combine filtering and voting.

We also tested this hypothesis in our setting. More precisely, we formed two cross-validated committees (Xcom), one from the XM(9)-filtered training sets, and one from the unfiltered training sets. Our Xcom(9) classifier is constructed by partitioning the training set in 9 subsets of (approximately) equal size. Tilde is trained 9 times, each time leaving out one of the subsets from the training set (as in the X(9) filter). The resulting 9 classifiers classify the examples in the test set, and a majority vote is taken to give the final classification to the test examples.

In Fig. 4 we compare the accuracies on the test set of Tilde / Xcom(9) run on a unfiltered / XM(9)-filtered Bongard training set. Up to a noise level of 15%, there is no significant difference in the accuracies of the classifiers. For high noise levels (25% and more), we see that it is better to either use the XM(9) filter or use a Xcom(9) classifier or use both. Further experiments are needed to see what the best method is.

We also tested whether bagging can be used instead of filtering (with a bagging filter). More precisely, we formed two bagging classifiers, one from the BaM(9)-filtered training sets, and one from the unfiltered training sets. Our bagging classifier Bagging(9) is constructed by taking 9 bootstrap replicates of the training set (as in the Ba(9) filter), and building 9 classifiers on these sets. These 9 classifiers classify the examples in the test set, and a majority vote is taken to give the final classification to the test

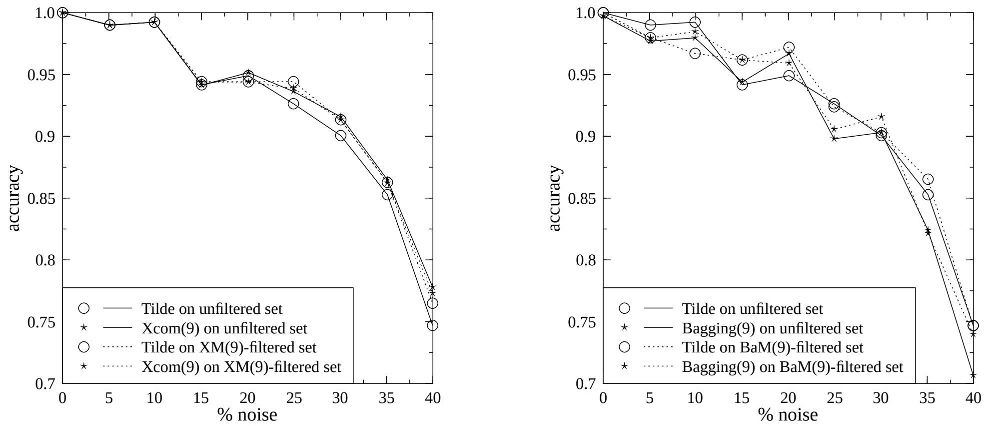


Figure 4: Results concerning filtering versus voting (left) and filtering versus bagging (right) on the Bongard data set.

examples.

In Fig. 4 we compare the accuracies on the test set of Tilde / Bagging(9) run on a unfiltered / XM(9)-filtered training set. For high noise levels (25% and more) bagging without filtering the training set first doesn't give good results. In general (see the appendix for the results on the other data sets), bagging after BaM(9)-filtering the training set gives the best results.

4 Conclusions and Future Work

We addressed the problem of training sets with mislabeled examples in classification tasks. We proposed a number of filter techniques, based on ensemble methods, for identifying and removing noisy examples. We experimentally evaluated these techniques on noise-free ILP data sets which we artificially corrupted with different levels of classification noise. We reported results concerning filter precision, tree size and accuracy.

Both the BaM and XM filters have a good precision. Surprisingly, the Bo filters did not perform so well. We plan to investigate in more detail how the boosting process can be used/modified to obtain a noise filter. We also plan to evaluate the proposed filters on more data sets, and test other voting schemes than majority and consensus.

When the data set is small and the cost of finding new training examples is high, one can choose to use a conservative filter, e.g. a BaC or XC filter. A better solution would be to detect and also correct labelling errors (and thus not removing any example). One way to do this is to present the suspicious data to a human expert and ask what to do with it. Another way is to automatically switch the class labels of the examples which are identified as noise. We will evaluate the performance of such an extension.

Acknowledgements

Sofie Verbaeten is a Postdoctoral Fellow of the Fund for Scientific Research - Flanders (Belgium) (F.W.O.- Vlaanderen). Anneleen Van Assche is supported by the GOA/2003/08(B0516) on Inductive Knowledge Bases.

References

[1] H. Blockeel and L. De Raedt. Top-down induction of first order logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, June 1998.

[2] M. Bongard. *Pattern Recognition.* Spartan Books, 1970.

[3] P. Brazdil and P. Clark. Learning from imperfect data. In P. Brazdil and K. Konolige, editors, *Machine Learning, Meta-reasoning and Logics*, pages 207–232. Kluwer Academic Press, 1990.

[4] C.E. Brodley and M.A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

[5] T.G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, First International Workshop*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2000.

[6] T.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.

[7] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[8] D. Gamberger, N. Lavrač, and S. Džeroski. Noise detection and elimination in data preprocessing: experiments in medical domains. *Applied Artificial Intelligence*, 14:205–223, 2000.

[9] Dragan Gamberger, Nada Lavrač, and Ciril Grošelj. Experiments with noise filtering in a medical domain. In *Proc. 16th International Conf. on Machine Learning*, pages 143–151. Morgan Kaufmann, San Francisco, CA, 1999.

[10] G.H. John. Robust decision trees: Removing outliers from databases. In U.M. Fayyad and R. Uthurusamy, editors, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 174–179. AAAI Press, 1995.

[11] N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, 1994.

[12] R.S. Michalski and J.B. Larson. Inductive inference of VL decision rules. Paper presented at Workshop in Pattern-Directed Inference Systems, Hawaii, 1977. SIGART Newsletter, ACM, 63, 38-44.

[13] Tim Oates and David Jensen. The effects of training set size on decision tree complexity. In *Proceedings of The Fourteenth International Conference on Machine Learning*, pages 254–262. Morgan Kaufmann, Nashville, TN, 1997.

[14] S. Parsons. Imperfect information in knowledge and databases: a survey of current approaches. *IEEE Transactions on Knowledge and Data Engineering*, 8(3):353–372, 1996.

[15] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

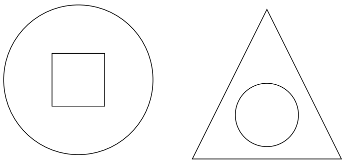
[16] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann series in machine learning. Morgan Kaufmann, 1993.

[17] A. Srinivasan, S. Muggleton, and M. Bain. Distinguishing exceptions from noise in non-monotonic learning. In *Proceedings of the 2nd International Workshop on Inductive Logic Programming*, 1992.

[18] S. Verbaeten. Identifying mislabeled training examples in ILP classification problems. In M. Wiering and W. de Back, editors, *Twelfth Dutch-Belgian Conference on Machine Learning*, pages 1–8, 2002.

A Bongard data set

Bongard data sets [2] consist of configurations of geometrical objects, more precisely triangles, squares and/or circles. The direction in which triangles point can be up or down. An object can be inside another object. Each example is classified as either positive or negative. A typical Bongard example is shown below.



```
circle(o1).        in(o2,o1).
square(o2).        in(o4,o3).
triangle(o3).      config(o3,up).
circle(o4).
```

negative.

A.1 Bongard, 392 examples

A.1.1 Filter Precision

Cross-Validated Committees - Consensus filters

XC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	4.88	4.65	0.48	0.002	0.089
9.92	8.79	8.79	1.24	0.000	0.114
15.02	13.18	12.58	2.80	0.007	0.162
20.07	15.70	15.36	5.55	0.004	0.235
24.94	17.20	16.35	10.34	0.011	0.344
30.05	17.24	16.84	15.82	0.006	0.440
35.09	15.84	14.74	24.08	0.017	0.580
39.97	14.06	12.50	31.90	0.026	0.687

XC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	4.82	4.59	0.54	0.002	0.100
9.92	9.27	9.10	0.90	0.002	0.083
15.02	13.01	12.64	2.73	0.004	0.158
20.07	15.53	15.39	5.49	0.002	0.233
24.94	17.49	16.75	9.82	0.010	0.328
30.05	19.90	19.36	13.25	0.008	0.356
35.09	17.69	16.44	22.54	0.019	0.532
39.97	12.95	11.51	32.56	0.024	0.712

XC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	4.88	4.65	0.48	0.002	0.089
9.92	9.21	9.18	0.80	0.000	0.074
15.02	13.46	12.90	2.45	0.007	0.142
20.07	16.89	16.67	4.06	0.003	0.170
24.94	18.62	17.52	9.02	0.015	0.298
30.05	19.64	19.08	13.53	0.008	0.365
35.09	16.55	15.28	23.64	0.020	0.565
39.97	12.78	11.28	32.75	0.025	0.718

XC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5.10	5.05	4.76	0.36	0.003	0.067
9.92	9.55	9.32	0.65	0.003	0.060
15.02	13.58	12.90	2.46	0.008	0.142
20.07	16.92	16.69	4.02	0.003	0.168
24.94	19.30	18.08	8.35	0.016	0.275
30.05	20.13	19.28	13.43	0.012	0.358
35.09	18.03	16.41	22.57	0.025	0.532
39.97	15.56	13.66	31.10	0.032	0.658

Cross-Validated Committees - Majority Vote filters

XM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.11	0.00	0.00	0.001	NA
5.10	5.50	4.85	0.27	0.007	0.050
9.92	10.74	9.75	0.19	0.011	0.017
15.02	15.73	13.80	1.45	0.023	0.081
20.07	20.32	18.82	1.56	0.019	0.062
24.94	25.25	22.19	3.66	0.041	0.110
30.05	29.65	26.59	4.90	0.044	0.115
35.09	33.33	26.10	13.49	0.111	0.256
39.97	35.54	26.25	21.18	0.155	0.343

XM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5.10	5.50	4.88	0.24	0.007	0.044
9.92	10.26	9.84	0.10	0.005	0.009
15.02	15.87	14.03	1.19	0.022	0.066
20.07	20.86	18.45	2.04	0.030	0.081
24.94	25.23	22.68	3.02	0.034	0.091
30.05	29.25	26.19	5.43	0.044	0.128
35.09	32.54	27.04	11.84	0.085	0.230
39.97	35.63	26.14	21.49	0.158	0.346

XM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5.10	5.56	4.88	0.24	0.007	0.044
9.92	10.26	9.81	0.13	0.005	0.011
15.02	15.85	13.95	1.28	0.022	0.072
20.07	21.09	18.28	2.26	0.035	0.089
24.94	25.06	22.53	3.20	0.034	0.097
30.05	29.11	25.51	6.35	0.051	0.151
35.09	32.14	25.99	13.29	0.095	0.260
39.97	36.25	25.65	22.49	0.177	0.358

XM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5.10	5.56	4.88	0.24	0.007	0.044
9.92	10.29	9.81	0.13	0.005	0.011
15.02	15.85	13.95	1.28	0.022	0.072
20.07	21.37	18.11	2.49	0.041	0.097
24.94	25.03	22.31	3.50	0.036	0.106
30.05	29.08	25.40	6.51	0.053	0.155
35.09	32.48	26.61	12.46	0.090	0.242
39.97	36.14	25.40	22.83	0.179	0.365

Bagging - Consensus filters

BaC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	4.05	4.05	1.09	0.000	0.206
9.92	7.99	7.96	2.12	0.000	0.197
15.02	10.32	10.18	5.38	0.002	0.323
20.07	11.31	11.31	9.85	0.000	0.437
24.94	12.30	11.96	14.76	0.005	0.520
30.05	10.57	10.35	21.97	0.003	0.656
35.09	9.35	8.79	28.95	0.009	0.750
39.97	7.54	6.63	36.01	0.015	0.834

BaC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	3.51	3.51	1.64	0.000	0.311
9.92	6.80	6.80	3.33	0.000	0.314
15.02	8.47	8.47	7.14	0.000	0.436
20.07	8.16	8.16	12.92	0.000	0.594
24.94	9.10	9.01	17.50	0.001	0.639
30.05	7.17	7.14	24.63	0.000	0.762
35.09	5.73	5.50	31.37	0.003	0.843
39.97	3.26	3.18	38.00	0.001	0.921

BaC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	3.26	3.26	1.90	0.000	0.361
9.92	6.24	6.24	3.92	0.000	0.371
15.02	7.88	7.88	7.74	0.000	0.475
20.07	6.69	6.66	14.34	0.000	0.668
24.94	6.89	6.89	19.36	0.000	0.724
30.05	6.41	6.35	25.29	0.001	0.789
35.09	3.26	3.18	32.97	0.001	0.909
39.97	1.98	1.98	38.74	0.000	0.950

BaC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5.10	3.12	3.12	2.04	0.000	0.389
9.92	5.53	5.53	4.64	0.000	0.443
15.02	6.58	6.55	9.05	0.000	0.564
20.07	5.41	5.41	15.47	0.000	0.730
24.94	5.02	5.02	20.96	0.000	0.799
30.05	5.27	5.27	26.12	0.000	0.825
35.09	1.76	1.73	33.95	0.000	0.951
39.97	1.64	1.62	38.99	0.000	0.960

Bagging - Majority Vote filters

BaM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.23	0.00	0.00	0.002	NA
5.10	6.18	4.79	0.33	0.015	0.061
9.92	10.35	9.41	0.57	0.010	0.051
15.02	15.05	14.00	1.20	0.012	0.068
20.07	20.86	17.40	3.39	0.043	0.133
24.94	24.32	20.44	5.96	0.052	0.181
30.05	28.26	22.85	9.99	0.077	0.240
35.09	31.43	23.44	16.90	0.123	0.332
39.97	34.35	22.90	25.97	0.191	0.427

BaM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5.10	5.61	4.82	0.30	0.008	0.056
9.92	10.83	9.64	0.32	0.013	0.029
15.02	15.16	13.46	1.83	0.020	0.104
20.07	19.59	18.28	2.22	0.016	0.089
24.94	23.55	20.69	5.55	0.038	0.170
30.05	27.32	22.90	9.83	0.063	0.238
35.09	31.55	25.79	13.55	0.089	0.265
39.97	32.82	23.47	24.50	0.156	0.413

BaM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.11	0.00	0.00	0.001	NA
5.10	5.56	4.88	0.24	0.007	0.044
9.92	10.69	9.67	0.29	0.011	0.026
15.02	15.31	13.83	1.41	0.017	0.079
20.07	20.24	17.91	2.71	0.029	0.107
24.94	23.95	21.57	4.43	0.032	0.135
30.05	27.66	24.77	7.25	0.041	0.175
35.09	30.36	24.06	15.79	0.097	0.314
39.97	32.74	24.24	23.32	0.142	0.394

BaM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.06	0.00	0.00	0.001	NA
5.10	5.36	4.82	0.30	0.006	0.056
9.92	10.37	9.75	0.19	0.007	0.017
15.02	15.25	13.77	1.48	0.017	0.083
20.07	19.73	18.45	2.00	0.016	0.081
24.94	23.72	21.63	4.32	0.028	0.133
30.05	26.76	23.61	8.73	0.045	0.214
35.09	30.02	25.31	13.93	0.072	0.279
39.97	32.54	23.05	25.11	0.158	0.423

Boosting Filter

Bo(3) maxsize 1000

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	6.80	4.68	0.46	0.022	0.083
9.92	16.10	9.18	0.82	0.077	0.074
15.02	22.25	14.46	0.73	0.092	0.038
20.07	25.51	17.40	3.50	0.102	0.132
24.94	30.10	23.07	2.63	0.094	0.075
30.05	35.74	26.90	4.93	0.126	0.105
35.09	49.77	27.44	15.42	0.344	0.218
39.97	52.38	26.14	28.19	0.437	0.346

Bo(5) maxsize 1000

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	6.32	4.39	0.76	0.020	0.139
9.92	12.13	9.07	0.97	0.034	0.086
15.02	17.15	12.75	2.75	0.052	0.151
20.07	23.35	16.73	4.36	0.083	0.166
24.94	27.24	19.50	7.45	0.103	0.218
30.05	33.19	24.26	8.67	0.128	0.192
35.09	42.21	26.33	15.22	0.245	0.250
39.97	44.87	26.12	25.15	0.312	0.347

Bo(10) maxsize 1000

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	6.12	4.19	0.97	0.020	0.178
9.92	10.86	8.33	1.78	0.028	0.160
15.02	16.24	11.22	4.53	0.059	0.253
20.07	21.51	14.81	6.68	0.084	0.262
24.94	26.19	17.80	9.69	0.112	0.286
30.05	31.41	20.58	13.80	0.155	0.315
35.09	36.20	22.62	19.56	0.209	0.355
39.97	41.00	23.96	27.14	0.284	0.401

Bo(17) maxsize 1000

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	5.98	3.94	1.24	0.022	0.228
9.92	10.71	7.81	2.36	0.032	0.213
15.02	16.00	10.99	4.80	0.059	0.268
20.07	20.98	14.13	7.49	0.086	0.295
24.94	26.11	17.23	10.43	0.118	0.309
30.05	31.18	19.70	15.03	0.164	0.344
35.09	36.48	21.37	21.61	0.233	0.391
39.97	40.57	22.71	29.04	0.298	0.432

Bo(3) maxsize 3000

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	7.51	4.08	1.09	0.036	0.200
9.92	16.89	9.81	0.13	0.079	0.011
15.02	22.25	14.46	0.73	0.092	0.038
20.07	25.51	17.40	3.50	0.102	0.132
24.94	30.10	23.07	2.63	0.094	0.075
30.05	35.74	26.90	4.93	0.126	0.105
35.09	49.77	27.44	15.42	0.344	0.218
39.97	52.38	26.14	28.19	0.437	0.346

Bo(5) maxsize 3000

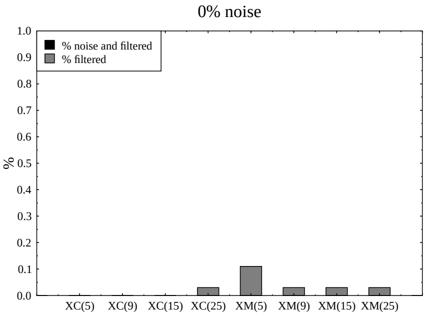
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	6.46	4.00	1.18	0.026	0.217
9.92	13.35	8.59	1.55	0.053	0.134
15.02	18.23	12.22	3.44	0.071	0.187
20.07	24.81	15.98	5.50	0.110	0.203
24.94	27.35	19.22	7.87	0.108	0.230
30.05	32.29	23.61	9.48	0.124	0.214
35.09	42.26	26.30	15.28	0.246	0.250
39.97	45.15	26.22	25.07	0.315	0.344

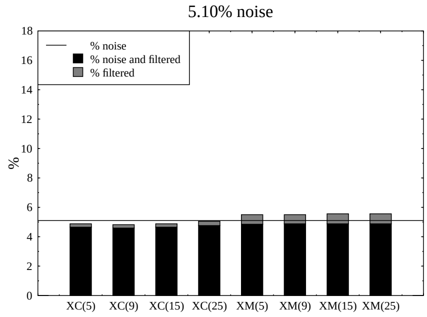
Bo(10) maxsize 3000

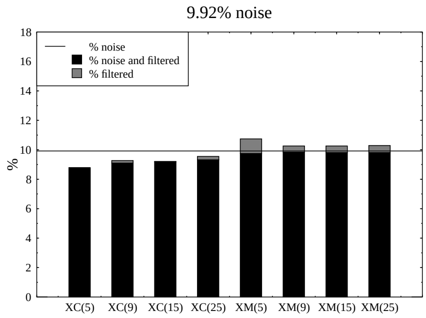
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	6.07	3.86	1.33	0.023	0.244
9.92	10.86	7.82	2.35	0.034	0.211
15.02	16.04	10.88	4.93	0.061	0.275
20.07	22.64	13.68	8.24	0.112	0.318
24.94	26.98	16.98	10.92	0.133	0.319
30.05	31.12	19.76	14.93	0.162	0.342
35.09	36.99	21.71	21.23	0.235	0.381
39.97	41.03	22.15	30.23	0.315	0.446

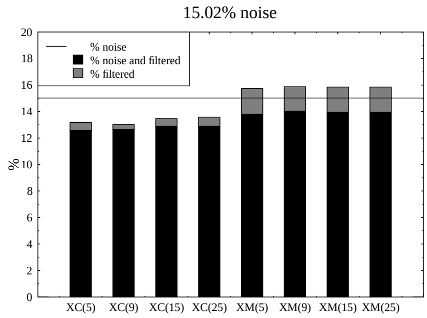
Bo(17) maxsize 3000

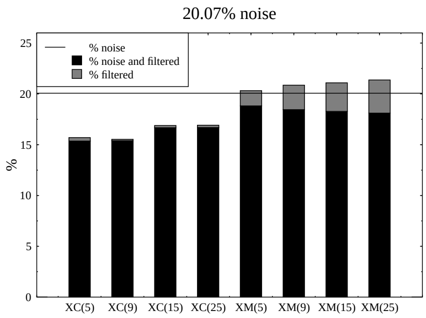
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5.10	5.89	3.78	1.41	0.022	0.259
9.92	10.88	7.43	2.80	0.038	0.251
15.02	15.79	10.71	5.12	0.060	0.287
20.07	20.98	13.68	8.07	0.091	0.318
24.94	26.84	16.72	11.23	0.135	0.330
30.05	31.86	18.82	16.47	0.186	0.374
35.09	36.31	21.00	22.13	0.236	0.402
39.97	41.18	21.62	31.21	0.326	0.459

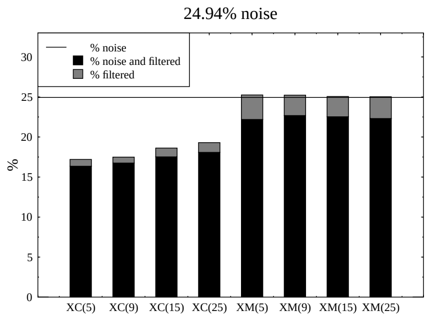

0% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

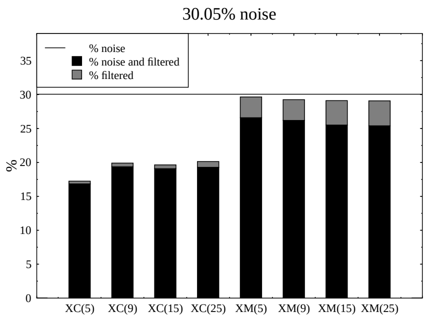

5.10% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

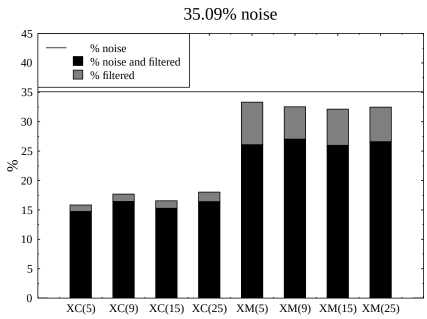

9.92% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

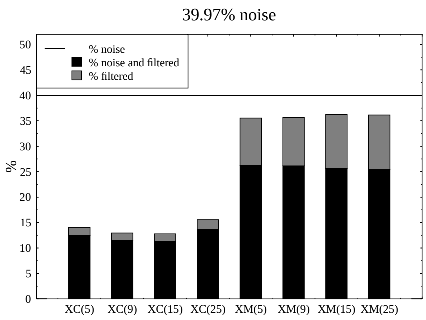

15.02% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)


20.07% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

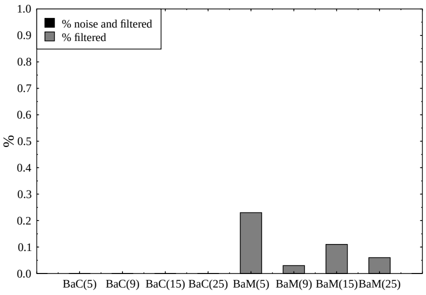

24.94% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)


30.05% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

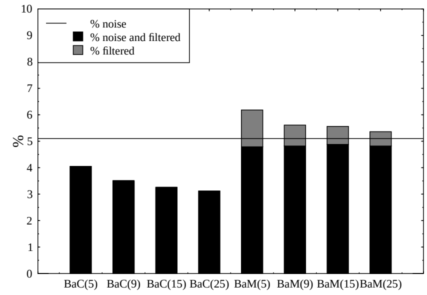

35.09% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)


39.97% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

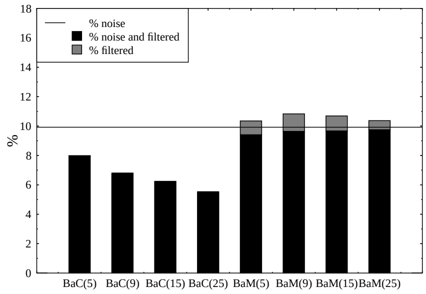
0% noise



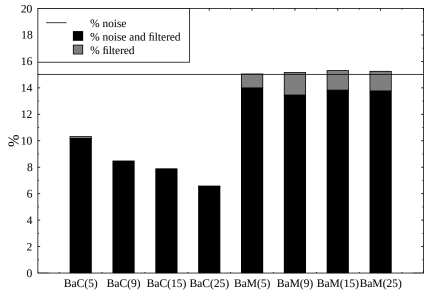
5.10% noise



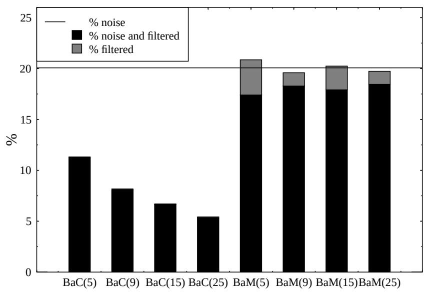
9.92% noise



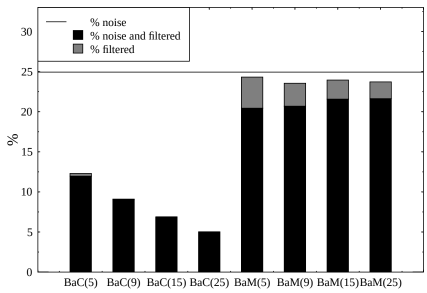
15.02% noise



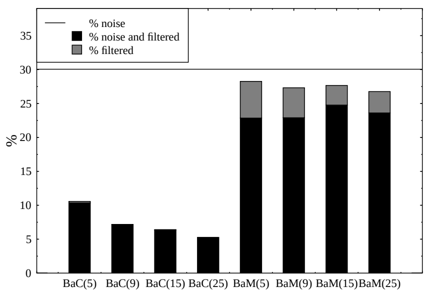
20.07% noise



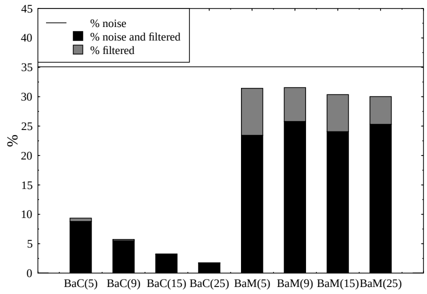
24.94% noise



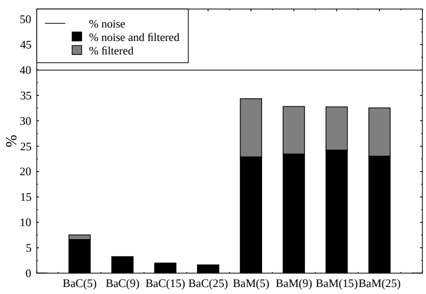
30.05% noise

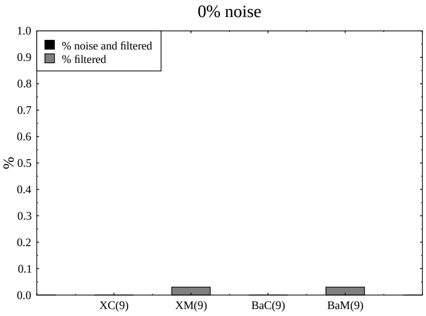


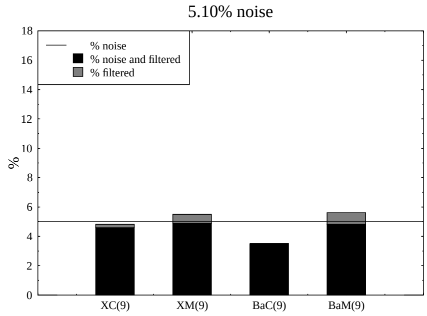
35.09% noise

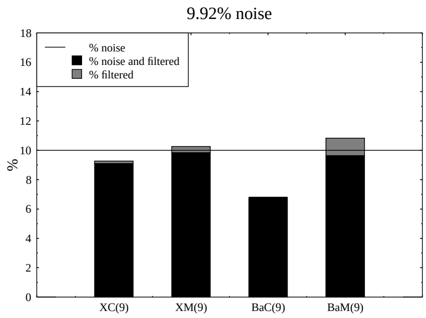


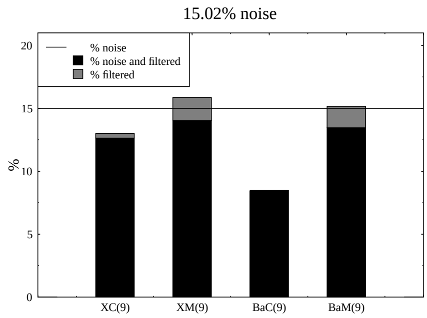
39.97% noise

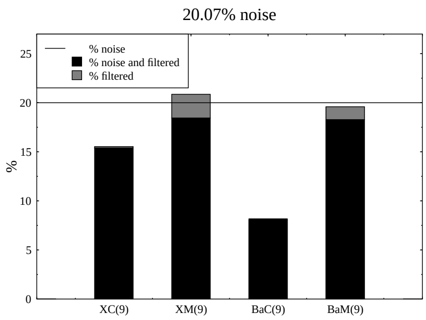


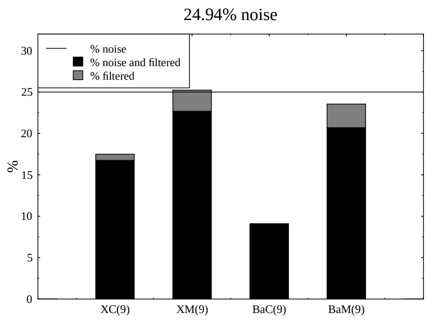

0% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

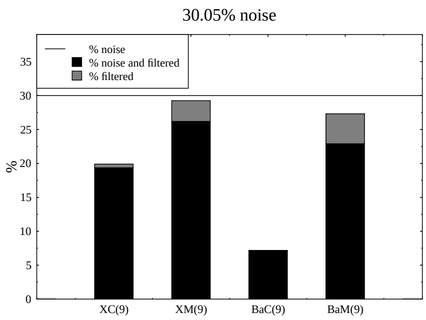

5.10% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

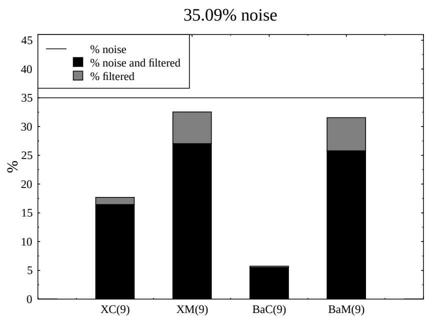

9.92% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

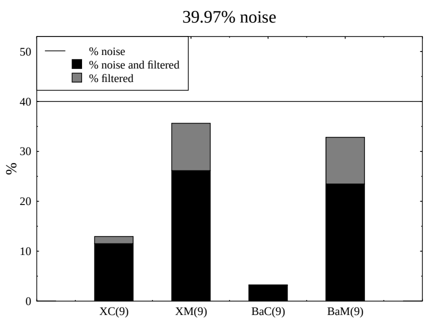

15.02% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


20.07% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

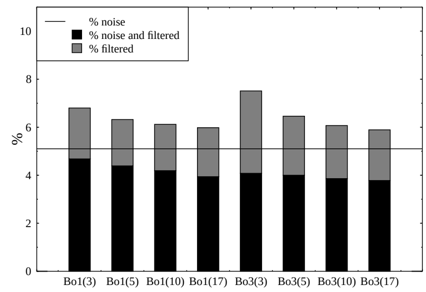

24.94% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


30.05% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

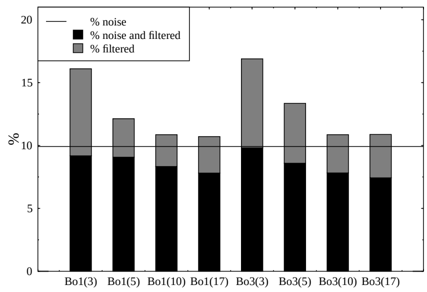

35.09% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


39.97% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

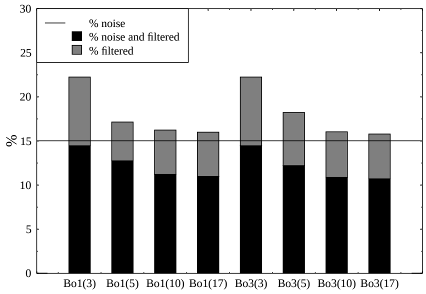
5.10% noise



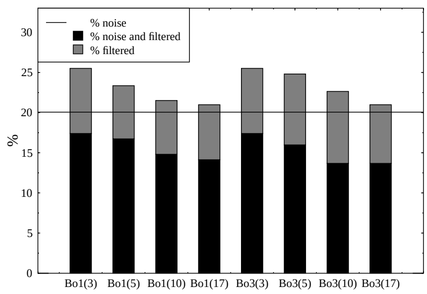
9.92% noise



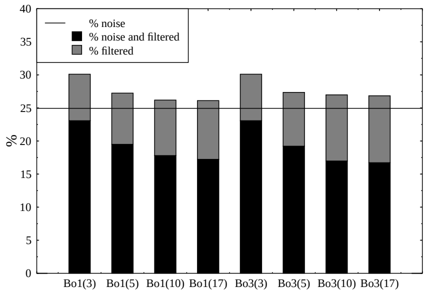
15.02% noise



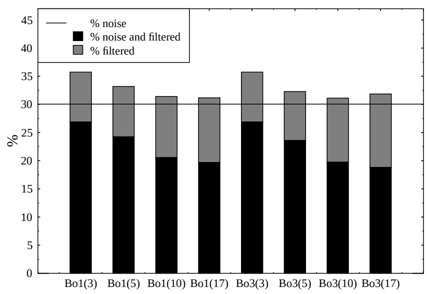
20.07% noise



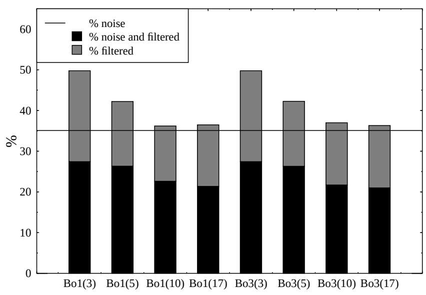
24.94% noise



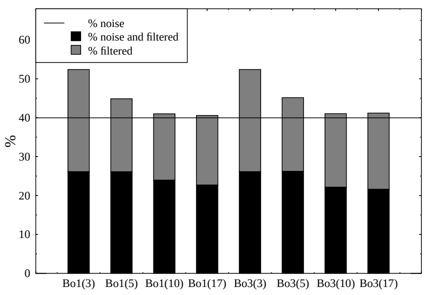
30.05% noise



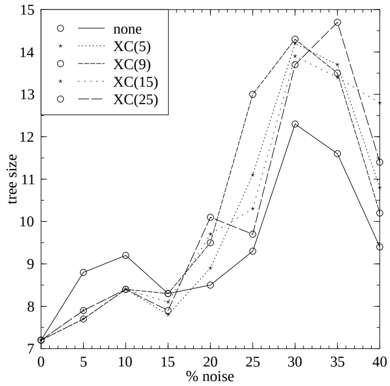
35.09% noise

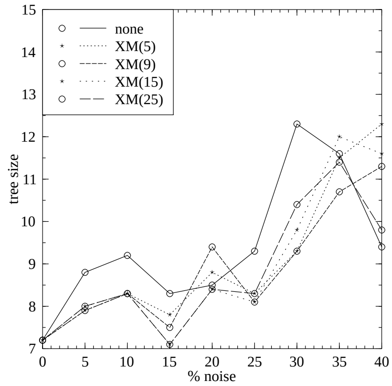


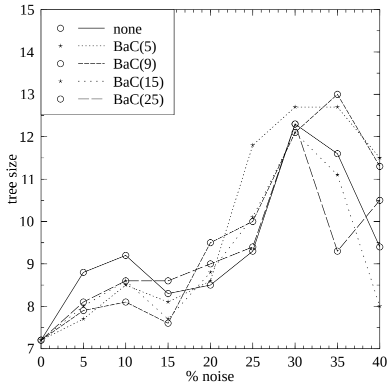
39.97% noise

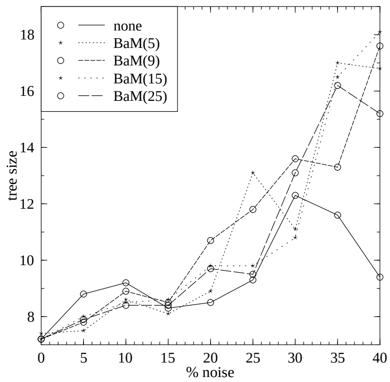


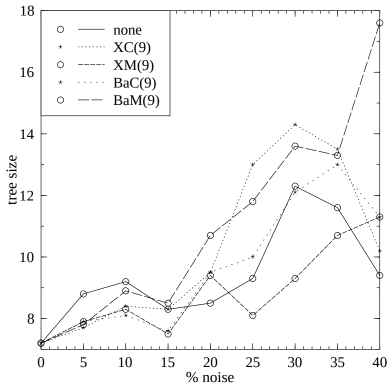
A.1.2 Tree size

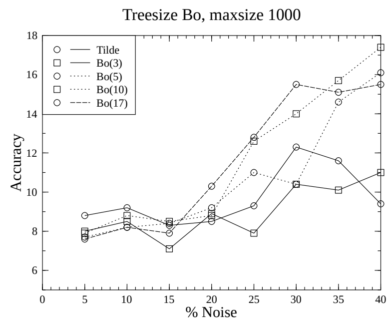


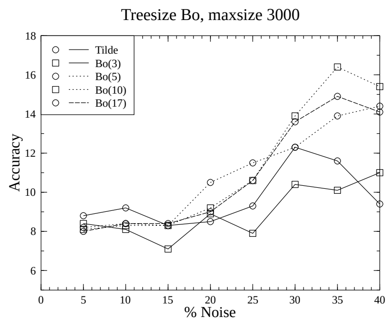




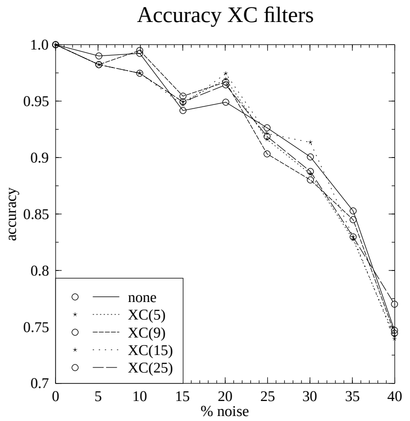


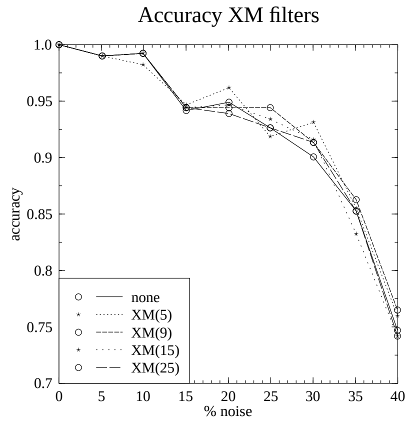

none
XC(9)
XM(9)
BaC(9)
BaM(9)
tree size
% noise

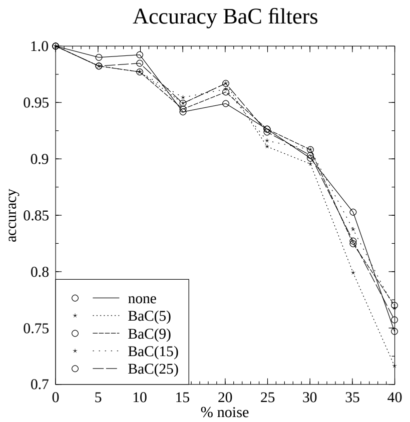

Treesize Bo, maxsize 1000
Tilde
Bo(3)
Bo(5)
Bo(10)
Bo(17)
Accuracy
% Noise

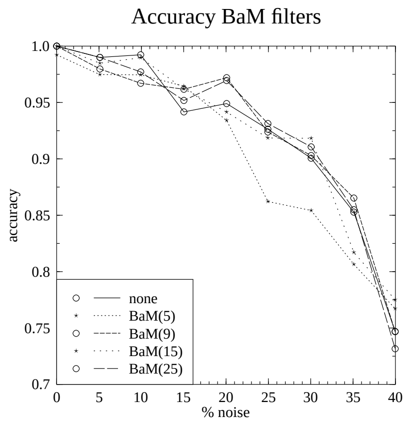

Treesize Bo, maxsize 3000
Tilde
Bo(3)
Bo(5)
Bo(10)
Bo(17)
Accuracy
% Noise

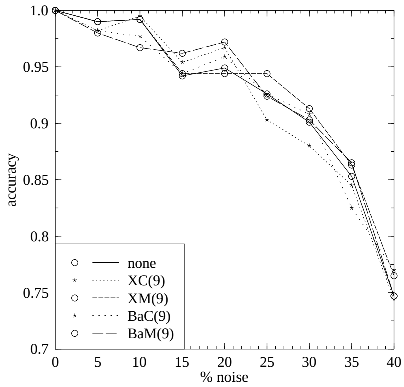
A.1.3 Accuracy

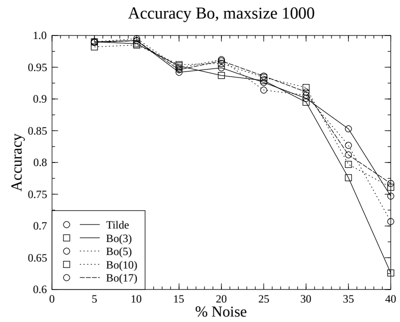


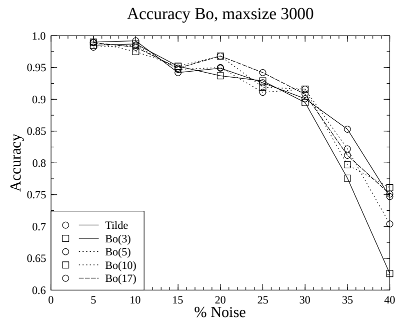




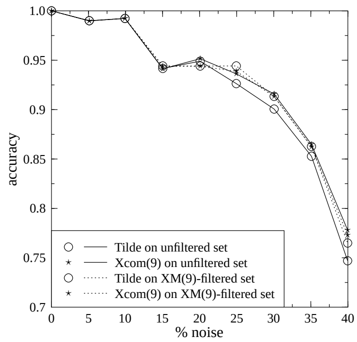


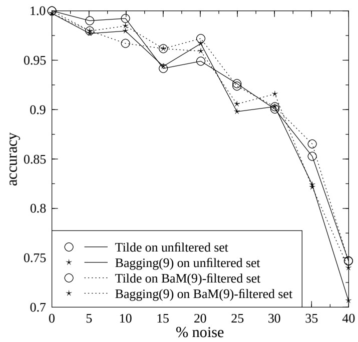

accuracy
% noise
none
XC(9)
XM(9)
BaC(9)
BaM(9)


Accuracy Bo, maxsize 1000
Accuracy
% Noise
Tilde
Bo(3)
Bo(5)
Bo(10)
Bo(17)


Accuracy Bo, maxsize 3000
Accuracy
% Noise
Tilde
Bo(3)
Bo(5)
Bo(10)
Bo(17)

A.1.4 Voting versus Filtering





B East-West Trains data set

The East-West Challenge problem has been proposed by Ryszard Michalski [12]. The task is to discover simple rules that distinguish examples of the Eastbound trains from those heading in the opposite direction. An example of such a rule is "If a train has a long closed car with more than one load, then it is heading west.". Each example in the data set is a train heading east or west, consisting of several cars. A car has a certain shape, lenght, number of wheels and might be double or not. It might have a roof with a specific form. Each car of the train has a load of a certain shape and number. A typical East-West Train data set is shown in Fig. 5.

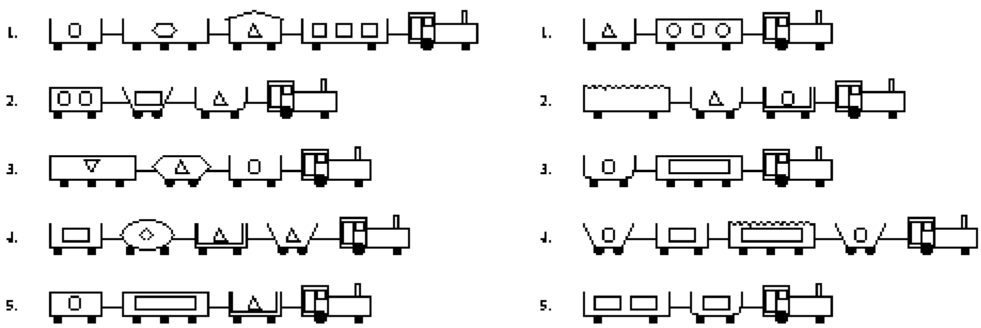


Figure 5: Trains example: the trains in the left column are eastbound trains, in the right column westbound trains.

B.1 Trains, 200 examples

B.1.1 Filter Precision

Cross-Validated Committees - Consensus filters

XC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.44	0.00	0.00	0.004	NA
5	4.06	3.39	1.68	0.007	0.322
10	7.11	6.50	3.70	0.007	0.347
15	7.89	7.28	8.15	0.007	0.509
20	10.72	9.78	11.27	0.012	0.509
25	8.44	7.89	18.37	0.007	0.682
30	10.78	9.22	23.04	0.022	0.691
35	9.06	7.72	29.85	0.020	0.779
40	5.39	4.17	37.80	0.020	0.896

XC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.50	0.00	0.00	0.005	NA
5	4.06	3.39	1.67	0.007	0.322
10	7.28	6.50	3.70	0.009	0.346
15	8.44	7.94	7.46	0.006	0.465
20	10.67	9.94	11.10	0.009	0.500
25	7.94	7.56	18.66	0.005	0.696
30	8.83	8.11	23.77	0.010	0.729
35	10.61	9.39	28.49	0.019	0.731
40	6.61	5.00	37.31	0.027	0.875

XC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.61	0.00	0.00	0.006	NA
5	3.94	3.33	1.73	0.006	0.333
10	7.39	6.67	3.52	0.008	0.330
15	8.33	7.83	7.56	0.006	0.472
20	10.39	9.72	11.31	0.008	0.511
25	9.22	8.83	17.51	0.005	0.644
30	10.33	9.28	22.91	0.015	0.690
35	12.56	10.50	27.67	0.032	0.700
40	6.28	4.83	37.41	0.024	0.879

XC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.67	0.00	0.00	0.007	NA
5	4.11	3.33	1.73	0.008	0.333
10	7.72	6.78	3.42	0.010	0.318
15	8.44	7.72	7.68	0.008	0.479
20	10.39	9.72	11.28	0.008	0.512
25	9.33	8.78	17.60	0.007	0.646
30	10.39	9.22	22.94	0.017	0.692
35	11.67	9.67	28.32	0.031	0.724
40	6.83	5.33	37.05	0.025	0.867

Cross-Validated Committees - Majority Vote filters

XM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	1.94	0.00	0.00	0.019	NA
5	7.22	4.28	0.78	0.031	0.144
10	10.89	8.50	1.61	0.027	0.145
15	14.61	11.11	4.33	0.041	0.251
20	18.89	14.72	6.36	0.052	0.260
25	21.56	16.61	10.34	0.066	0.330
30	24.33	17.50	16.32	0.097	0.415
35	28.28	19.67	21.25	0.132	0.437
40	28.83	18.56	30.13	0.171	0.535

XM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	1.89	0.00	0.00	0.019	NA
5	6.17	4.39	0.65	0.019	0.122
10	11.17	8.78	1.30	0.027	0.117
15	15.33	11.11	4.37	0.050	0.252
20	17.89	14.39	6.68	0.044	0.276
25	21.33	16.72	10.20	0.061	0.326
30	23.33	16.94	16.85	0.091	0.433
35	27.94	20.06	20.63	0.121	0.426
40	28.33	18.78	29.51	0.159	0.530

XM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	2.11	0.00	0.00	0.021	NA
5	6.39	4.33	0.71	0.022	0.133
10	11.17	8.61	1.49	0.028	0.134
15	15.33	11.44	3.99	0.046	0.229
20	18.33	14.33	6.79	0.050	0.279
25	21.11	16.56	10.39	0.061	0.332
30	23.22	16.78	17.04	0.092	0.439
35	28.11	20.28	20.36	0.120	0.420
40	28.50	18.39	30.26	0.168	0.540

XM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	2.22	0.00	0.00	0.022	NA
5	6.50	4.28	0.77	0.023	0.144
10	11.33	8.61	1.50	0.030	0.134
15	15.28	11.39	4.04	0.046	0.233
20	18.28	14.33	6.78	0.049	0.279
25	21.22	16.28	10.77	0.066	0.343
30	23.50	17.28	16.41	0.089	0.422
35	28.56	20.11	20.72	0.130	0.424
40	28.67	18.94	29.45	0.162	0.526

Bagging - Consensus filters

BaC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.39	0.00	0.00	0.004	NA
5	2.44	2.39	2.67	0.001	0.522
10	4.06	3.61	6.58	0.005	0.637
15	4.83	4.78	10.54	0.001	0.678
20	6.00	5.67	15.11	0.004	0.715
25	5.78	5.17	20.77	0.008	0.791
30	4.78	4.56	26.57	0.003	0.848
35	4.56	3.67	32.74	0.014	0.895
40	4.17	3.39	38.13	0.013	0.915

BaC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	1.56	1.56	3.49	0.000	0.689
10	3.39	3.39	6.77	0.000	0.659
15	3.44	3.39	11.83	0.001	0.772
20	3.67	3.50	16.98	0.002	0.825
25	1.89	1.78	23.43	0.001	0.928
30	3.33	3.28	27.51	0.001	0.890
35	2.61	2.44	33.36	0.003	0.930
40	1.44	1.06	39.46	0.006	0.974

BaC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	1.56	1.56	3.49	0.000	0.689
10	1.83	1.83	8.25	0.000	0.815
15	2.39	2.39	12.74	0.000	0.839
20	2.17	2.17	18.11	0.000	0.891
25	1.61	1.61	23.54	0.000	0.935
30	1.50	1.50	28.81	0.000	0.950
35	1.17	1.17	34.17	0.000	0.967
40	0.44	0.44	39.67	0.000	0.989

BaC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	0.72	0.72	4.31	0.000	0.856
10	1.56	1.56	8.52	0.000	0.843
15	2.17	2.17	12.93	0.000	0.854
20	1.44	1.44	18.71	0.000	0.928
25	1.00	1.00	24.01	0.000	0.959
30	0.67	0.67	29.41	0.000	0.978
35	1.11	1.11	34.20	0.000	0.968
40	0.33	0.22	39.85	0.002	0.994

Bagging - Majority Vote filters

BaM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	2.00	0.00	0.00	0.020	NA
5	5.67	4.06	1.00	0.017	0.189
10	10.72	7.56	2.66	0.035	0.241
15	13.94	10.94	4.50	0.035	0.263
20	18.39	12.78	8.69	0.070	0.358
25	20.33	13.78	13.76	0.087	0.444
30	22.78	15.89	18.10	0.098	0.468
35	26.61	16.22	25.48	0.160	0.536
40	26.56	14.50	34.60	0.201	0.637

BaM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	1.33	0.00	0.00	0.013	NA
5	5.72	4.22	0.82	0.016	0.156
10	9.67	7.56	2.63	0.023	0.240
15	13.28	10.61	4.85	0.031	0.284
20	16.28	12.61	8.67	0.046	0.366
25	19.78	14.94	12.25	0.064	0.397
30	21.72	15.78	17.99	0.085	0.472
35	25.17	17.72	22.92	0.114	0.493
40	24.94	14.72	33.60	0.170	0.631

BaM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	1.33	0.00	0.00	0.013	NA
5	5.44	3.94	1.12	0.016	0.211
10	10.22	8.28	1.85	0.022	0.167
15	12.89	10.72	4.68	0.025	0.277
20	16.06	12.89	8.30	0.040	0.352
25	19.00	15.06	11.98	0.052	0.393
30	20.06	15.39	18.11	0.067	0.485
35	24.00	16.50	24.25	0.115	0.528
40	23.72	14.56	33.29	0.153	0.636

BaM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	1.17	0.00	0.00	0.012	NA
5	5.22	4.11	0.94	0.012	0.178
10	9.33	8.28	1.83	0.012	0.167
15	12.94	11.28	4.05	0.020	0.240
20	16.17	13.72	7.30	0.031	0.310
25	18.50	14.83	12.18	0.049	0.401
30	19.56	15.44	17.92	0.059	0.484
35	23.00	16.56	23.82	0.099	0.526
40	23.22	15.17	32.21	0.134	0.620

Boosting filter

Bo(3)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	10.72	2.94	2.32	0.082	0.411
10	17.33	6.67	4.04	0.118	0.330
15	23.28	8.67	8.06	0.171	0.417
20	29.11	10.83	12.46	0.228	0.454
25	33.44	11.28	19.97	0.295	0.544
30	39.78	16.89	21.77	0.326	0.436
35	51.22	23.11	24.14	0.432	0.338
40	54.83	25.50	31.54	0.488	0.361

Bo(5)

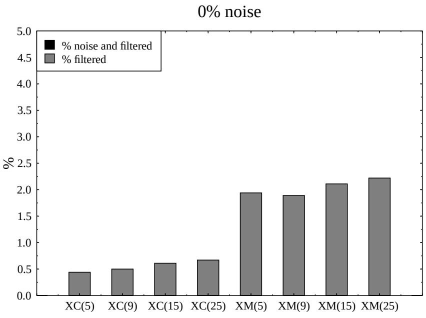
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	9.33	2.50	2.75	0.072	0.500
10	13.67	4.78	6.00	0.099	0.519
15	17.94	8.06	8.25	0.116	0.457
20	23.83	9.56	13.59	0.178	0.519
25	28.33	12.78	16.73	0.207	0.484
30	32.72	13.83	23.85	0.269	0.537
35	39.06	17.17	29.15	0.336	0.509
40	43.44	20.33	34.64	0.385	0.491

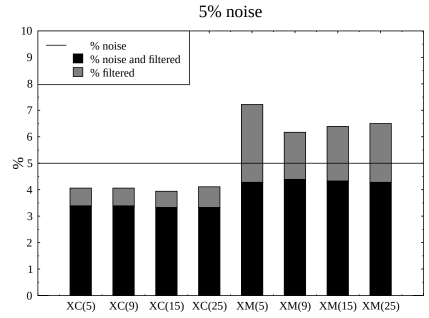
Bo(10)

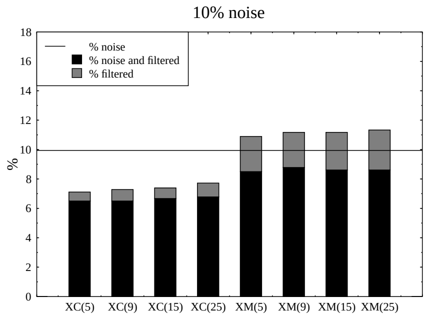
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	7.33	2.28	2.94	0.053	0.544
10	12.67	4.89	5.79	0.086	0.508
15	17.44	7.17	9.27	0.121	0.517
20	22.06	9.11	13.83	0.162	0.542
25	26.94	11.39	18.33	0.207	0.540
30	32.00	13.11	24.69	0.269	0.561
35	36.78	15.00	31.54	0.335	0.571
40	41.33	19.11	35.52	0.370	0.521

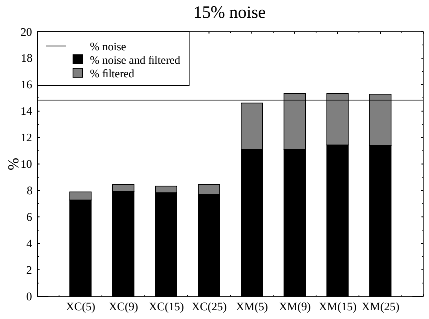
Bo(17)

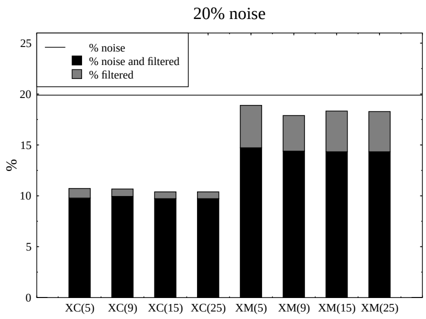
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	7.39	2.33	2.88	0.053	0.533
10	12.50	4.83	5.84	0.085	0.513
15	17.06	7.11	9.31	0.117	0.521
20	21.67	9.11	13.76	0.157	0.542
25	26.89	11.33	18.40	0.207	0.543
30	31.78	13.06	24.69	0.267	0.563
35	36.72	15.17	31.24	0.331	0.566
40	41.28	19.33	35.10	0.365	0.516

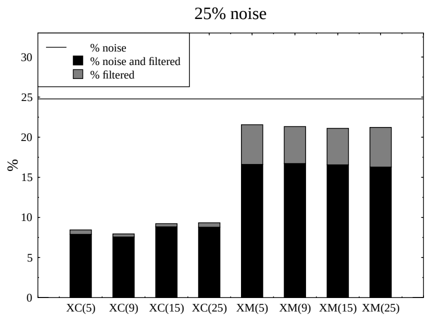

0% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

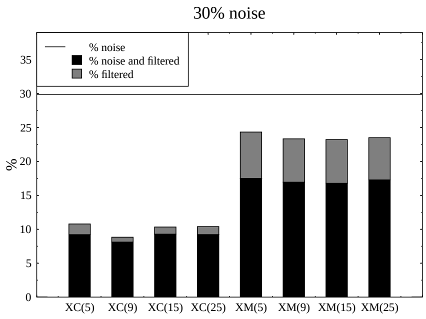

5% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

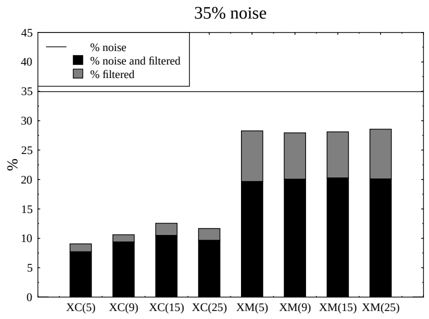

10% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

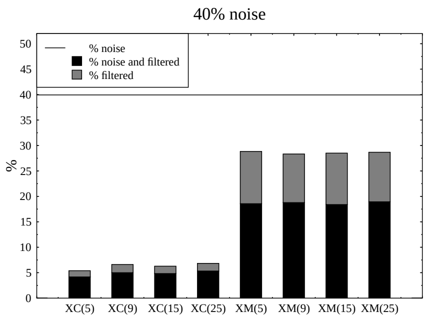

15% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

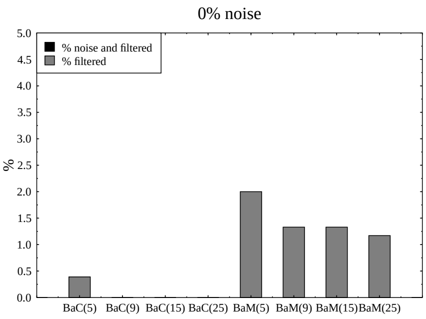

20% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

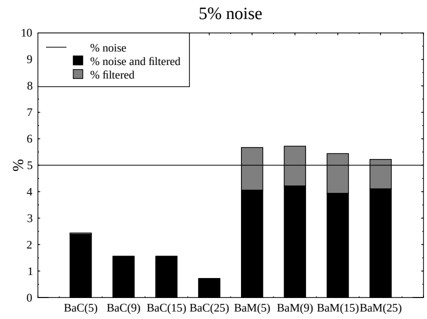

25% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

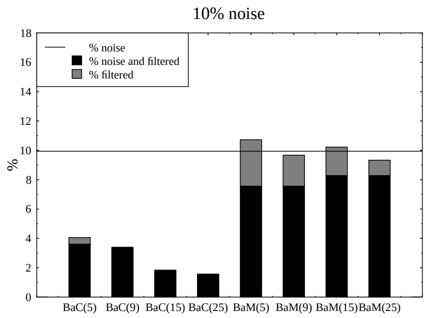

30% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

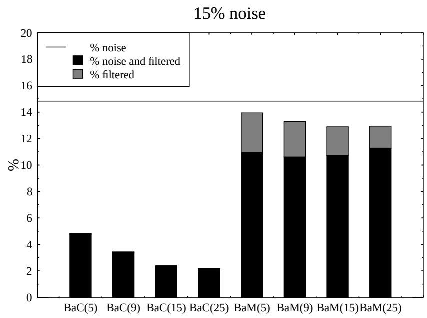

35% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

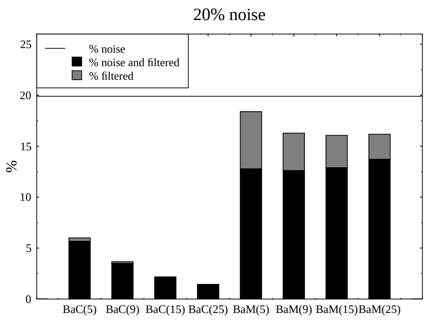

40% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

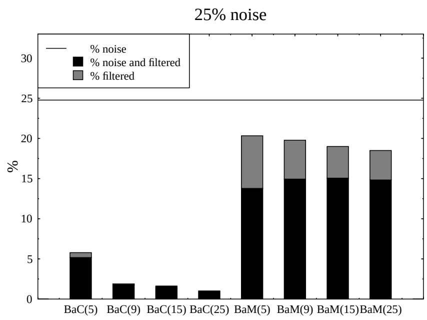

0% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

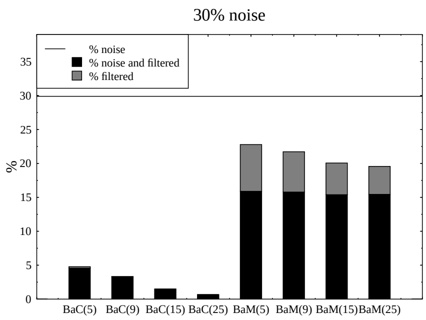

5% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

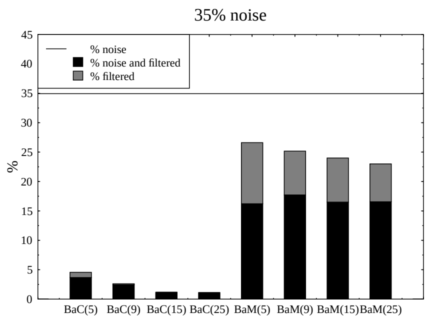

10% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

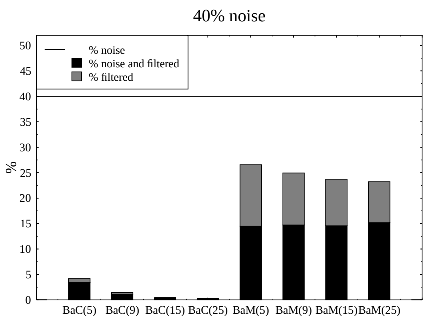

15% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

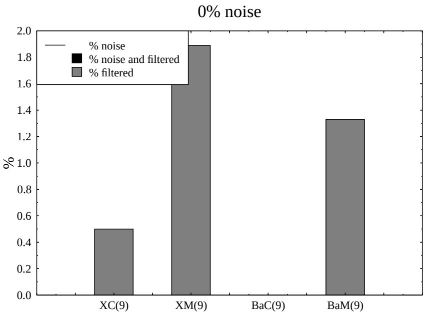

20% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

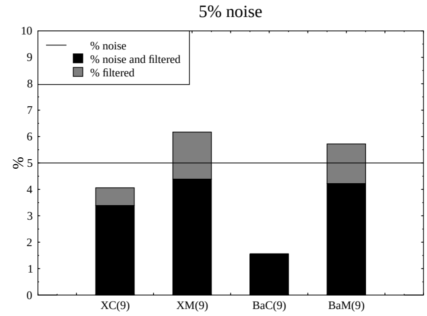

25% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

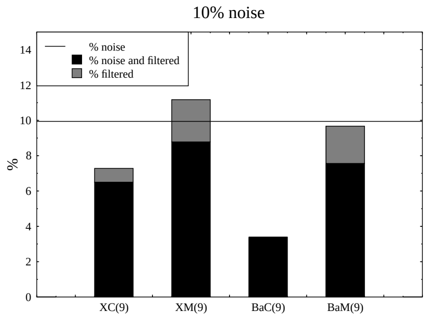

30% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

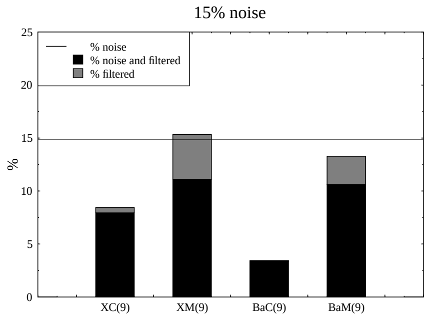

35% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

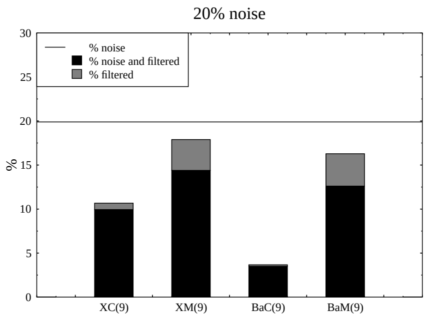

40% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

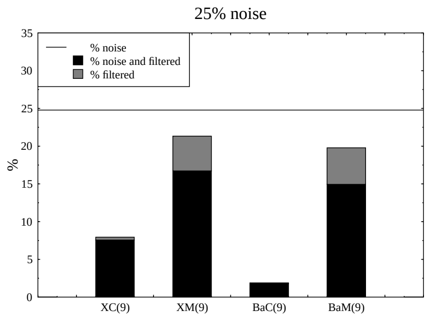

0% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

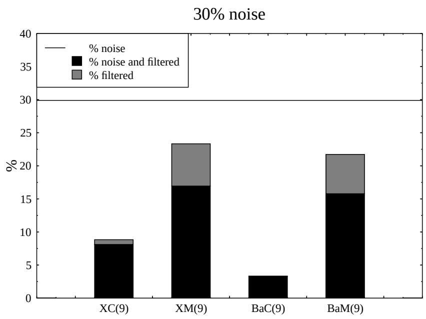

5% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

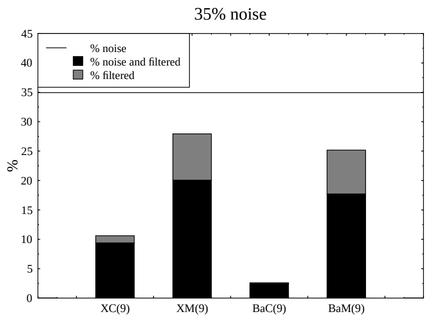

10% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

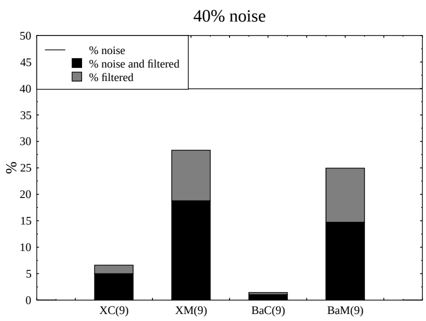

15% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


20% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

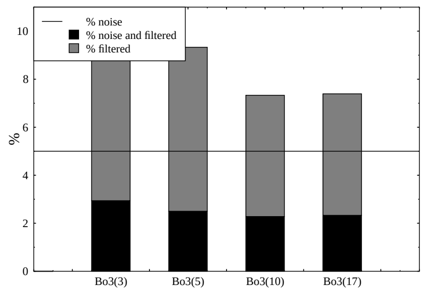

25% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


30% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

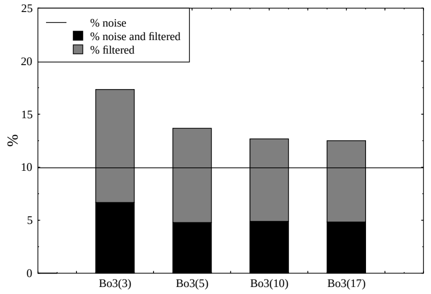

35% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


40% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

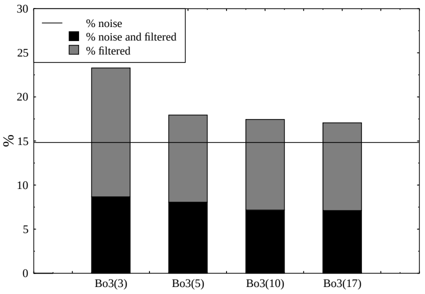
5% noise



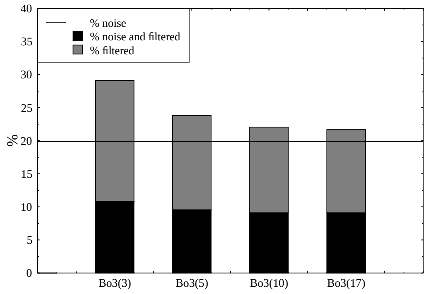
10% noise



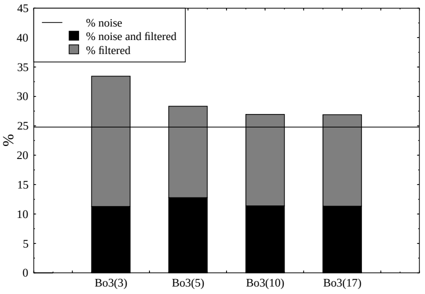
15% noise



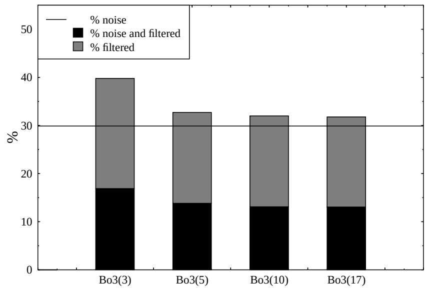
20% noise



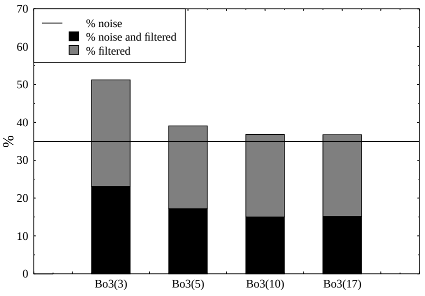
25% noise



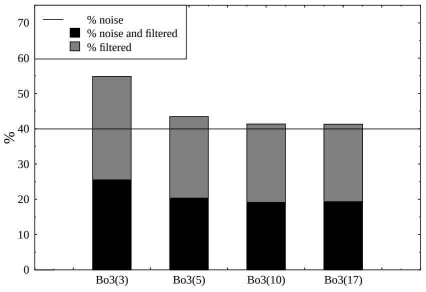
30% noise



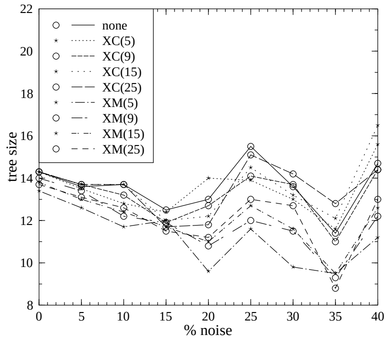
35% noise

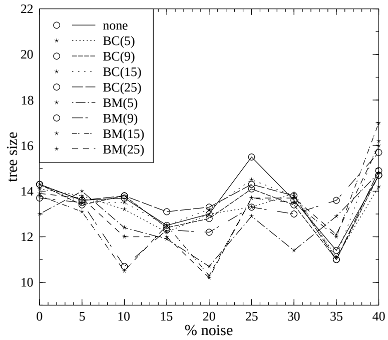


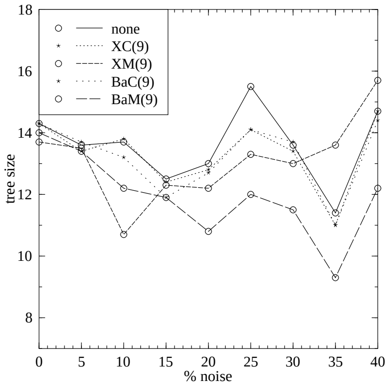
40% noise

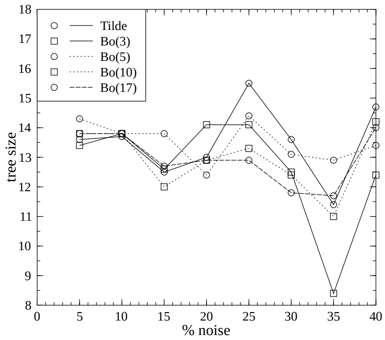


B.1.2 Tree size

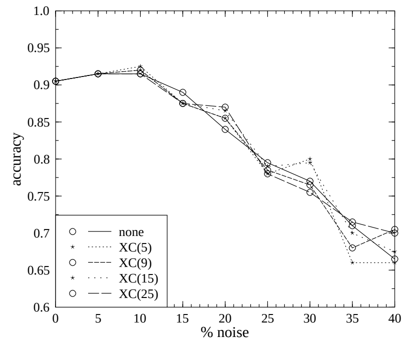


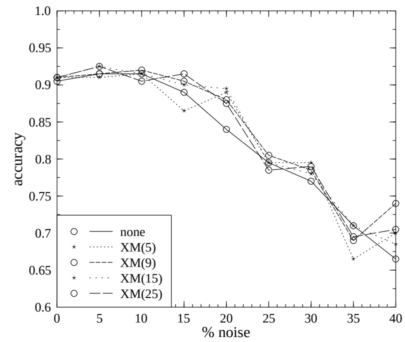


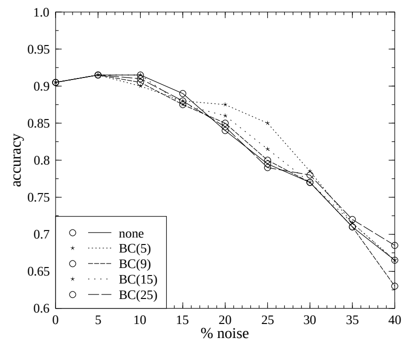


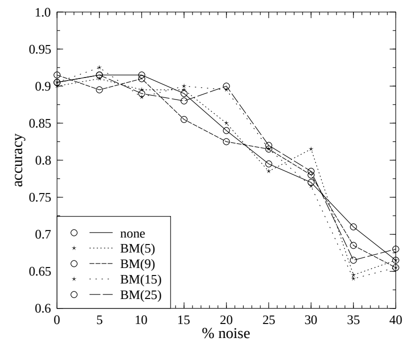


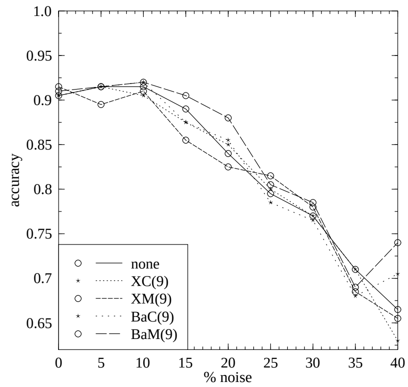
B.1.3 Accuracy

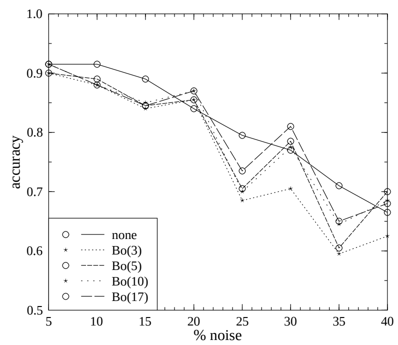




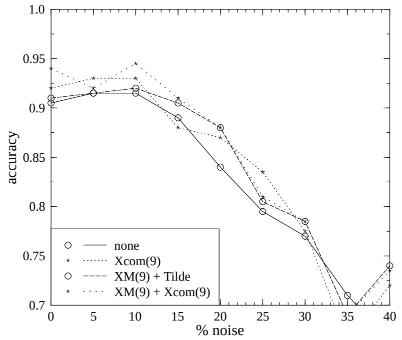


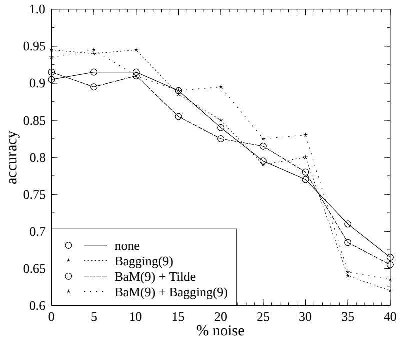






B.1.4 Voting versus Filtering





B.2 Trains, 400 examples

B.2.1 Filter Precision

Cross-Validated Committees - Consensus filters

XC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	4.11	4.08	0.92	0.000	0.178
10	7.78	7.50	2.70	0.003	0.250
15	11.19	11.03	4.44	0.002	0.265
20	12.08	11.58	9.50	0.006	0.420
25	13.89	13.72	12.98	0.002	0.449
30	12.50	12.03	20.41	0.007	0.598
35	11.17	10.33	27.61	0.013	0.705
40	9.14	7.33	35.69	0.030	0.816

XC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5	4.36	4.25	0.75	0.001	0.145
10	7.67	7.53	2.67	0.002	0.247
15	11.03	10.81	4.68	0.003	0.280
20	12.56	12.03	9.06	0.007	0.398
25	14.19	14.00	12.67	0.003	0.438
30	13.64	13.14	19.40	0.007	0.561
35	12.33	11.75	26.42	0.009	0.664
40	9.78	8.42	34.75	0.023	0.788

XC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5	4.17	4.14	0.87	0.000	0.167
10	7.75	7.39	2.83	0.004	0.261
15	11.50	11.28	4.17	0.003	0.248
20	12.53	11.75	9.36	0.010	0.412
25	13.72	13.61	13.03	0.001	0.454
30	14.14	13.47	19.13	0.010	0.550
35	11.53	11.08	26.89	0.007	0.683
40	10.17	8.44	34.84	0.029	0.788

XC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5	4.28	4.22	0.78	0.001	0.151
10	7.75	7.39	2.83	0.004	0.261
15	11.58	11.47	3.96	0.001	0.235
20	12.56	11.78	9.32	0.010	0.410
25	14.67	14.53	12.10	0.002	0.417
30	13.75	13.08	19.47	0.010	0.563
35	12.00	11.42	26.69	0.009	0.674
40	9.72	8.03	35.12	0.028	0.798

Cross-Validated Committees - Majority Vote filters

XM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.67	0.00	0.00	0.007	NA
5	5.58	4.56	0.44	0.011	0.084
10	10.22	9.08	1.02	0.013	0.092
15	15.03	13.61	1.63	0.017	0.093
20	18.86	16.06	4.79	0.035	0.196
25	23.42	20.28	6.04	0.042	0.186
30	26.56	21.83	11.04	0.067	0.271
35	28.22	22.58	17.24	0.087	0.354
40	30.39	21.75	25.91	0.144	0.454

XM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.67	0.00	0.00	0.007	NA
5	5.61	4.58	0.41	0.011	0.078
10	10.36	8.94	1.18	0.016	0.106
15	15.22	13.39	1.90	0.022	0.107
20	18.81	16.08	4.76	0.034	0.195
25	23.31	20.11	6.25	0.043	0.193
30	25.67	21.86	10.83	0.054	0.270
35	28.11	22.86	16.81	0.081	0.346
40	30.17	20.56	27.52	0.160	0.483

XM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.64	0.00	0.00	0.006	NA
5	5.47	4.53	0.47	0.010	0.089
10	10.22	9.06	1.05	0.013	0.094
15	14.89	13.39	1.89	0.018	0.107
20	18.86	16.22	4.59	0.033	0.188
25	22.94	20.31	5.97	0.035	0.185
30	26.39	22.33	10.32	0.058	0.254
35	28.50	21.81	18.41	0.103	0.376
40	29.83	20.56	27.40	0.154	0.484

XM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.61	0.00	0.00	0.006	NA
5	5.44	4.56	0.44	0.009	0.083
10	10.28	9.06	1.05	0.014	0.094
15	15.39	13.28	2.03	0.025	0.115
20	18.92	16.08	4.76	0.035	0.195
25	22.94	20.42	5.82	0.034	0.180
30	26.22	21.86	10.93	0.062	0.270
35	28.39	21.83	18.33	0.101	0.376
40	29.78	20.47	27.50	0.155	0.486

Bagging - Consensus filters

BaC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	3.42	3.33	1.70	0.001	0.330
10	5.36	5.28	4.98	0.001	0.472
15	7.53	7.47	8.11	0.001	0.502
20	9.14	8.97	12.06	0.002	0.551
25	9.83	9.72	16.82	0.001	0.610
30	8.08	7.78	24.06	0.004	0.740
35	6.17	5.67	31.20	0.008	0.838
40	4.97	4.28	37.36	0.012	0.893

BaC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	2.56	2.56	2.48	0.000	0.485
10	4.69	4.69	5.56	0.000	0.531
15	6.42	6.39	9.20	0.000	0.574
20	6.81	6.78	14.13	0.000	0.661
25	7.50	7.47	18.81	0.000	0.700
30	5.61	5.61	25.76	0.000	0.813
35	4.08	3.94	32.33	0.002	0.887
40	2.22	1.89	38.77	0.006	0.953

BaC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	2.25	2.25	2.78	0.000	0.547
10	3.78	3.78	6.46	0.000	0.622
15	5.19	5.19	10.34	0.000	0.654
20	5.89	5.89	14.95	0.000	0.705
25	6.36	6.36	19.80	0.000	0.745
30	3.97	3.97	27.03	0.000	0.867
35	2.56	2.50	33.32	0.001	0.929
40	1.36	1.36	38.97	0.000	0.966

BaC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	2.14	2.14	2.89	0.000	0.570
10	3.56	3.56	6.68	0.000	0.644
15	4.03	4.03	11.43	0.000	0.731
20	4.58	4.58	16.12	0.000	0.771
25	3.72	3.72	21.98	0.000	0.851
30	2.75	2.75	27.95	0.000	0.908
35	2.08	2.08	33.57	0.000	0.940
40	1.14	1.08	39.16	0.001	0.973

Bagging - Majority Vote filters

BaM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.83	0.00	0.00	0.008	NA
5	5.89	4.42	0.59	0.015	0.111
10	10.08	8.58	1.57	0.017	0.142
15	14.44	13.08	2.24	0.016	0.128
20	18.00	14.67	6.45	0.042	0.266
25	22.31	18.56	8.17	0.050	0.255
30	25.08	19.47	13.96	0.080	0.350
35	27.56	19.28	21.64	0.127	0.449
40	28.47	17.86	30.69	0.176	0.551

BaM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.50	0.00	0.00	0.005	NA
5	5.31	4.64	0.35	0.007	0.067
10	9.50	8.58	1.56	0.010	0.142
15	14.33	13.31	1.98	0.012	0.113
20	17.25	14.94	6.06	0.029	0.252
25	21.14	18.19	8.51	0.039	0.270
30	24.56	19.50	13.83	0.072	0.349
35	27.00	19.44	21.26	0.116	0.444
40	26.64	17.92	29.86	0.145	0.550

BaM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.56	0.00	0.00	0.006	NA
5	5.19	4.53	0.47	0.007	0.089
10	9.72	8.92	1.20	0.009	0.108
15	14.36	13.28	2.00	0.013	0.115
20	17.36	15.50	5.40	0.023	0.224
25	21.14	18.47	8.16	0.036	0.259
30	23.56	20.03	12.96	0.050	0.331
35	25.19	18.89	21.51	0.097	0.460
40	26.36	17.08	30.88	0.154	0.571

BaM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.33	0.00	0.00	0.003	NA
5	5.08	4.50	0.50	0.006	0.095
10	9.61	8.86	1.26	0.008	0.114
15	14.06	13.17	2.13	0.010	0.122
20	17.33	15.36	5.56	0.025	0.231
25	20.78	18.36	8.25	0.032	0.263
30	23.19	19.28	13.87	0.056	0.356
35	24.75	20.11	19.71	0.071	0.425
40	25.50	17.44	30.00	0.134	0.562

Boosting filter

Bo(3)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	6.69	4.36	0.65	0.025	0.122
10	12.14	9.11	1.01	0.034	0.089
15	17.89	12.86	2.62	0.059	0.143
20	29.22	13.64	8.88	0.195	0.317
25	30.42	16.08	12.86	0.191	0.355
30	39.44	18.94	18.03	0.293	0.367
35	48.86	21.72	25.25	0.417	0.379
40	50.92	22.56	35.56	0.471	0.433

Bo(5)

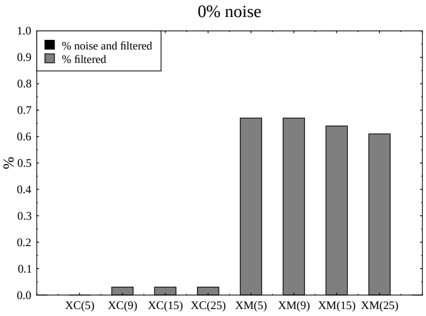
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	7.56	2.97	2.16	0.048	0.402
10	14.78	5.86	4.88	0.099	0.414
15	19.36	9.47	6.87	0.116	0.369
20	28.39	12.11	11.03	0.203	0.394
25	30.50	15.19	14.02	0.204	0.390
30	38.42	16.94	21.36	0.307	0.434
35	39.06	18.92	26.32	0.310	0.459
40	44.33	20.14	35.35	0.402	0.494

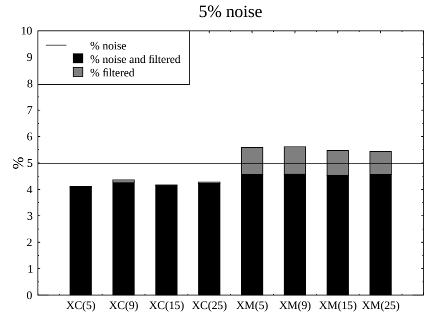
Bo(10)

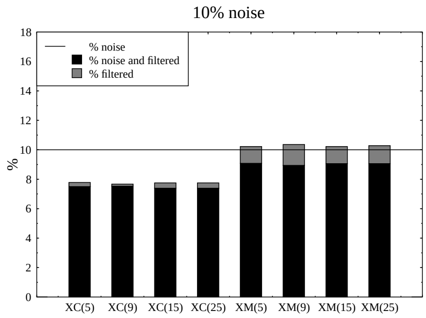
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	6.33	2.83	2.28	0.037	0.430
10	12.19	5.64	4.98	0.073	0.436
15	17.44	8.67	7.69	0.103	0.422
20	25.53	10.83	12.37	0.184	0.458
25	27.14	13.64	15.48	0.180	0.453
30	32.53	15.58	21.35	0.242	0.480
35	36.97	17.81	27.24	0.295	0.491
40	42.14	20.11	34.06	0.366	0.495

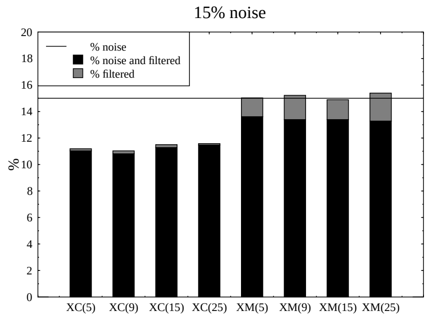
Bo(17)

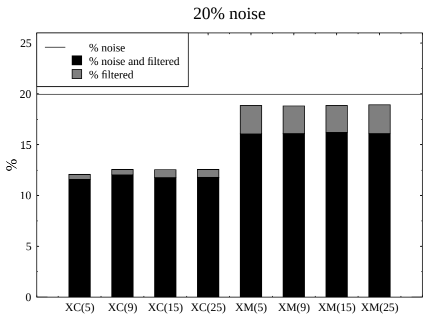
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	6.28	2.92	2.18	0.035	0.413
10	12.58	5.44	5.22	0.079	0.456
15	16.53	8.61	7.65	0.093	0.426
20	22.81	10.92	11.74	0.149	0.454
25	26.51	13.49	15.54	0.173	0.458
30	31.31	15.58	20.90	0.224	0.480
35	37.08	18.00	26.97	0.293	0.485
40	41.22	20.14	33.49	0.350	0.494

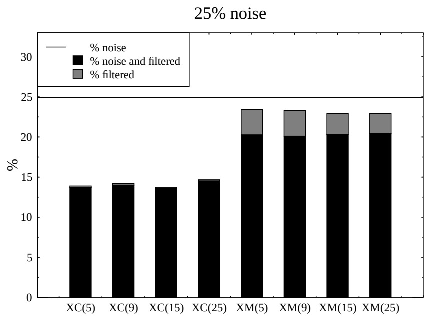

0% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

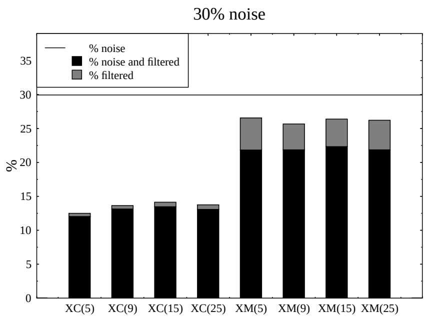

5% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

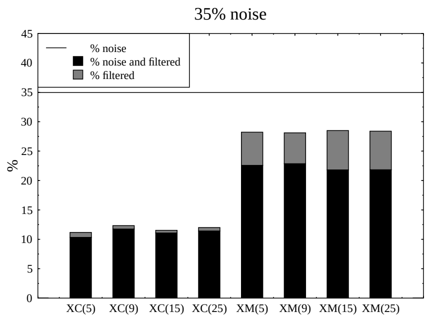

10% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

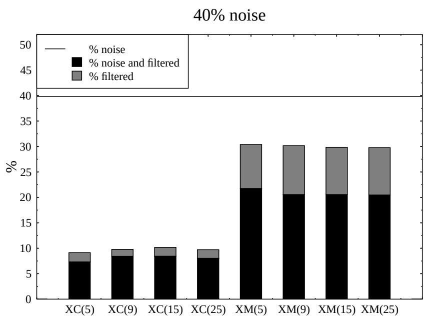

15% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

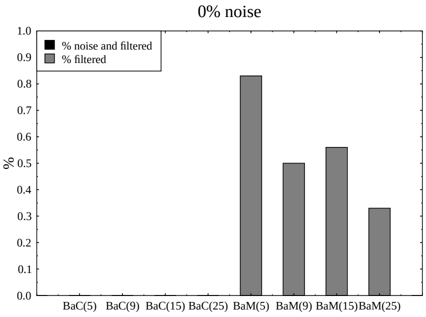

20% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

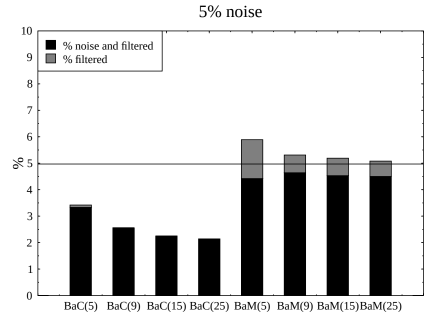

25% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

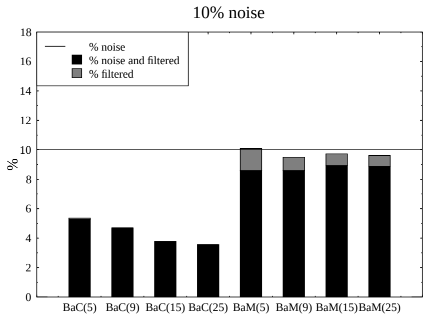

30% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

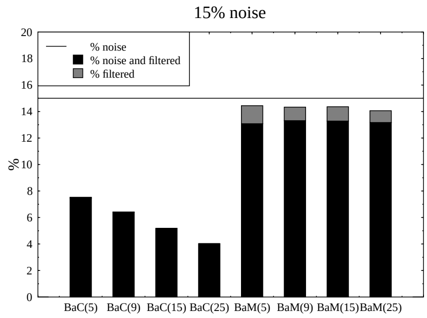

35% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

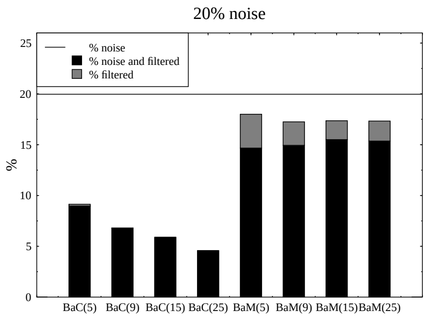

40% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

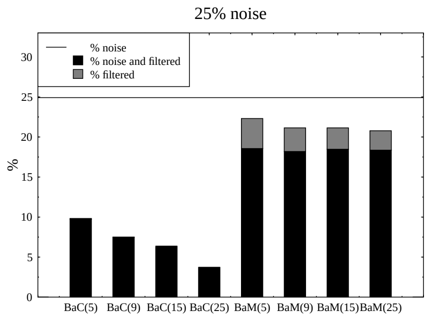

0% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

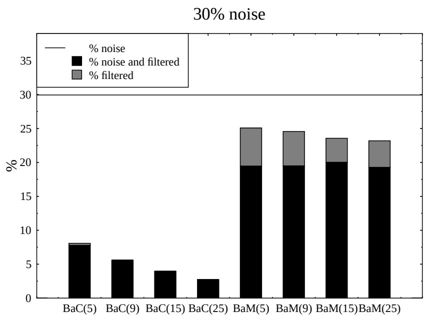

5% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

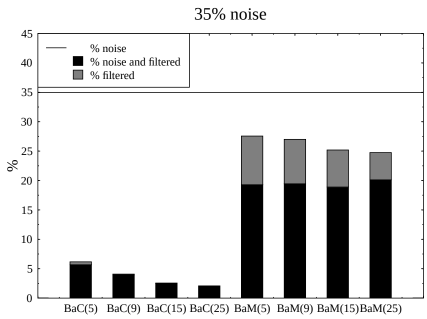

10% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

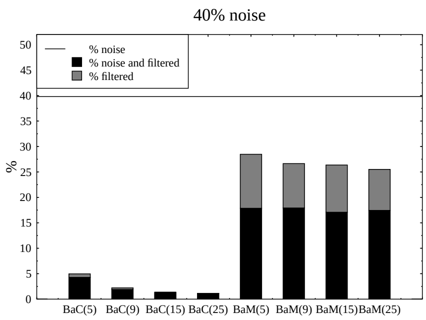

15% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

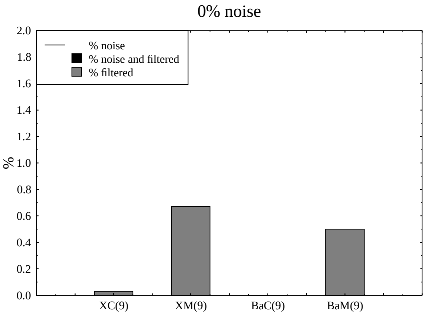

20% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

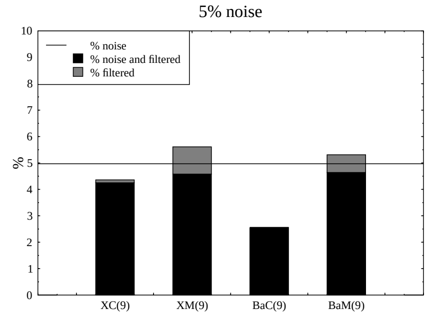

25% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

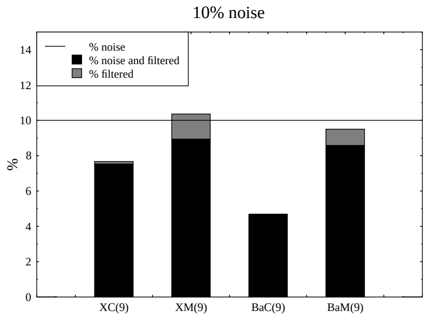

30% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

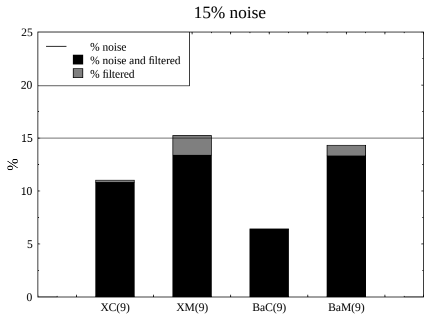

35% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

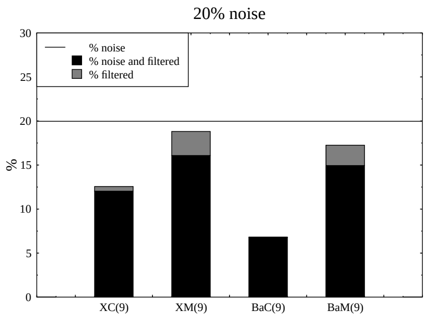

40% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

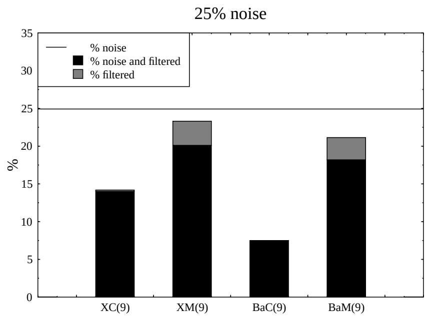

0% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

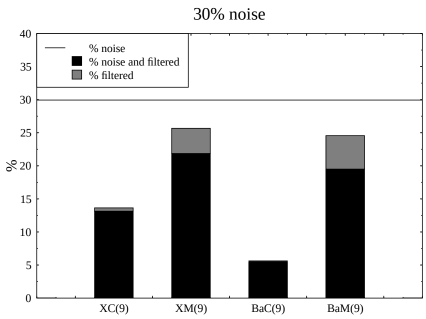

5% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

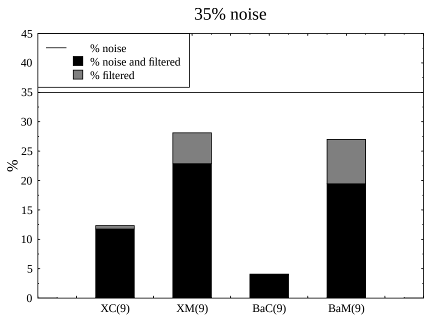

10% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

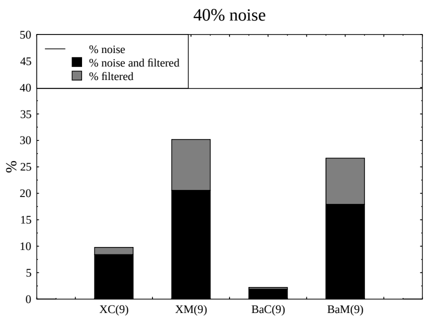

15% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


20% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

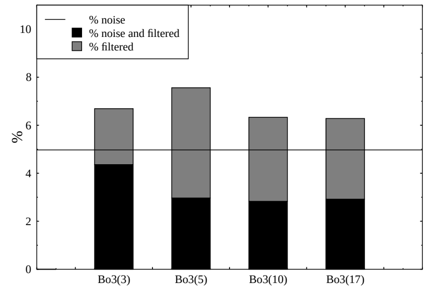

25% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


30% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

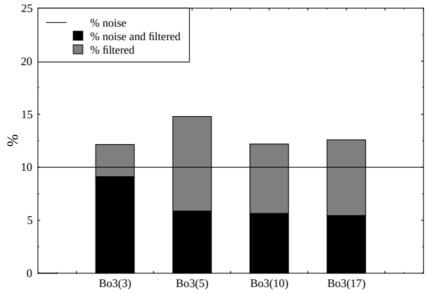

35% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


40% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

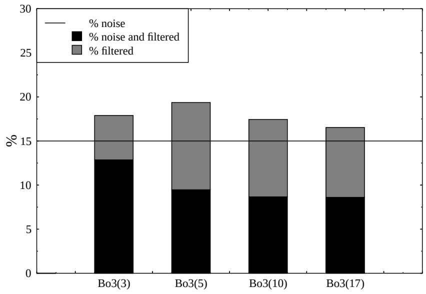
5% noise



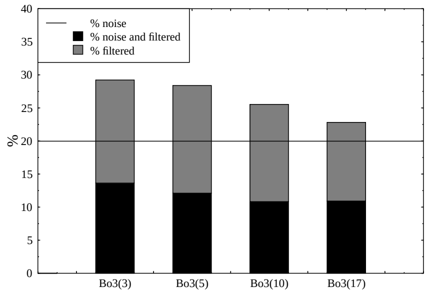
10% noise



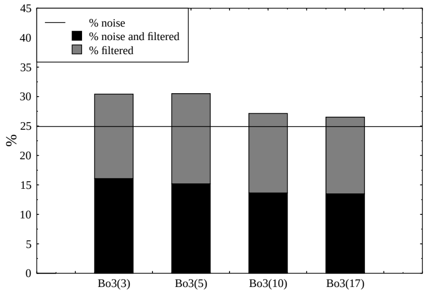
15% noise



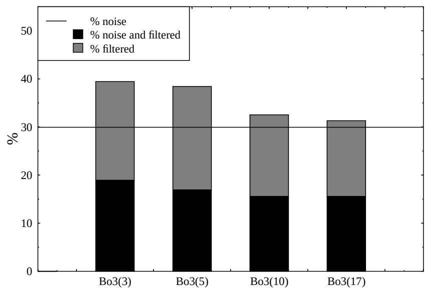
20% noise



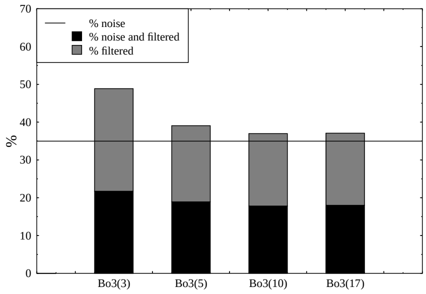
25% noise



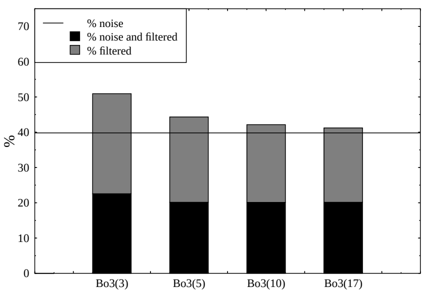
30% noise



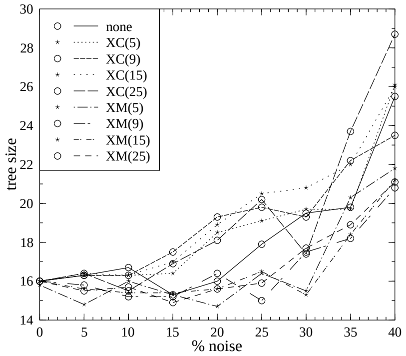
35% noise

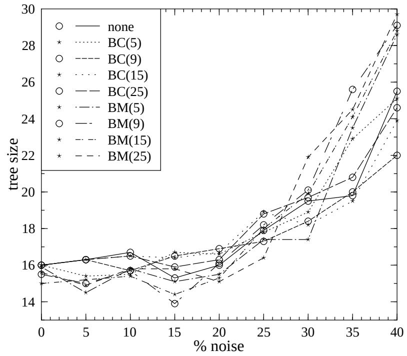


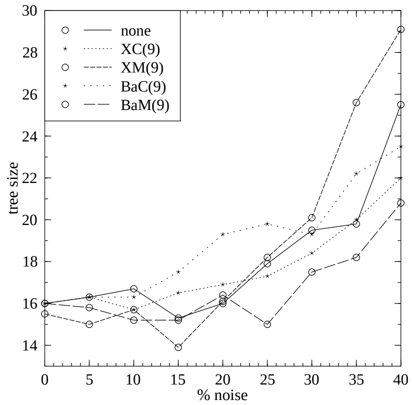
40% noise

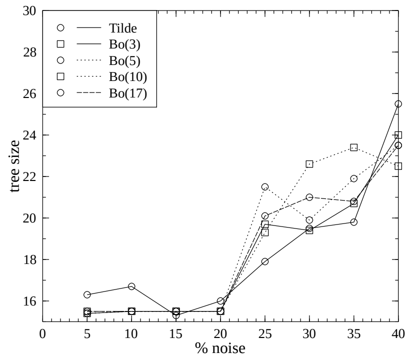


B.2.2 Tree size

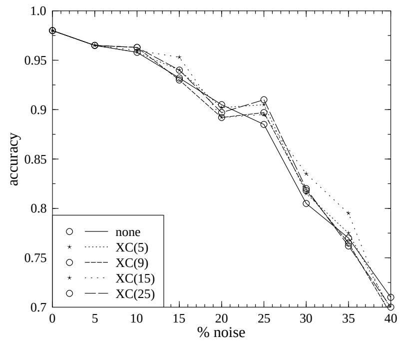


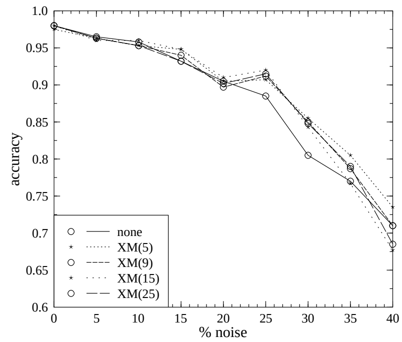


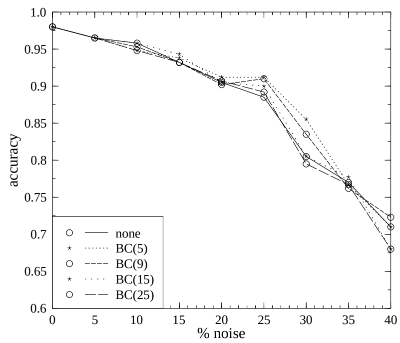


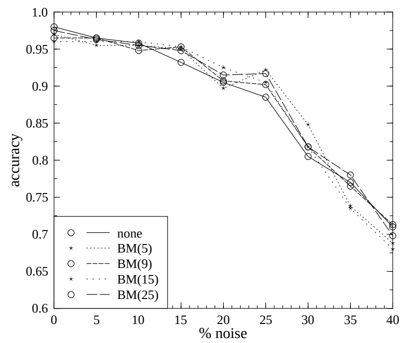


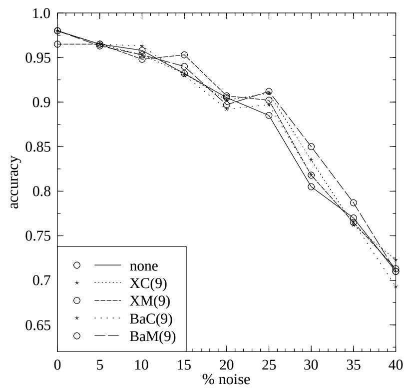
B.2.3 Accuracy

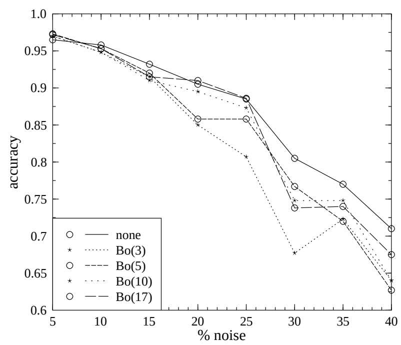




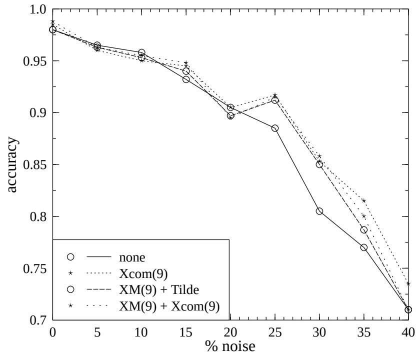


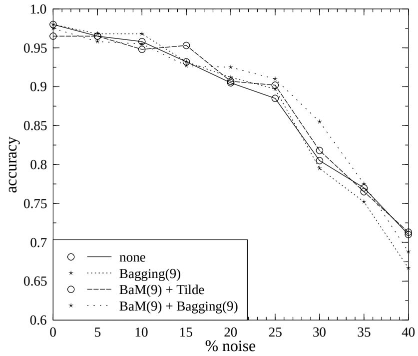






B.2.4 Voting versus Filtering





B.3 Trains, 800 examples

B.3.1 Filter Precision

Cross-Validated Committees - Consensus filters

XC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.03	0.00	0.00	0.000	NA
5	4.58	4.50	0.51	0.001	0.097
10	8.65	8.56	1.56	0.001	0.143
15	12.29	12.12	3.25	0.002	0.191
20	15.00	14.76	6.10	0.003	0.260
25	18.53	18.19	8.29	0.004	0.271
30	19.32	18.85	13.70	0.007	0.370
35	17.57	16.96	21.76	0.009	0.515
40	13.19	12.21	31.90	0.016	0.694

XC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.04	0.00	0.00	0.000	NA
5	4.64	4.56	0.45	0.001	0.086
10	8.64	8.56	1.56	0.001	0.143
15	12.21	12.04	3.34	0.002	0.196
20	15.32	15.17	5.65	0.002	0.240
25	17.62	17.44	9.10	0.002	0.301
30	18.53	18.06	14.53	0.007	0.397
35	18.22	17.42	21.22	0.012	0.501
40	12.90	12.17	31.83	0.012	0.695

XC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.08	0.00	0.00	0.001	NA
5	4.69	4.57	0.44	0.001	0.083
10	8.72	8.58	1.53	0.002	0.141
15	12.24	12.11	3.26	0.001	0.192
20	15.47	15.31	5.49	0.002	0.233
25	18.28	17.93	8.59	0.005	0.282
30	18.49	18.12	14.44	0.005	0.394
35	17.76	17.14	21.49	0.010	0.509
40	12.14	11.42	32.39	0.012	0.714

XC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.10	0.00	0.00	0.001	NA
5	4.72	4.60	0.41	0.001	0.078
10	8.76	8.60	1.52	0.002	0.139
15	12.28	12.11	3.26	0.002	0.192
20	15.50	15.24	5.57	0.003	0.237
25	18.69	18.29	8.18	0.005	0.267
30	18.85	18.43	14.13	0.006	0.384
35	17.29	16.68	21.91	0.009	0.523
40	13.38	12.51	31.58	0.014	0.687

Cross-Validated Committees - Majority Vote filters

XM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.25	0.00	0.00	0.003	NA
5	5.49	4.86	0.13	0.007	0.025
10	10.24	9.51	0.53	0.008	0.047
15	15.12	13.93	1.24	0.014	0.070
20	19.50	18.07	2.34	0.018	0.095
25	24.42	22.47	3.29	0.026	0.100
30	28.00	25.43	6.24	0.037	0.150
35	31.26	26.94	11.61	0.066	0.229
40	33.08	25.64	21.36	0.124	0.358

XM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.31	0.00	0.00	0.003	NA
5	5.50	4.83	0.16	0.007	0.031
10	10.31	9.50	0.54	0.009	0.049
15	15.04	13.97	1.19	0.013	0.068
20	19.24	18.26	2.09	0.012	0.085
25	24.08	22.35	3.44	0.023	0.105
30	27.78	25.32	6.35	0.035	0.154
35	30.57	26.38	12.30	0.064	0.245
40	32.64	25.58	21.31	0.118	0.359

XM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.32	0.00	0.00	0.003	NA
5	5.58	4.79	0.21	0.008	0.039
10	10.31	9.51	0.53	0.009	0.047
15	14.99	13.78	1.42	0.014	0.081
20	19.33	18.01	2.41	0.016	0.097
25	24.28	22.11	3.76	0.029	0.114
30	27.88	25.39	6.27	0.035	0.152
35	30.86	26.50	12.14	0.067	0.241
40	32.60	24.65	22.70	0.132	0.383

XM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.32	0.00	0.00	0.003	NA
5	5.51	4.79	0.21	0.008	0.039
10	10.36	9.53	0.51	0.009	0.046
15	14.99	13.76	1.44	0.014	0.082
20	19.44	17.96	2.48	0.019	0.100
25	24.26	21.67	4.34	0.035	0.132
30	27.94	25.12	6.65	0.040	0.161
35	30.86	26.40	12.31	0.068	0.244
40	32.54	24.46	22.95	0.135	0.388

Bagging - Consensus filters

BaC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.01	0.00	0.00	0.000	NA
5	3.79	3.78	1.25	0.000	0.242
10	6.75	6.72	3.50	0.000	0.327
15	9.94	9.90	5.64	0.000	0.339
20	12.21	12.15	8.88	0.001	0.391
25	12.79	12.71	14.02	0.001	0.491
30	12.12	11.94	20.44	0.003	0.601
35	11.49	11.14	26.84	0.005	0.681
40	8.60	7.56	35.39	0.017	0.811

BaC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	3.31	3.31	1.74	0.000	0.337
10	6.12	6.11	4.12	0.000	0.388
15	8.36	8.36	7.22	0.000	0.442
20	10.15	10.15	10.90	0.000	0.491
25	10.17	10.15	16.45	0.000	0.593
30	9.10	9.07	22.94	0.000	0.697
35	6.74	6.64	30.30	0.001	0.810
40	4.38	4.15	37.42	0.004	0.896

BaC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	3.08	3.08	1.96	0.000	0.381
10	5.44	5.44	4.80	0.000	0.455
15	7.92	7.92	7.67	0.000	0.472
20	8.79	8.79	12.24	0.000	0.560
25	8.11	8.11	18.32	0.000	0.675
30	6.54	6.54	25.02	0.000	0.781
35	5.39	5.17	31.45	0.003	0.852
40	2.78	2.75	38.24	0.000	0.931

BaC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.00	0.00	0.00	0.000	NA
5	2.81	2.81	2.24	0.000	0.437
10	5.43	5.43	4.81	0.000	0.456
15	6.75	6.75	8.83	0.000	0.550
20	7.65	7.65	13.31	0.000	0.616
25	7.31	7.31	19.02	0.000	0.707
30	6.08	6.08	25.39	0.000	0.797
35	4.58	4.58	31.78	0.000	0.869
40	1.81	1.81	38.83	0.000	0.955

Bagging - Majority Vote filters

BaM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.40	0.00	0.00	0.004	NA
5	5.39	4.64	0.37	0.008	0.070
10	10.36	9.21	0.87	0.013	0.078
15	14.65	13.76	1.43	0.010	0.082
20	18.64	16.86	3.80	0.022	0.155
25	23.64	21.04	5.13	0.035	0.157
30	26.76	23.25	9.11	0.050	0.223
35	28.99	23.14	16.57	0.090	0.338
40	31.60	20.61	28.21	0.183	0.484

BaM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.26	0.00	0.00	0.003	NA
5	5.14	4.81	0.19	0.004	0.036
10	10.19	9.33	0.73	0.010	0.065
15	14.49	13.62	1.59	0.010	0.091
20	18.67	17.49	3.03	0.015	0.124
25	22.93	21.25	4.80	0.022	0.149
30	26.24	23.00	9.38	0.046	0.232
35	28.07	22.50	17.26	0.086	0.356
40	29.56	20.93	26.99	0.144	0.476

BaM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.28	0.00	0.00	0.003	NA
5	5.31	4.83	0.16	0.005	0.031
10	10.08	9.44	0.60	0.007	0.054
15	14.46	13.83	1.34	0.007	0.077
20	18.32	17.61	2.87	0.009	0.118
25	22.86	21.74	4.17	0.015	0.129
30	25.74	23.07	9.22	0.038	0.229
35	27.40	23.54	15.66	0.059	0.326
40	28.75	21.07	26.49	0.128	0.472

BaM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	0.19	0.00	0.00	0.002	NA
5	5.17	4.86	0.13	0.003	0.025
10	9.81	9.36	0.69	0.005	0.063
15	14.43	13.83	1.35	0.007	0.077
20	18.42	17.72	2.74	0.009	0.112
25	22.74	21.49	4.49	0.017	0.139
30	25.58	23.65	8.42	0.028	0.210
35	26.99	23.49	15.65	0.054	0.328
40	28.21	20.71	26.77	0.125	0.481

Boosting filter

Bo(3)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	6.49	4.82	0.18	0.018	0.033
10	16.65	9.58	0.49	0.079	0.040
15	18.24	13.76	1.49	0.053	0.082
20	23.99	18.19	2.36	0.072	0.088
25	33.08	22.32	3.90	0.143	0.106
30	39.65	22.38	13.30	0.247	0.252
35	48.26	24.40	20.03	0.367	0.301
40	53.58	24.88	32.17	0.478	0.377

Bo(5)

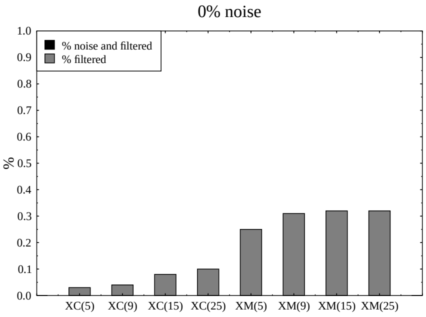
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	8.10	3.81	1.29	0.045	0.237
10	14.69	7.07	3.45	0.085	0.292
15	17.47	10.43	5.52	0.083	0.304
20	26.96	14.18	8.01	0.160	0.289
25	28.07	18.93	8.39	0.122	0.242
30	36.42	20.04	15.62	0.234	0.330
35	41.36	21.12	23.51	0.311	0.395
40	46.86	22.67	32.59	0.403	0.433

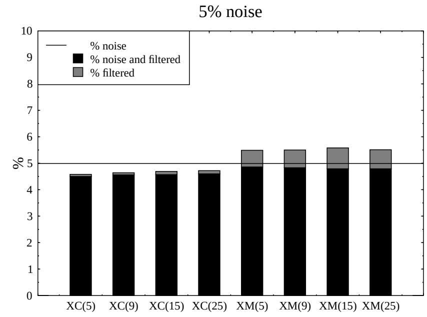
Bo(10)

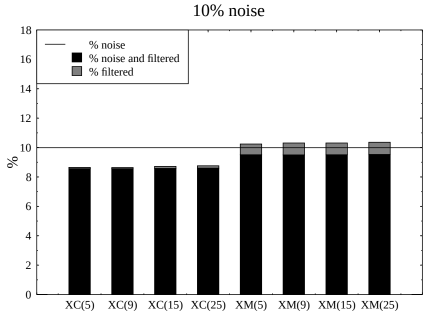
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	5.81	3.43	1.65	0.025	0.312
10	11.38	6.79	3.60	0.051	0.320
15	15.79	10.21	5.68	0.066	0.319
20	21.08	13.40	8.31	0.096	0.328
25	25.92	16.53	11.38	0.125	0.338
30	31.17	18.67	16.37	0.178	0.376
35	35.94	20.36	22.74	0.239	0.417
40	41.15	21.44	31.45	0.328	0.463

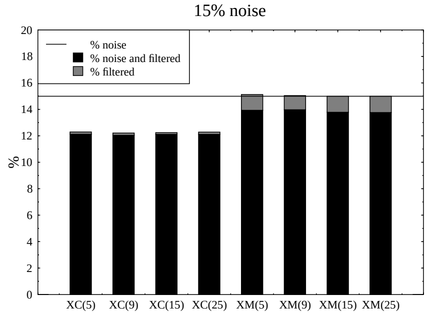
Bo(17)

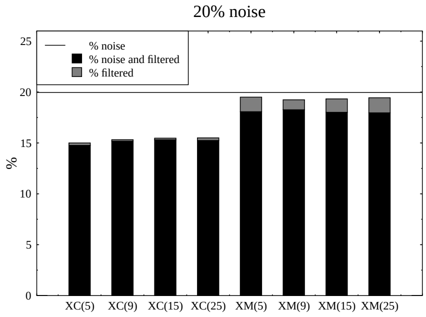
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	5.51	3.36	1.72	0.023	0.326
10	10.75	6.68	3.70	0.045	0.331
15	15.60	10.03	5.87	0.066	0.331
20	21.11	13.26	8.49	0.098	0.335
25	26.00	16.58	11.32	0.125	0.336
30	30.50	18.44	16.53	0.172	0.384
35	36.42	20.32	22.99	0.247	0.418
40	41.24	21.33	31.69	0.331	0.466

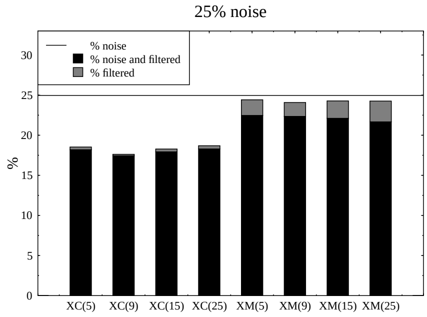

0% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

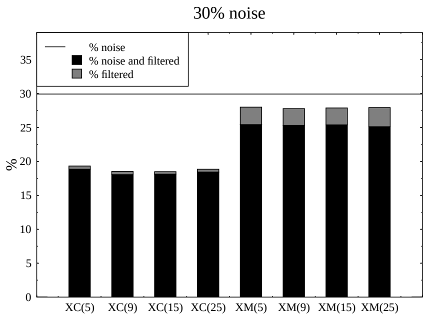

5% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

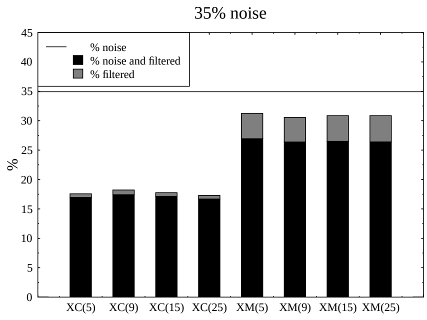

10% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

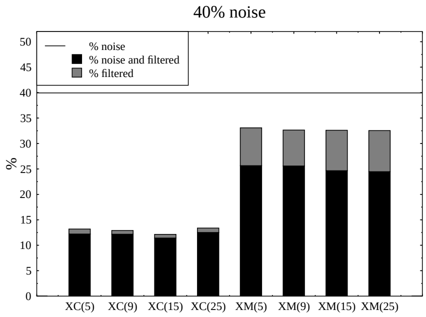

15% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

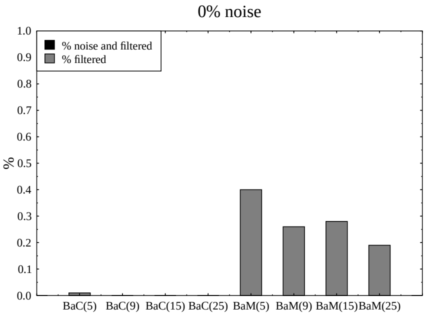

20% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

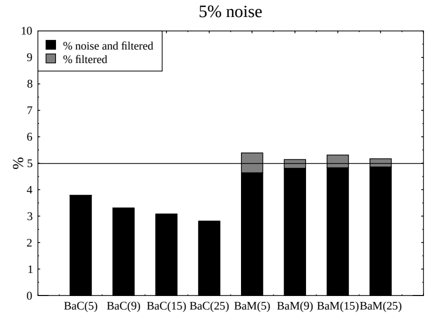

25% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

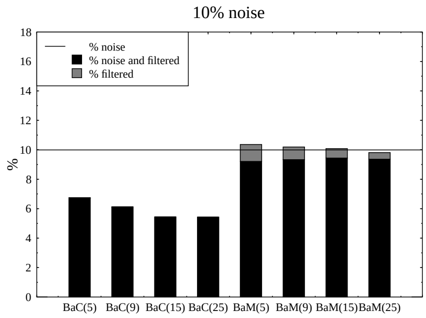

30% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

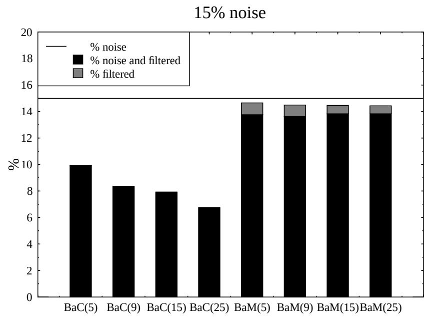

35% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

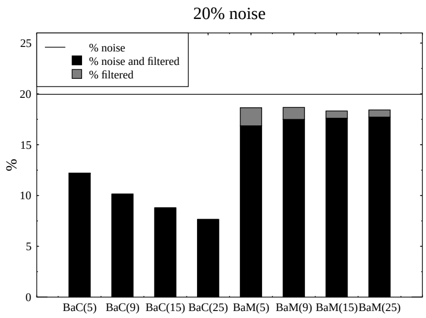

40% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

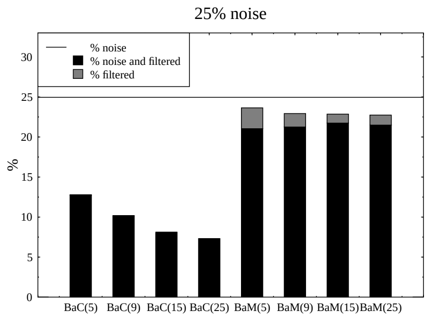

0% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

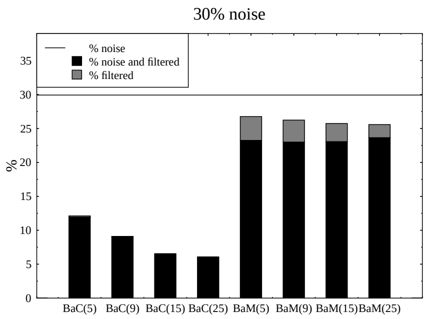

5% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

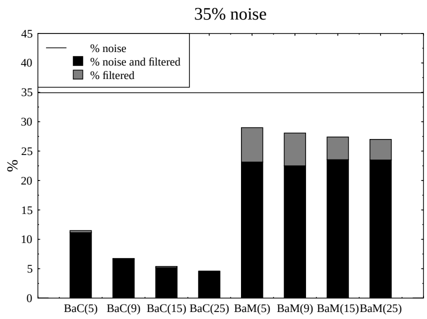

10% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

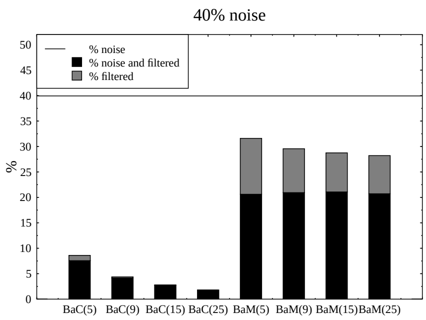

15% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

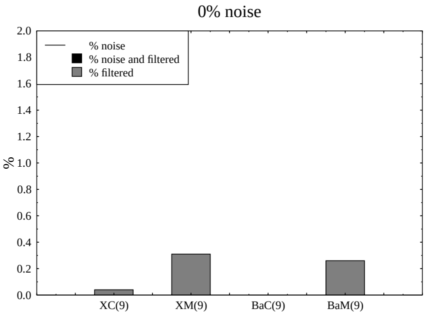

20% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

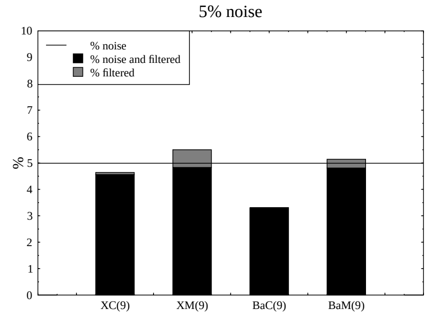

25% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

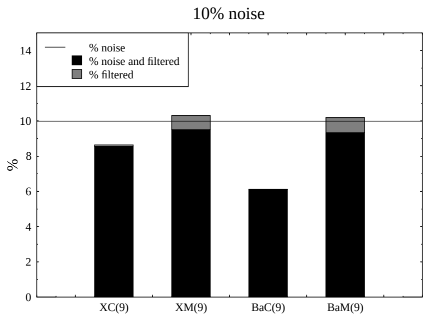

30% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

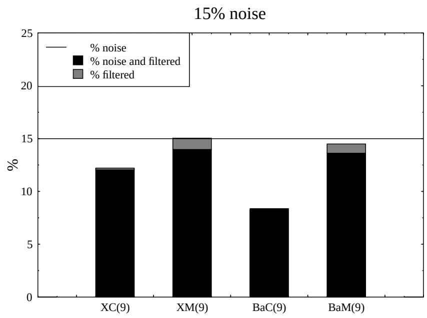

35% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

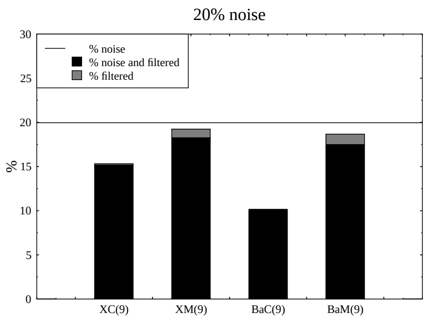

40% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

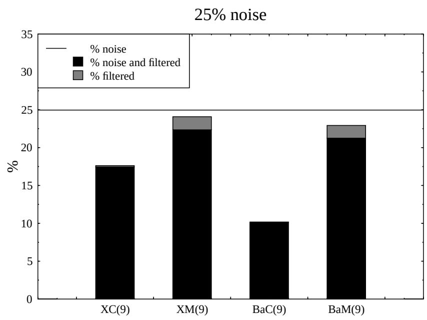

0% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

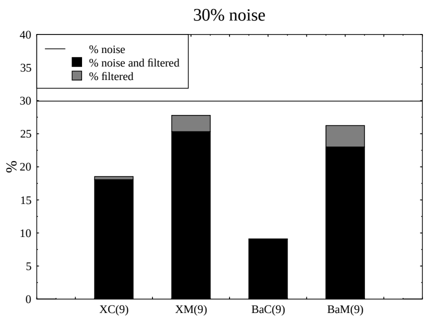

5% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

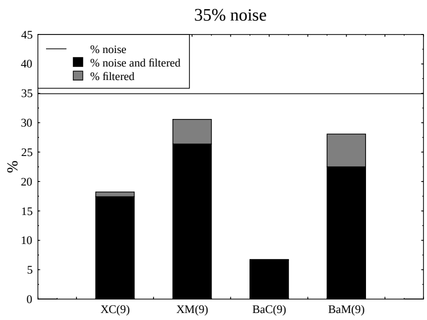

10% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

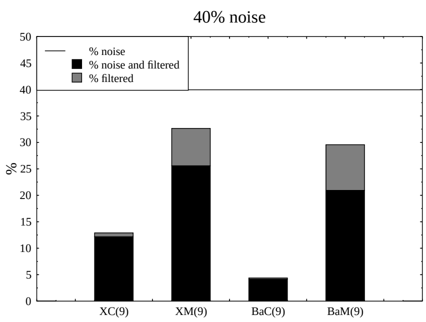

15% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


20% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

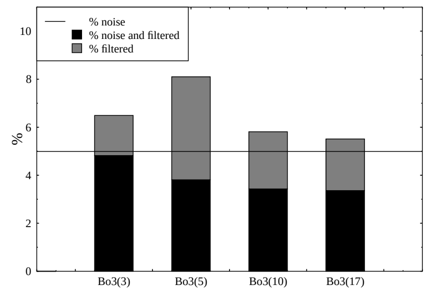

25% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


30% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

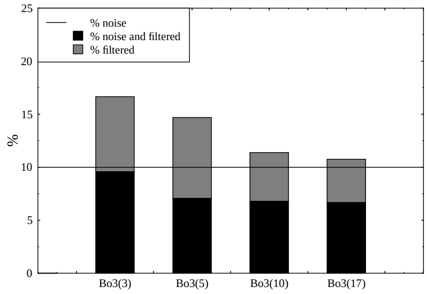

35% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


40% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

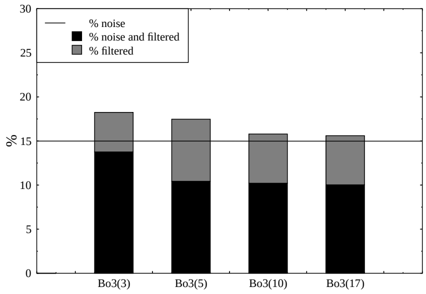
5% noise



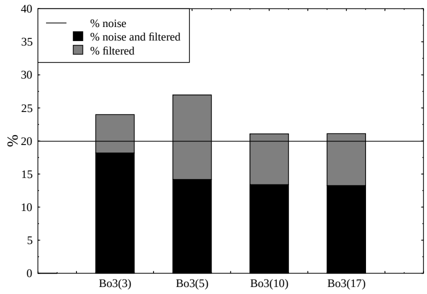
10% noise



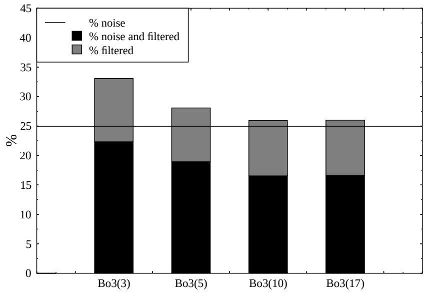
15% noise



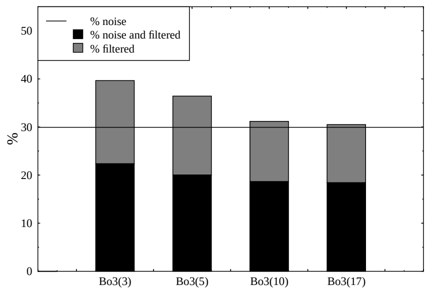
20% noise



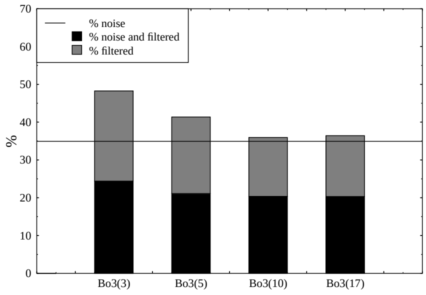
25% noise



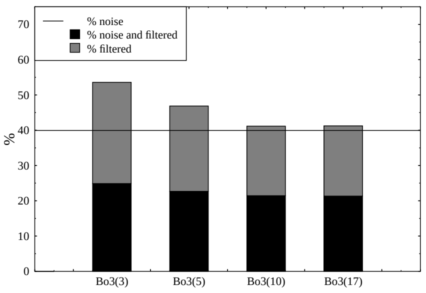
30% noise



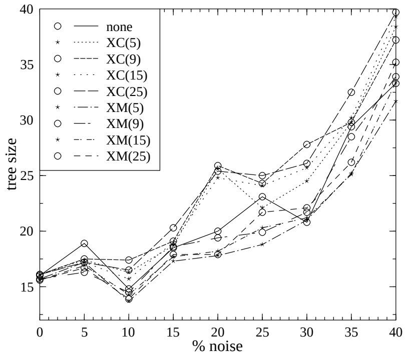
35% noise

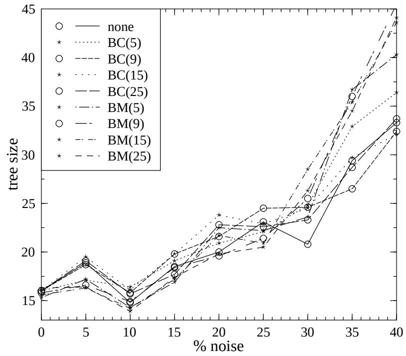


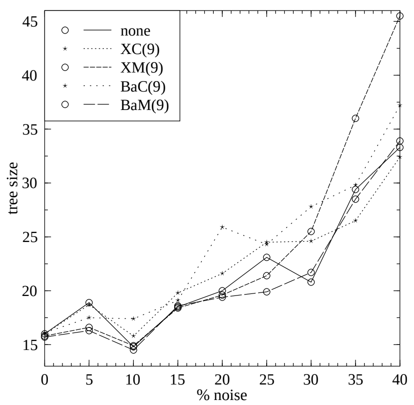
40% noise

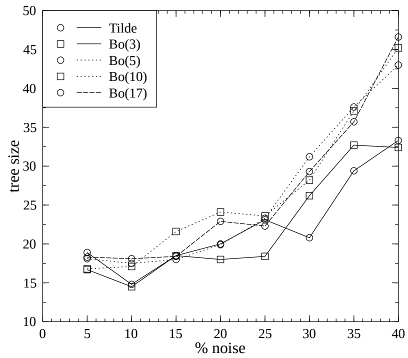


B.3.2 Tree size

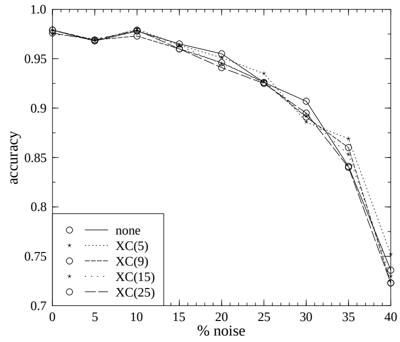


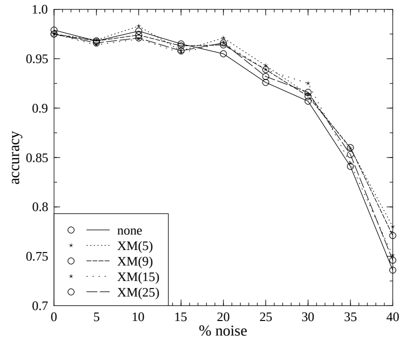


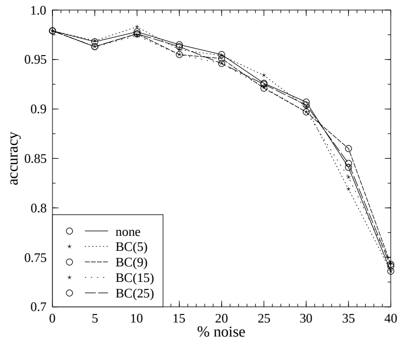


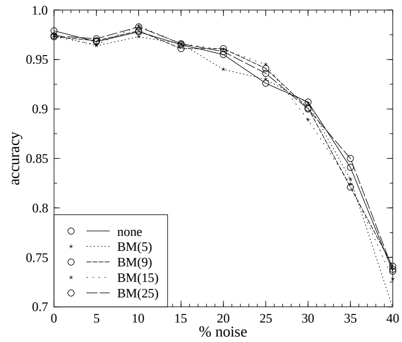


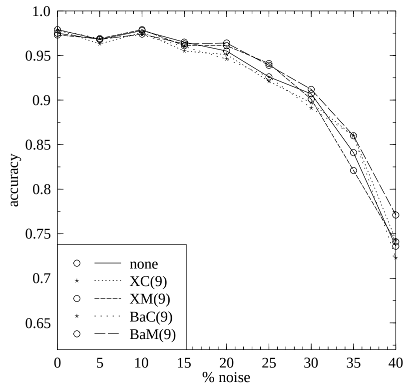
B.3.3 Accuracy

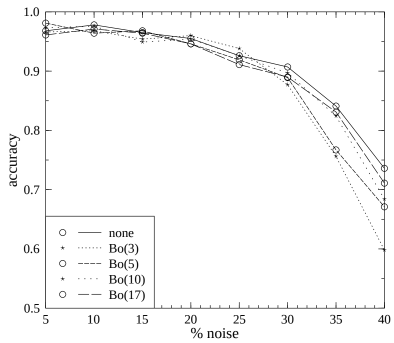




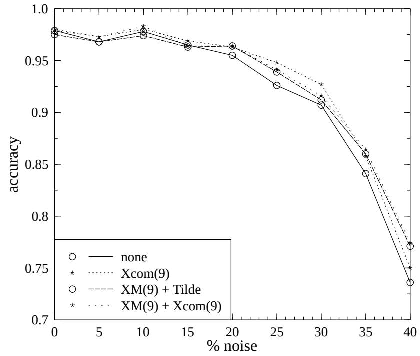


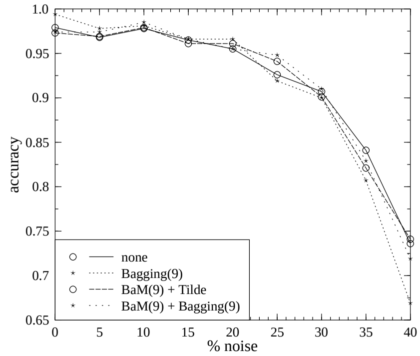






B.3.4 Voting versus Filtering





C Illegal Chess Endgame Positions data set (KRK)

This data set [11] concerns the chess endgame KRK with three pieces left on the chess board: White King, White Rook and Black King. The classification task is to distinguish between illegal and legal board positions. The following information is used to learn the concept of illegal position. The examples are represented by the predicate illegal/6 specifying the column and row coordinates of White King, White Rook and Black King, respectively. The data on ordering and adjacency of rows and columns are provided as a background knowledge using the predicates lt/2 (less_than) and adj/2.

C.1 KRK, 200 examples

C.1.1 Filter Precision

Cross-Validated Committees - Consensus filters

XC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.50	0.00	0.00	0.075	NA
5	11.11	3.89	1.24	0.076	0.222
10	14.50	7.11	3.37	0.082	0.289
15	15.28	8.94	7.11	0.075	0.404
20	18.94	12.44	9.22	0.081	0.378
25	19.11	13.61	13.98	0.073	0.456
30	21.50	16.39	17.27	0.073	0.454
35	20.89	16.94	22.77	0.061	0.516
40	21.39	16.89	29.02	0.075	0.578

XC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.50	0.00	0.00	0.075	NA
5	11.33	3.89	1.24	0.078	0.222
10	14.78	7.56	2.85	0.080	0.244
15	15.28	8.78	7.29	0.076	0.415
20	18.72	12.56	9.09	0.077	0.372
25	19.22	13.56	14.05	0.076	0.458
30	20.44	15.72	17.88	0.067	0.476
35	19.83	16.33	23.25	0.054	0.533
40	21.56	15.83	30.86	0.095	0.604

XC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.50	0.00	0.00	0.075	NA
5	11.33	3.89	1.24	0.078	0.222
10	15.22	7.83	2.54	0.082	0.217
15	15.44	9.17	6.84	0.074	0.389
20	18.56	12.28	9.41	0.078	0.386
25	18.83	13.44	14.17	0.072	0.462
30	20.72	16.00	17.60	0.067	0.467
35	20.56	16.72	22.96	0.059	0.522
40	21.39	16.17	30.36	0.087	0.596

XC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.50	0.00	0.00	0.075	NA
5	11.11	3.89	1.24	0.076	0.222
10	14.72	7.44	2.98	0.081	0.256
15	15.61	9.11	6.94	0.076	0.393
20	18.67	12.39	9.27	0.078	0.381
25	18.83	13.28	14.38	0.074	0.469
30	20.56	15.94	17.65	0.066	0.469
35	20.50	16.83	22.79	0.056	0.519
40	21.39	16.00	30.54	0.090	0.600

Cross-Validated Committees - Majority Vote filters

XM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	8.11	0.00	0.00	0.081	NA
5	12.44	4.06	1.07	0.088	0.189
10	16.39	8.33	1.99	0.090	0.167
15	19.22	11.33	4.51	0.093	0.244
20	22.67	15.39	5.92	0.091	0.231
25	26.83	18.39	8.96	0.113	0.264
30	27.00	19.94	13.68	0.101	0.335
35	28.50	21.94	18.20	0.101	0.373
40	32.22	22.67	25.45	0.159	0.433

XM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.67	0.00	0.00	0.077	NA
5	12.22	4.06	1.06	0.086	0.189
10	16.17	8.39	1.91	0.086	0.161
15	19.11	11.28	4.57	0.092	0.248
20	22.17	15.06	6.30	0.089	0.247
25	25.89	17.78	9.71	0.108	0.289
30	27.28	20.17	13.47	0.102	0.328
35	27.72	20.89	19.36	0.105	0.403
40	29.50	20.83	27.14	0.144	0.479

XM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.50	0.00	0.00	0.075	NA
5	12.28	4.06	1.06	0.087	0.189
10	16.11	8.33	1.98	0.086	0.167
15	18.94	11.06	4.85	0.093	0.263
20	22.44	15.22	6.11	0.090	0.239
25	25.72	17.94	9.43	0.104	0.282
30	27.11	19.78	13.96	0.105	0.341
35	27.72	20.83	19.49	0.106	0.405
40	29.78	21.17	26.80	0.144	0.471

XM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.67	0.00	0.00	0.077	NA
5	12.28	4.06	1.06	0.087	0.189
10	16.11	8.33	1.98	0.086	0.167
15	19.11	11.06	4.85	0.095	0.263
20	22.33	15.06	6.31	0.091	0.247
25	25.72	17.67	9.81	0.107	0.293
30	27.06	19.61	14.17	0.106	0.346
35	27.39	20.67	19.59	0.103	0.410
40	30.22	21.28	26.72	0.149	0.468

Bagging - Consensus filters

BaC(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	6.22	0.00	0.00	0.062	NA
5	9.28	3.22	1.95	0.064	0.356
10	11.78	6.00	4.52	0.064	0.400
15	12.50	6.83	9.31	0.067	0.544
20	15.00	10.33	11.34	0.058	0.483
25	16.22	11.56	16.01	0.062	0.538
30	16.22	12.56	20.57	0.052	0.581
35	18.11	14.89	24.53	0.050	0.575
40	13.67	11.72	32.25	0.032	0.707

BaC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	6.06	0.00	0.00	0.061	NA
5	8.83	3.11	2.06	0.060	0.378
10	11.28	5.44	5.12	0.065	0.456
15	11.00	6.00	10.09	0.059	0.600
20	13.94	9.56	12.09	0.055	0.522
25	15.28	10.83	16.68	0.059	0.567
30	17.44	13.33	20.15	0.059	0.556
35	17.89	14.72	24.67	0.049	0.579
40	13.33	11.39	32.50	0.032	0.715

BaC(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	5.78	0.00	0.00	0.058	NA
5	8.39	2.78	2.42	0.059	0.444
10	10.22	5.00	5.55	0.058	0.500
15	10.44	5.56	10.53	0.058	0.630
20	13.44	9.22	12.43	0.053	0.539
25	15.39	10.89	16.65	0.060	0.564
30	9.67	7.44	24.56	0.032	0.752
35	13.72	11.39	27.04	0.036	0.675
40	11.17	9.44	33.88	0.029	0.764

BaC(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	5.83	0.00	0.00	0.058	NA
5	8.06	2.72	2.47	0.056	0.456
10	10.17	4.89	5.68	0.059	0.511
15	10.33	5.44	10.65	0.058	0.637
20	13.39	9.11	12.55	0.053	0.544
25	15.11	10.67	16.86	0.059	0.573
30	9.67	7.39	24.63	0.033	0.754
35	13.67	11.39	27.02	0.035	0.675
40	3.50	2.94	38.13	0.009	0.926

Bagging - Majority Vote filters

BaM(5)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	8.06	0.00	0.00	0.081	NA
5	12.17	3.94	1.20	0.087	0.211
10	15.89	7.89	2.51	0.089	0.211
15	17.72	10.00	6.05	0.091	0.333
20	21.17	13.67	7.99	0.094	0.317
25	23.72	16.00	11.73	0.103	0.360
30	25.67	19.00	14.76	0.095	0.367
35	26.72	20.11	20.21	0.102	0.425
40	29.22	20.39	27.72	0.147	0.490

BaM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.72	0.00	0.00	0.077	NA
5	12.00	4.17	0.94	0.082	0.167
10	16.28	8.11	2.25	0.091	0.189
15	17.78	10.28	5.70	0.088	0.315
20	21.61	14.39	7.09	0.090	0.281
25	23.28	15.94	11.74	0.098	0.362
30	24.06	17.78	16.04	0.090	0.407
35	24.94	19.50	20.58	0.084	0.443
40	27.06	18.78	29.15	0.138	0.531

BaM(15)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.78	0.00	0.00	0.078	NA
5	12.22	4.17	0.94	0.085	0.167
10	16.11	8.50	1.78	0.085	0.150
15	17.94	10.89	4.99	0.083	0.274
20	20.67	14.11	7.37	0.082	0.294
25	22.61	15.83	11.79	0.090	0.367
30	24.11	18.17	15.51	0.085	0.394
35	24.94	19.56	20.51	0.083	0.441
40	26.06	18.67	28.89	0.123	0.533

BaM(25)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	7.56	0.00	0.00	0.076	NA
5	11.78	4.00	1.12	0.082	0.200
10	15.39	8.00	2.36	0.082	0.200
15	18.17	11.28	4.53	0.081	0.248
20	21.06	14.06	7.47	0.087	0.297
25	22.00	15.39	12.25	0.088	0.384
30	23.44	18.17	15.40	0.075	0.394
35	24.50	19.06	21.04	0.084	0.456
40	25.06	18.00	29.40	0.118	0.550

Boosting filter

Bo(3)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	9.56	3.00	2.19	0.069	0.400
10	15.44	7.83	2.52	0.085	0.217
15	23.33	10.72	6.15	0.148	0.285
20	29.83	12.78	10.76	0.213	0.361
25	50.00	16.56	18.95	0.446	0.338
30	64.61	21.39	23.56	0.617	0.287
35	57.28	19.83	34.77	0.576	0.433
40	61.83	25.39	39.11	0.607	0.365

Bo(5)

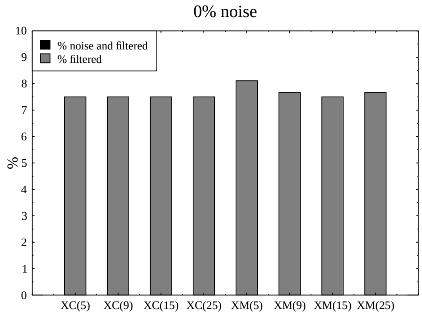
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	8.50	2.72	2.48	0.061	0.456
10	34.28	5.28	7.88	0.322	0.472
15	46.44	6.78	15.91	0.467	0.548
20	49.28	10.67	18.36	0.483	0.467
25	49.28	12.22	25.17	0.494	0.511
30	49.28	15.17	29.28	0.487	0.494
35	49.28	17.28	34.94	0.492	0.506
40	49.17	19.67	39.99	0.492	0.508

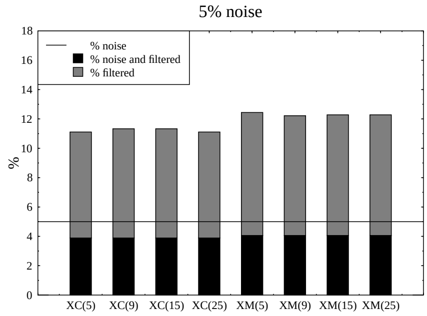
Bo(10)

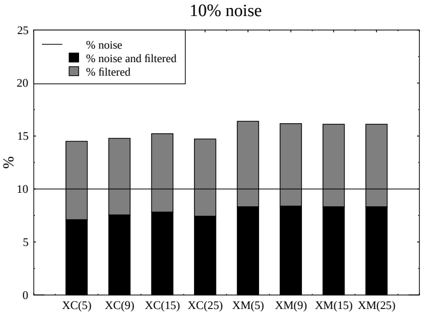
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	8.50	2.67	2.54	0.061	0.467
10	34.22	5.22	7.97	0.322	0.478
15	49.06	6.22	17.25	0.504	0.585
20	49.06	10.67	18.29	0.480	0.467
25	49.06	12.00	25.51	0.494	0.520
30	49.06	15.00	29.48	0.487	0.500
35	49.17	17.22	34.97	0.491	0.508
40	49.06	19.56	40.12	0.492	0.511

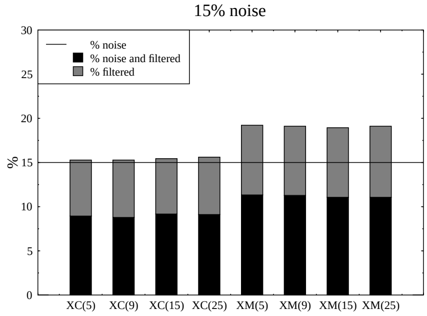
Bo(17)

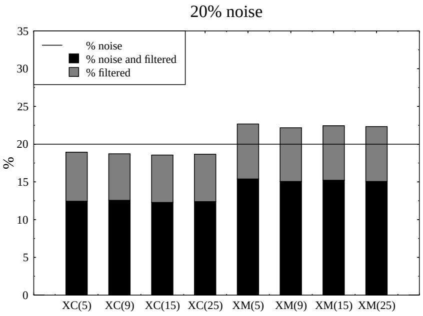
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	8.40	2.64	2.57	0.061	0.472
10	39.93	5.35	8.31	0.384	0.465
15	48.95	5.99	17.68	0.505	0.601
20	48.95	10.49	18.59	0.481	0.475
25	49.06	12.06	25.40	0.493	0.518
30	49.06	15.17	29.15	0.484	0.494
35	49.06	17.28	34.79	0.489	0.506
40	49.06	19.61	40.01	0.491	0.510

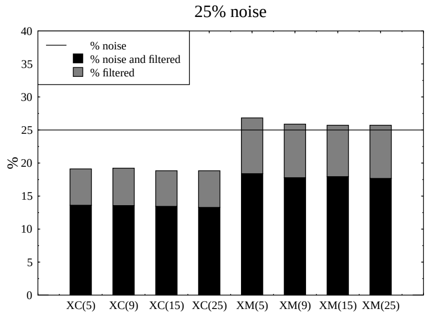

0% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

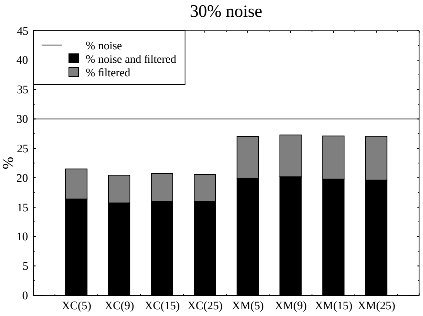

5% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

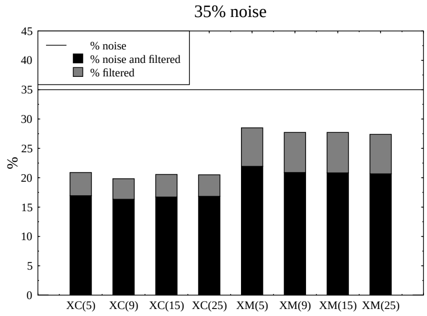

10% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

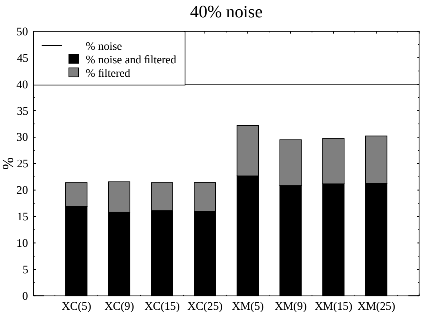

15% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

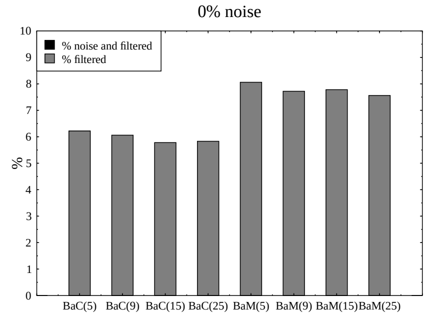

20% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

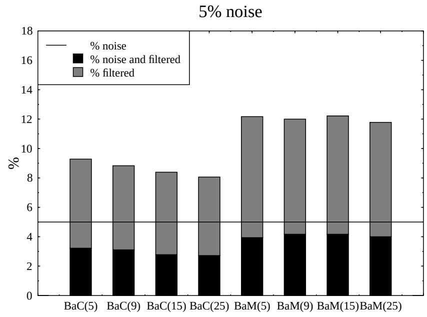

25% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

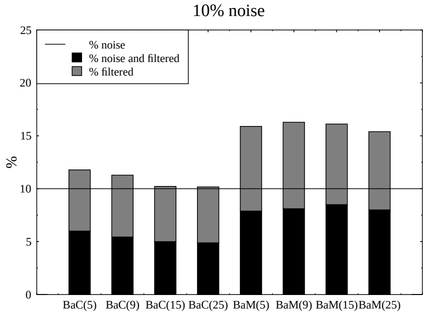

30% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

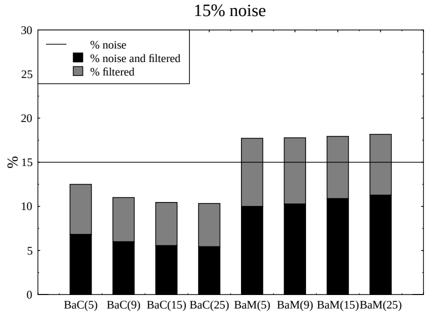

35% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

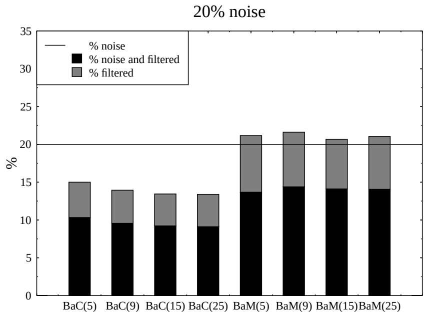

40% noise
% noise
% noise and filtered
% filtered
%
XC(5) XC(9) XC(15) XC(25) XM(5) XM(9) XM(15) XM(25)

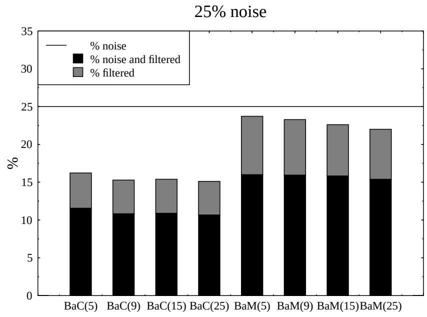

0% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

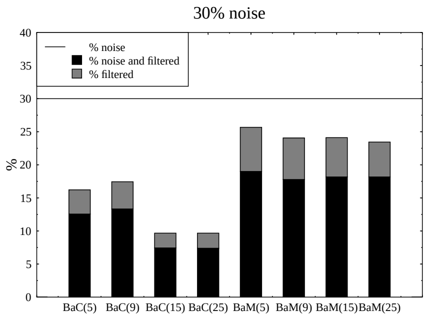

5% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

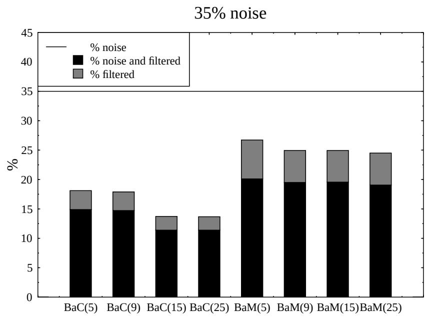

10% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

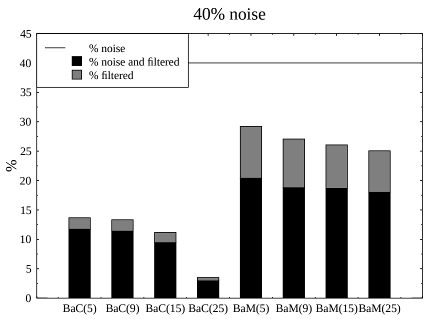

15% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

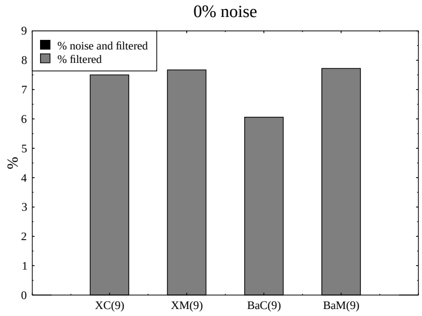

20% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

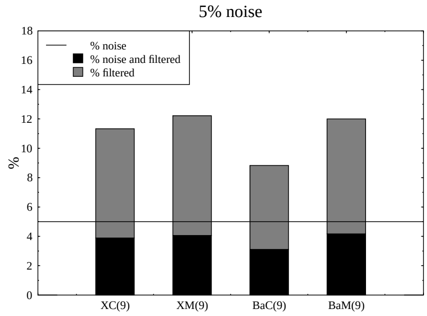

25% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

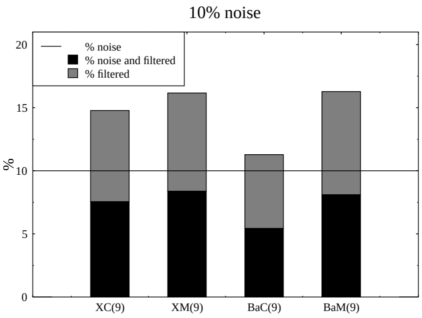

30% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

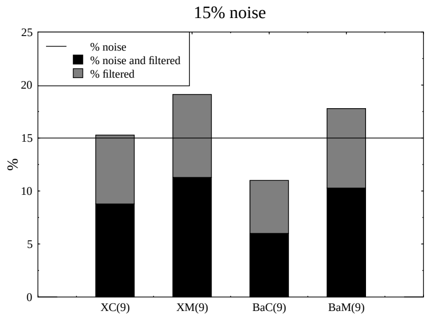

35% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

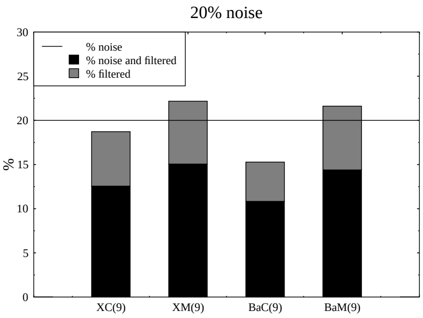

40% noise
% noise
% noise and filtered
% filtered
%
BaC(5) BaC(9) BaC(15) BaC(25) BaM(5) BaM(9) BaM(15) BaM(25)

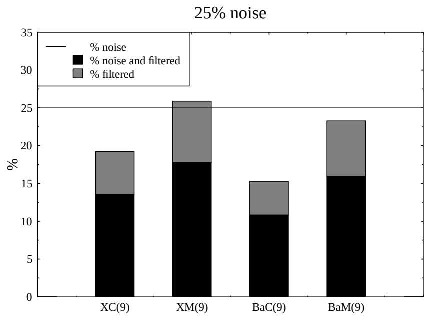

0% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

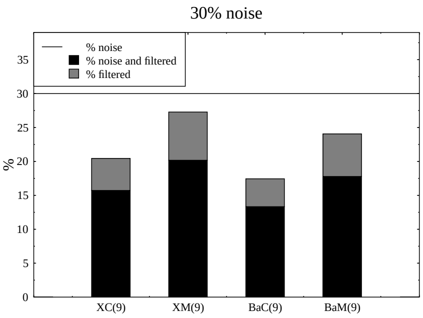

5% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

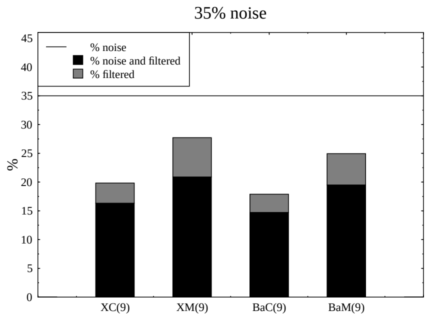

10% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

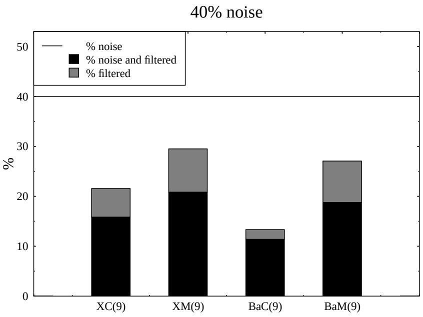

15% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


20% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

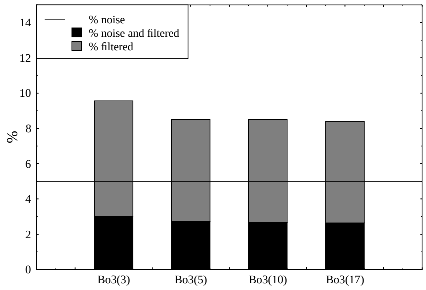

25% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


30% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

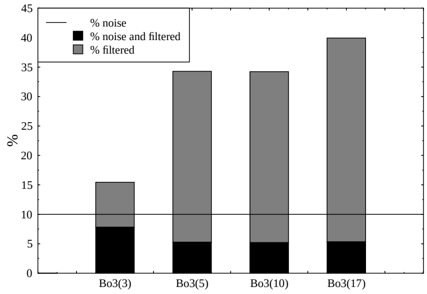

35% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)


40% noise
% noise
% noise and filtered
% filtered
%
XC(9)
XM(9)
BaC(9)
BaM(9)

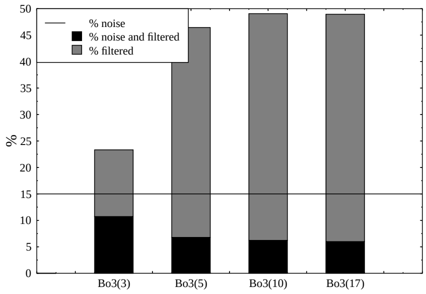
5% noise



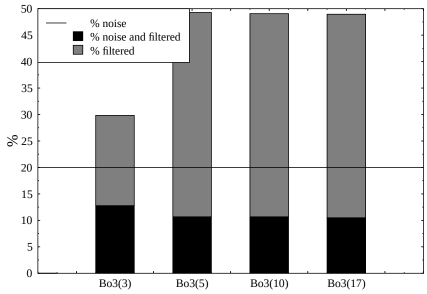
10% noise



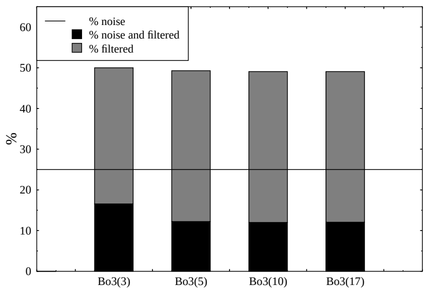
15% noise



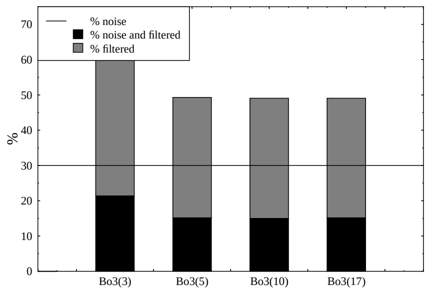
20% noise



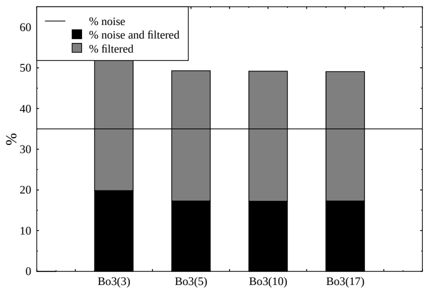
25% noise



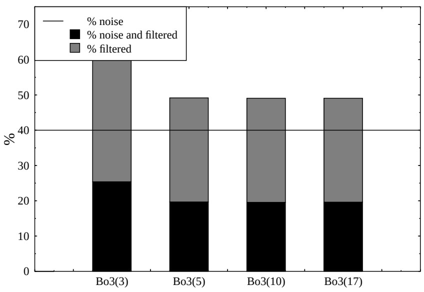
30% noise



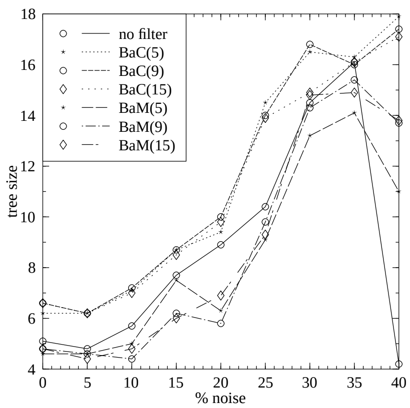
35% noise

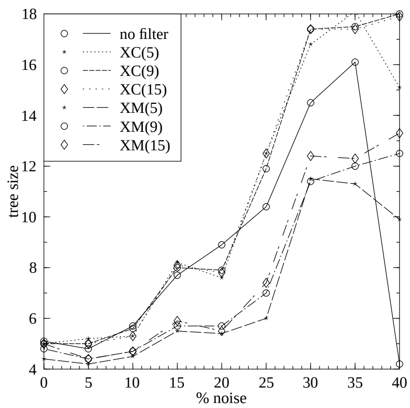


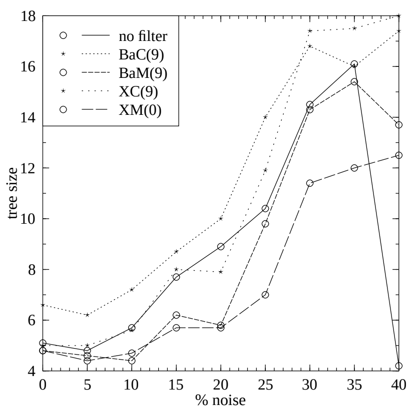
40% noise

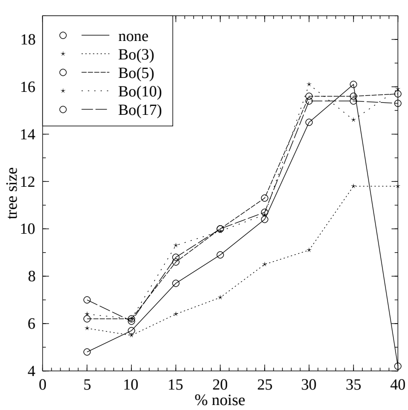


C.1.2 Tree size

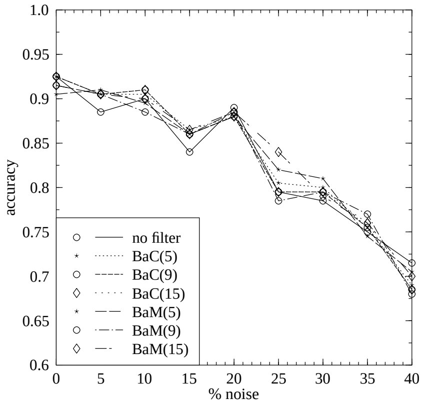


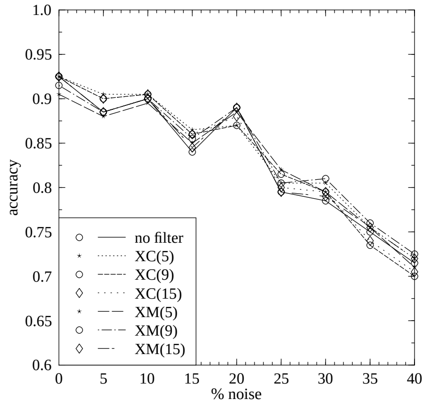


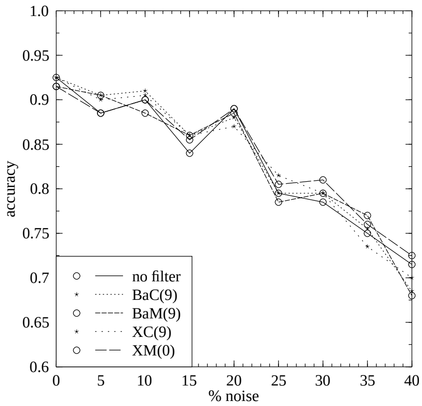


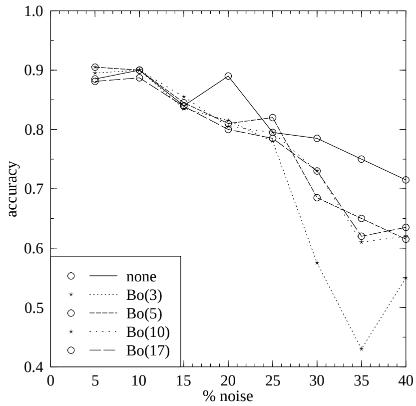


C.1.3 Accuracy

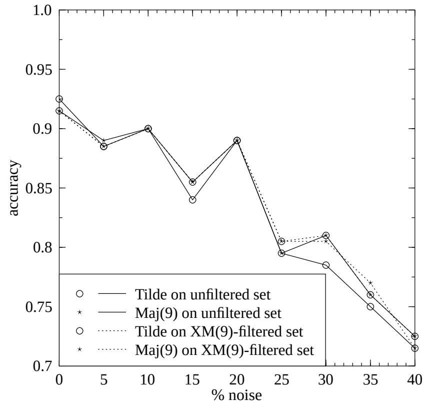


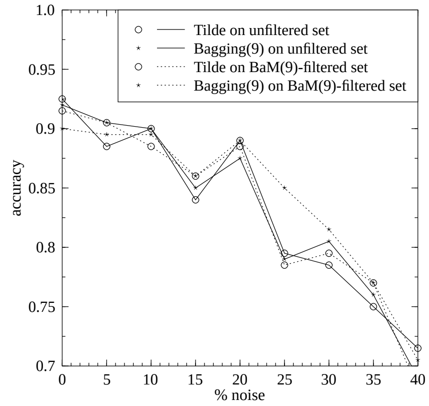






C.1.4 Voting versus Filtering





C.2 KRK, 400 examples

C.2.1 Filter Precision

Cross-Validated Committees - Consensus filters
XC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	3.97	0.00	0.00	0.040	NA
5	8.81	4.44	0.61	0.046	0.111
10	10.33	6.86	3.49	0.039	0.314
15	12.58	9.00	6.85	0.042	0.400
20	15.56	12.36	8.99	0.040	0.382
25	15.97	12.81	14.42	0.042	0.488
30	15.69	13.39	19.68	0.033	0.554
35	18.22	15.72	23.56	0.038	0.551
40	19.78	17.22	28.37	0.043	0.569

Cross-Validated Committees - Majority Vote filters
XM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	4.19	0.00	0.00	0.042	NA
5	9.47	4.64	0.40	0.051	0.072
10	13.31	8.67	1.53	0.052	0.133
15	17.11	11.78	3.87	0.063	0.215
20	21.00	16.06	4.98	0.062	0.197
25	23.14	18.53	8.40	0.061	0.259
30	22.92	18.92	14.35	0.057	0.369
35	25.78	21.22	18.44	0.070	0.394
40	27.67	21.58	25.39	0.101	0.460

Bagging - Consensus filters
BaC(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	3.44	0.00	0.00	0.034	NA
5	5.42	2.17	2.99	0.034	0.567
10	7.53	4.50	5.94	0.034	0.550
15	8.89	6.00	9.87	0.034	0.600
20	11.17	8.50	12.93	0.033	0.575
25	12.33	9.83	17.29	0.033	0.607
30	13.67	11.67	21.23	0.029	0.611
35	16.61	14.36	24.74	0.035	0.590
40	10.72	9.47	33.70	0.021	0.763

Bagging - Majority Vote filters
BaM(9)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
0	4.69	0.00	0.00	0.047	NA
5	8.67	4.25	0.82	0.046	0.150
10	12.33	8.22	2.02	0.046	0.178
15	15.17	11.06	4.64	0.048	0.263
20	18.42	14.58	6.62	0.048	0.271
25	20.00	15.86	11.39	0.055	0.366
30	21.19	17.33	16.05	0.055	0.422
35	23.58	19.06	20.85	0.070	0.456
40	25.72	20.58	26.11	0.086	0.485

Boosting filter

Bo(3)

% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	7.53	3.39	1.72	0.044	0.322
10	16.19	8.14	2.52	0.090	0.186
15	27.94	11.28	5.73	0.196	0.248
20	25.69	15.83	6.01	0.123	0.208
25	46.72	14.58	19.83	0.429	0.417
30	51.72	14.97	28.63	0.525	0.501
35	53.64	19.11	33.28	0.531	0.454
40	51.97	21.44	38.58	0.509	0.464

Bo(5)

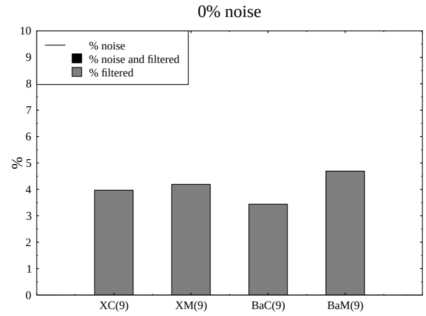
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	10.72	3.14	2.19	0.080	0.372
10	43.53	4.69	9.40	0.431	0.531
15	43.67	6.53	15.04	0.437	0.565
20	43.94	9.08	19.46	0.436	0.546
25	43.72	10.69	25.42	0.440	0.572
30	43.75	13.11	30.02	0.438	0.563
35	43.78	15.58	34.54	0.434	0.555
40	43.69	17.47	40.01	0.437	0.563

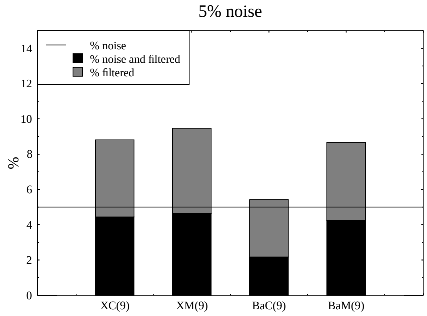
Bo(10)

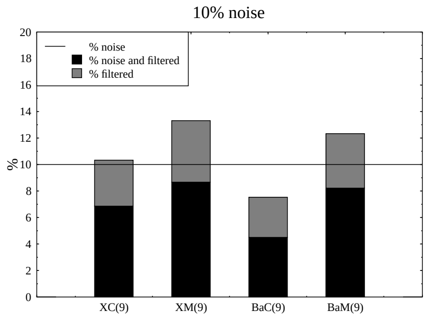
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	22.53	2.25	3.83	0.213	0.549
10	43.53	4.72	9.35	0.431	0.528
15	43.64	6.50	15.08	0.437	0.567
20	43.72	9.06	19.44	0.433	0.547
25	43.53	10.67	25.38	0.438	0.573
30	43.53	13.00	30.10	0.436	0.567
35	43.61	15.50	34.59	0.432	0.557
40	43.61	17.36	40.15	0.438	0.566

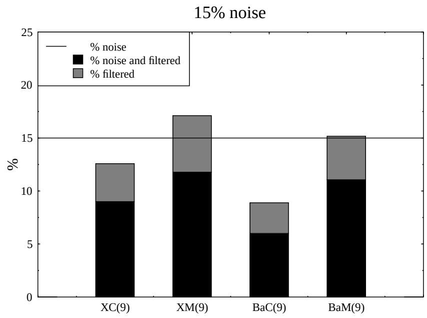
Bo(17)

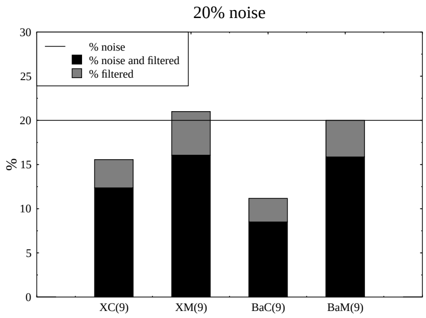
% noise	% filtered	% noise and filtered	% noise in filtered set	$P(E_1)$	$P(E_2)$
5	18.61	2.13	3.92	0.173	0.574
10	43.61	4.40	9.92	0.436	0.560
15	43.64	6.48	15.12	0.437	0.568
20	43.72	9.11	19.34	0.433	0.544
25	43.73	10.60	25.60	0.442	0.576
30	43.65	12.98	30.20	0.438	0.567
35	43.70	15.14	35.29	0.439	0.567
40	43.53	17.14	40.48	0.440	0.571

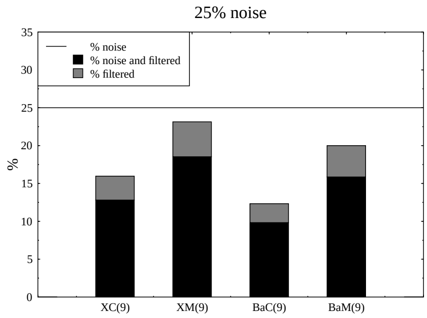

0% noise
% noise
% noise and filtered
% filtered
%
0 1 2 3 4 5 6 7 8 9 10
XC(9) XM(9) BaC(9) BaM(9)

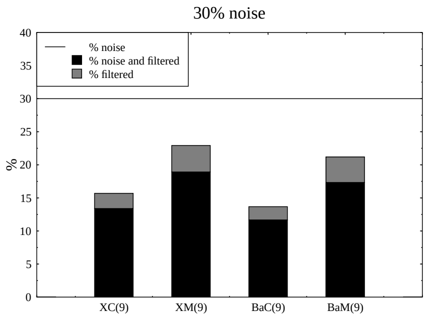

5% noise
% noise
% noise and filtered
% filtered
%
0 2 4 6 8 10 12 14
XC(9) XM(9) BaC(9) BaM(9)

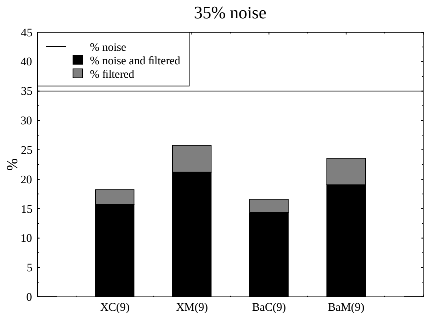

10% noise
% noise
% noise and filtered
% filtered
%
0 2 4 6 8 10 12 14 16 18 20
XC(9) XM(9) BaC(9) BaM(9)

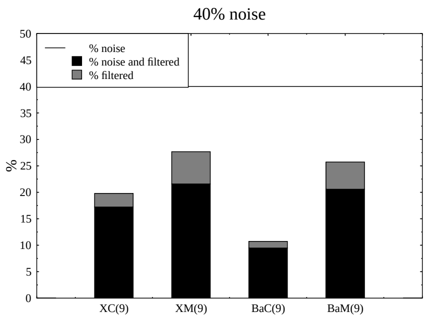

15% noise
% noise
% noise and filtered
% filtered
%
0 5 10 15 20 25
XC(9) XM(9) BaC(9) BaM(9)

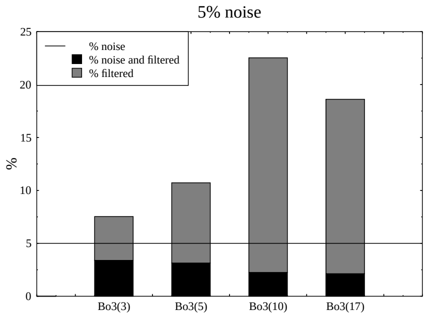

20% noise
% noise
% noise and filtered
% filtered
%
0 5 10 15 20 25 30
XC(9) XM(9) BaC(9) BaM(9)

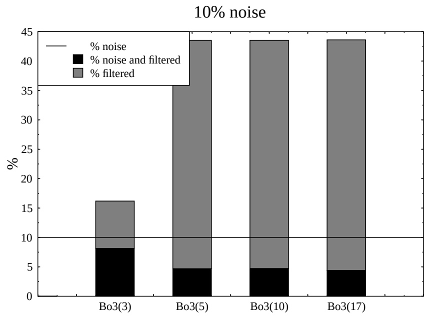

25% noise
% noise
% noise and filtered
% filtered
%
0 5 10 15 20 25 30 35
XC(9) XM(9) BaC(9) BaM(9)

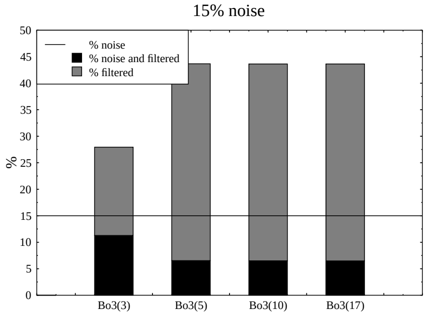

30% noise
% noise
% noise and filtered
% filtered
%
0 5 10 15 20 25 30 35 40
XC(9) XM(9) BaC(9) BaM(9)

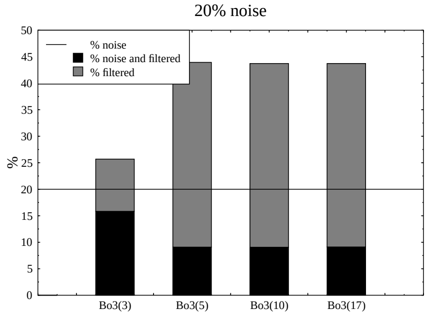

35% noise
% noise
% noise and filtered
% filtered
%
0 5 10 15 20 25 30 35 40 45
XC(9) XM(9) BaC(9) BaM(9)

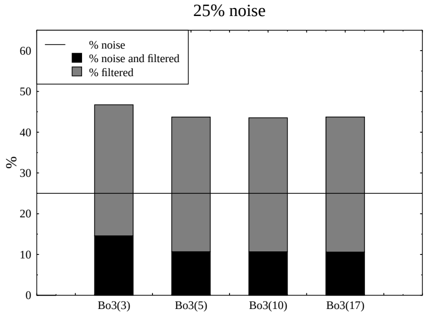

40% noise
% noise
% noise and filtered
% filtered
%
0 5 10 15 20 25 30 35 40 45 50
XC(9) XM(9) BaC(9) BaM(9)

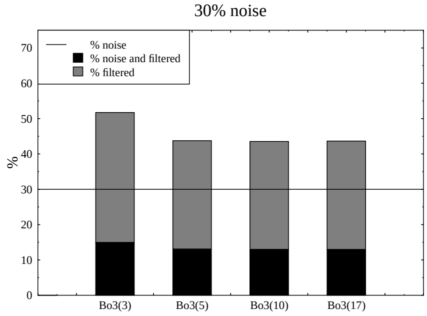

5% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)

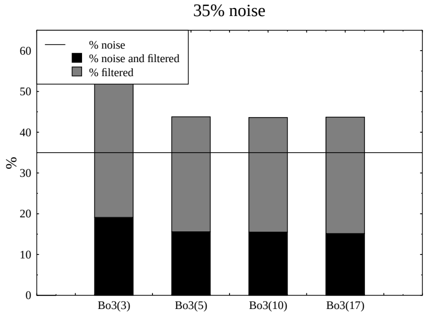

10% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)

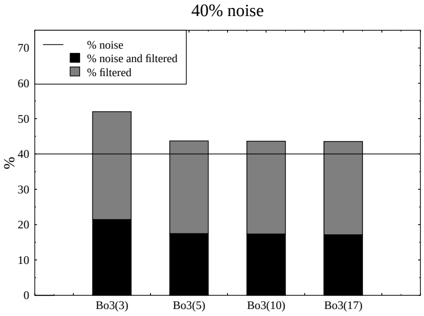

15% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)


20% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)

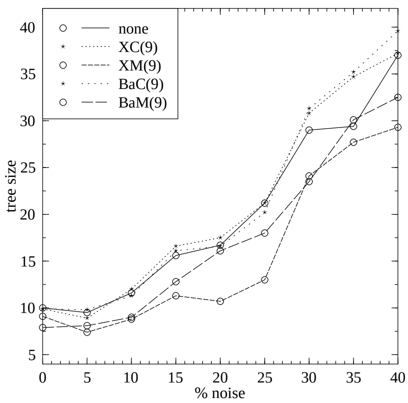

25% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)

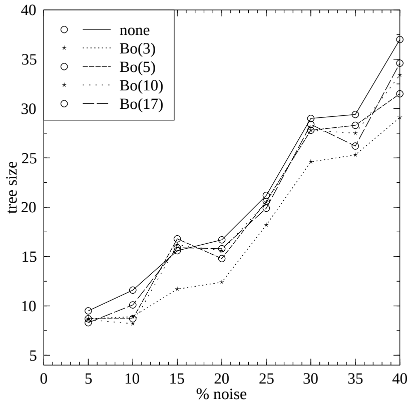

30% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)


35% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)

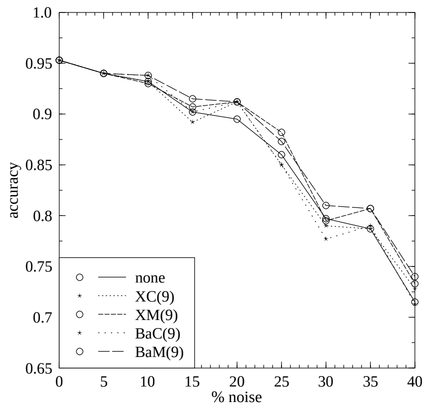

40% noise
% noise
% noise and filtered
% filtered
%
Bo3(3)
Bo3(5)
Bo3(10)
Bo3(17)

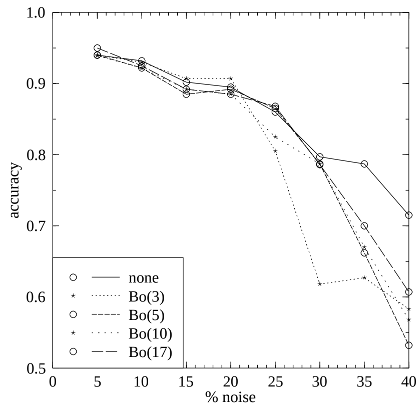
C.2.2 Tree size





C.2.3 Accuracy





C.2.4 Voting versus Filtering