ACTIVE LEARNING IN NEURAL NETWORKS

M. Hasenjäger and **H. Ritter**
Technische Fakultät
Universität Bielefeld
Germany
{pmhasenj,helge}@techfak.uni-bielefeld.de

We discuss a new paradigm for supervised learning that aims at improving the efficiency of neural network training procedures: *active learning*. The starting point for active learning is the observation that the traditional approach of randomly selecting training samples leads to large, highly redundant training sets. This redundancy is not always desirable. Especially if the acquisition of training data is expensive, one is rather interested in small, informative training sets. Such training sets can be obtained if the learner is enabled to select those training data that he or she expects to be most informative. In this case, the learner is no longer a passive recipient of information but takes an active role in the selection of the training data.

In our contribution, we review recent research on active learning. We discuss the main approaches and give experimental results to demonstrate the power of active learning.

1 Introduction

In supervised learning, we are interested in training a student on a set of input–output pairs generated by an unknown target function in such a way that the student does not only remember these samples but is capable of making sensible predictions of the outputs of previously unseen samples as well, i. e. the student should be able to generalize well.

The final generalization ability of the student depends on a number of factors among them the architecture of the student, the training procedure and the training data. In recent years, much research effort has been directed towards the optimization of the learning process with regard to both the learning efficiency and generalization performance. This includes algorithms that aim at the adaptive optimization of the learning architecture, e.g. [1] [2] [3], as well as algorithms that improve existing training procedures, e.g. [4]. It was only recently that advanced techniques for the selection of the training data moved into the focus of interest.

Traditionally, the student is trained with samples that are randomly chosen according to some probability distribution on the input space. A shortcoming of this approach is that the average amount of novel information per sample decreases as learning proceeds. The reason for this is that with growing size of the training set, the student's knowledge about large regions of the input space becomes more and more confident so that additional samples from these regions are basically redundant in so far as they do not contribute considerably to an improvement in generalization ability.

Recently a new paradigm of supervised learning has emerged that tries to remedy this problem: *active learning* or *query learning*. Here the student takes an active role in the selection of the training samples instead of passively receiving some random input stream of data.

The aim of active learning methods is to produce concise and highly informative training sets. This is desirable in two completely different situations. There are learning tasks in which training data abound, e.g. in signal processing or speech processing. In these cases training may become unnecessarily inefficient by the need to process large redundant training sets, so that the selection of concise subsets of training data may reduce training times. The inverse situation is given in learning tasks in which training data are sparse and the resources to obtain data have to be allocated in a well-considered way. This is for example the case if the training data have to be provided by trained human experts, which is both expensive and slow.

The remainder of our contribution is organized as follows: After a short section on learning in general (Sec. 2), we will review recent research on active learning in Sec. 3. We will discuss the main approaches to this problem with an emphasis on algorithms for a kind of active learning that is often called *query-based learning* or *active sampling* and that is applied when the training data are sparse and expensive. Here the student is allowed to select an unlabeled sample and to ask the correct label from the teacher. The aim of this procedure is to improve the generalization ability of the student, or alternatively, to achieve the same level of generalization ability with fewer training samples.

In Secs. 4 and 5, we discuss with "Query by committee" and "Active learning in local models" two query algorithms for the selection of unlabeled training data in classification tasks in greater detail. These algorithms exemplify the two main approaches to active learning, the principled approach that selects a query according to the maximum expected information gain and the heuristic approach that tries to select queries from the current classification boundary of the student. In both cases, we give some experimental results to demonstrate the power of active learning.

In Sec. 6, we conclude with a summary and an outlook on possible directions for future research in active learning.

2 Learning

Let us consider a learning scenario in which the learner or student receives an input or a sample $\mathbf{x}$ from an input space $\mathcal{X}$ and is supposed to map this input to a desired output $\mathbf{y}$ in an output space $\mathcal{Y}$. Thus the student s can be thought of as realizing a function $\mathcal{X} \mapsto \mathcal{Y} : s_{\mathbf{w}}(\mathbf{x}) = \mathbf{y}$. Here $\mathbf{w}$ denotes a set of adaptive internal parameters of the student that are adjusted during the learning process until the student realizes the desired function.

The input space $\mathcal{X}$ is usually assumed to be fixed and known, a common choice being $\mathbb{R}^p$, with the samples $\mathbf{x}$ distributed according to some probability distribution $\mathcal{P}$. The output space $\mathcal{Y}$ can be continuous, e.g. $\mathbb{R}^q$, or discrete, e.g. $\{0,1\}$. In the first case, the learning task is regression, while in the second case it is a classification problem, where the input vectors have to be assigned to a discrete set of classes.

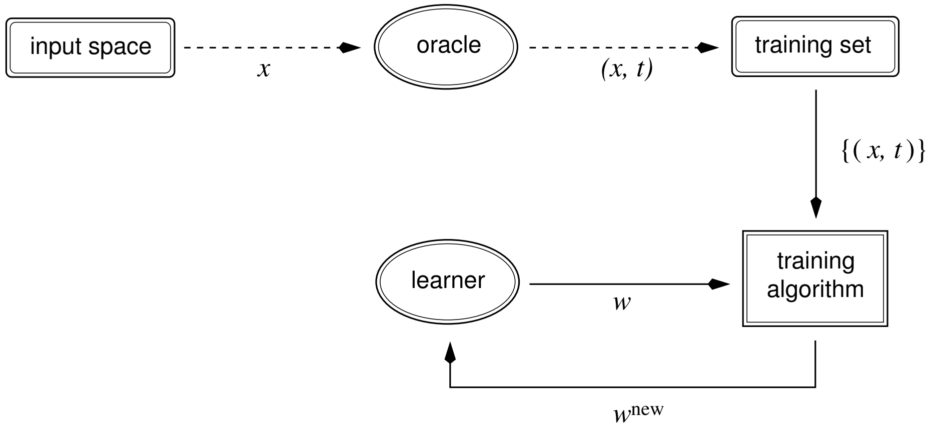


Figure 1. Sketch of passive learning. Samples are generated at random from the input space and labeled by an oracle or teacher (dashed arrows). This training set is used by a training algorithm to iteratively adjust the internal parameters of the student (solid arrows).

In the following, we will mainly be concerned with supervised learning or learning with a teacher. In this learning paradigm, the student is supplied with the desired output value for each input it receives during training, i.e. a training sample $(\mathbf{x}, \mathbf{t}(\mathbf{x}))$ consists of a sample $\mathbf{x}$, which is drawn at random from the probability distribution $\mathcal{P}$ on the input space, along with a label $\mathbf{t}(\mathbf{x})$, where the label is provided by a an oracle or a teacher $\mathbf{t}$ and represents the target output of the student.

The training procedure is then in general formulated in terms of minimization of a suitably chosen error function E that is a function of the adaptive parameters $\mathbf{w}$ of the student. In the case of a binary classification task, for example, where the samples belong to either a class C_0 or a class C_1, this error can be measured by the probability that student and teacher will disagree on a randomly chosen example drawn from the input distribution.

During training, the adaptive parameters $\mathbf{w}$ of the student are iteratively adjusted until the student realizes a good approximation to the teacher function. Note, that the goal of training in general is not to learn an exact representation of the training data but rather to extract a model of the teacher function, i.e. a model of the process that generated the data. This is particularly important if one is interested in the generalization ability of the student, i.e. if the student should be able to make good predictions for new, previously unseen samples.

This standard procedure that is usually applied in supervised learning is sketched in Fig. 1. The crucial point that we want to stress here is that the training samples are chosen *at random*.

3 Active Learning

Randomly chosen training samples raise the following problem that we will illustrate with an example of a binary classification task, as pictured in Fig. 2.

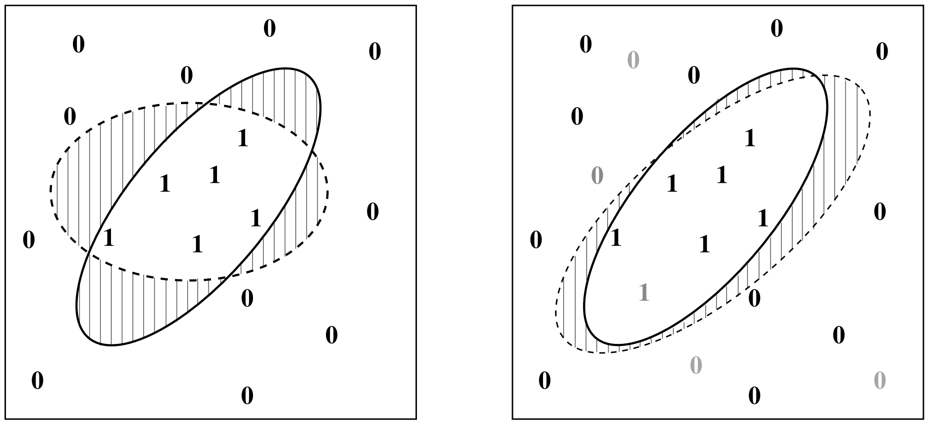


Figure 2. Sketch of the generalization error in a binary classification task. The solid line represents the teacher's classification boundary, while the dashed line is the student's approximation to the teacher. Points in the hatched regions contribute to the generalization error: here student and teacher disagree. The right figure shows how additional samples – marked with gray labels – lead to the reduction of the generalization error.

The task is to assign every point within the solid ellipse to class C_1 and all other points to class C_0. The training samples that are available to the student are marked with their class membership in Fig. 2. The dashed ellipse shows the representation of the classification task a student has acquired after having been trained with these training samples. The student classifies all training samples correctly, but there are still regions of input space where the student and the teacher disagree. Points from these regions, that are hatched in Fig. 2, contribute to the generalization error of the student.

The generalization error of the student, however, can only be reduced if he/she receives additional training samples from these hatched regions. All other training samples will be classified correctly anyway, they do not carry new information and hence are redundant with respect to the minimization of the generalization error.

If additional training samples are selected at random, some of them will be from regions in which student and teacher disagree, cf. Fig. 2 (right), but as learning proceeds and with improving performance of the student, the probability to hit upon an informative sample at random rapidly decreases. So here we have a source of possible inefficiency in the training process. Redundant training samples slow down the training and lead to unnecessarily large training sets. While short training times are always of supreme interest, there are many situations in which the acquisition of labeled training data is expensive, so that small informative training sets are desirable. Just think of a human "oracle", whose working time is expensive and the quality of whose labeling will decrease in the course of time due to fatigue. Another example are situations in which the labeling of a sample requires to perform a costly experiment, as for example is the case in geological test drillings.

To summarize, the problem with randomly selected training samples is that the current state of the student is completely ignored. The training data are not adjusted to the current knowledge of the student. There is no possibility for feedback from the student to the teacher. The student is just a passive recipient of information. The solution to this problem is to provide the student with the ability to interact with the environment, so that the student can take an active part in the learning process.

Active learning can be formalized by extending the elements of the learning process as sketched in Fig. 1 by a new element, the query algorithm. It is the task of the query algorithm to endow the student with the ability to selectively gather new information that matches the student's current knowledge. As shown in Fig. 3, the query algorithm receives the current internal state of the student and produces as output a query that is passed on to the oracle, or the teacher. The oracle answers the query by providing a label with the sample and this new labeled sample is added to the training set. The training with the enlarged training set is carried out then

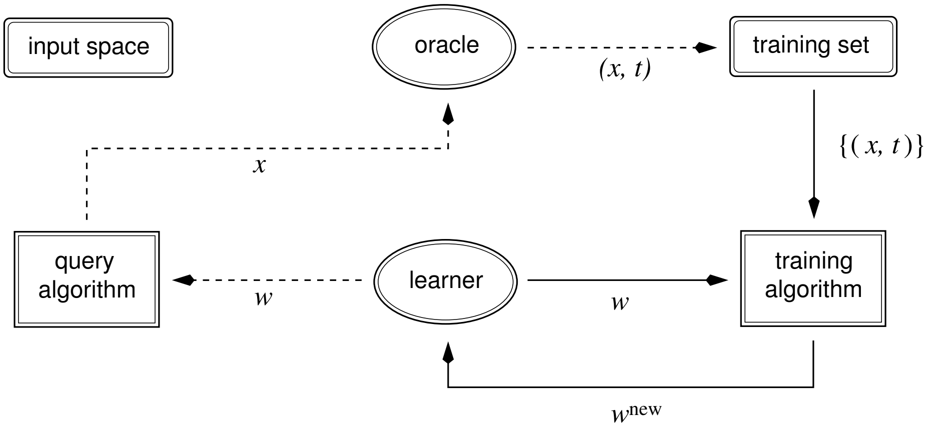


Figure 3. Sketch of active learning. Here the student can influence the selection of training data (dashed arrows) via a query algorithm. The actively selected training set is used by a training algorithm to iteratively adjust the internal parameters of the student (solid arrows).

in the same way as in passive learning.

The goals of active learning can be summarized as follows:

- improve the utility of the training set, i.e. make better use of the information that is available from the current training data with the aim to use less training data than passive learning to achieve the same generalization ability.
- improve the cost efficiency of data acquisition by labeling only those data that are expected to be informative with respect to the improvement of the student's performance.
- facilitate training by removing redundancy from the training set.

3.1 Variations on Active Learning

The basic idea of active learning may be traced back to the statistical theory of optimum experimental design (OED), cf. e.g. [5] [6] [7]. Limited experimental resources and increasing costs in experimental investigations led to the need of extracting an increased quantity of data from processes under study while only finite resources were placed at disposal.

This motivated the development of statistical tools for the design of optimally informative experiments. In the framework we introduced above, the experimental design corresponds to the query algorithm, an experiment is equivalent to a query and the oracle's answer is given by the experimental outcome.

Valiant [8] was the first to formalize the notion of a teacher-student interaction that is not restricted to a passive student in the theory of concept learning. In his framework of probably approximately correct (PAC) learning of Boolean functions, a learning machine consists of a deduction procedure or learning algorithm and of a learning protocol that describes the manner in which information is obtained from the environment. Additionally to a data source that provides labeled samples according to an arbitray probability distribution, he considers an oracle that, when presented with a sample, will provide the correct label. This is the prevalent type of queries, but alternative kinds of queries are possible and have been considered in the context of concept learning [9] [10] [11].

In the previous section, we defined what we mean by *active learning*, namely the ability of the student to select a sample from a set of unlabeled samples and then ask for the correct label. However, there are a number of related approaches that fall into the realm of *active data selection*. In the following section, we give a summary of this work.

3.1.1 Selection of a concise subset from the training set

The problem of selecting a concise subset from the training set is the reverse of the problem discussed in Sec. 3. Here one has to deal with very large and highly redundant training sets, as is typically the case in speech recognition and signal processing. Since in this situation not all training samples are equally important to the learning task, the question is which samples can safely be removed from the training set without loosing too much information?

In this scenario the input as well as the target output are known to the student and he/she has to select a concise subset from the complete training set. The goals of this procedure are to make training more reliable and efficient by selecting a small subset from the complete training data that nevertheless contains enough information to avoid overfitting so that the student still will have a good generalization performance.

There are two basic approaches to be distinguished in this group of algorithms: those that start with a very small subset of the complete training set and sequentially add training samples, i.e. the training set is grown, and those that delete training samples from the complete training set, i.e. the training set is pruned.

Plutowski et al. [12] [13] propose an algorithm that sequentially adds new training samples to the training set. The prerequisite of this algorithm are clean, i.e. noisefree, training data. A new training sample is then added to the training set with the aim to maximize the expected decrement in square error that would result from adding the training sample to the training set and training upon the resulting set.

A related approach is taken by Röbel [14]. In his “Dynamic Pattern Selection Algorithm”, he employs the simple and fast criterion of selecting the training sample with the highest error contribution, while permanently revising the training set with cross validation to ensure valid generalization.

Pruning of training sets can be achieved in a natural way by using Support Vector Machines (SVM) [15], a very flexible type of learner that has become established recently.

In classification tasks, SVMs efficiently construct separating hyperplanes between two classes by maximizing the margin between these classes in some high-dimensioal feature space and then mapping the resulting separating hyperplane in a non-linear fashion back to the input space. It turns out that only a small number of the training samples, the so-called support vectors, suffice to define the separating hyperplane. Thus these support vectors are highly informative training samples. Guyon et al. [16] show how the support vectors can be used to select training samples from the complete training set and to clean a data set from outliers and errors.

Jung and Opper [17] derive a similiar algorithm for the pruning of data sets and the elimination of outliers for linear classifiers within the framework of statistical physics.

3.1.2 Pedagogical pattern selection

A different approach to data selection is called *pedagogical pattern selection*. Here the aim is to find an optimal training sequence while the number of training samples remains fixed. Munro[18] and Cachin [19] make use of the observation that in online learning, where the training samples are presented sequentially and the student updates its parameters after each presentation of a training sample, the performance of the student is sensitive to the sequence of the training samples. Hence they propose several heuristics that present patterns with high training error more frequently to the student than those with low training error. The authors find that depending on the task at hand convergence time and accuracy can be improved by such selection strategies.

3.1.3 Active Exploration in Learning Control

A completely different setting of active learning is given, if the environment is not static. This is the case e.g. in learning control or in game playing. Here the student may take actions that influence the environment, so that the state of the environment is changed and this new state of the environment in turn determines what the student may observe. There are long-time dependencies between action and observation and an observation may depend on many previous observations. For strategies of how to select actions in a directed manner in this context, see [20] [21] and the references therein.

3.1.4 Other related approaches

A different kind of queries is considered by Ratsaby [22]. He allows queries for additional examples from a selected pattern class.

Heskes [23] presents a learning algorithm for perceptron learning from an unreliable teacher that is applicable to learning in noisy and unstationary environments. He shows that the student can always outperform the teacher, if he/she neglects some of the teacher's answers, particularly those the student is confident about.

Other approaches that are related to active learning include the active selection of training samples for leave-one-out cross-validation [24] and a group of algorithms that aim at the detection and removal of irrelevant input variables, cf. e.g. [25].

3.2 Active Learning in Classification Tasks

In this section, we will be concerned with the main focus of this chapter: algorithms for the selection of informative samples for which the student should query the labels from the teacher. The scope of the more detailed discussion will be limited to classification problems. In particular, we will deal with binary classification problems where each sample is assigned to either of two classes C_0 or C_1, as sketched in Fig. 2.

What do we mean by *informative sample* in this context? On a descriptive level, an informative sample is one whose label can not be predicted with certainty, so that the disclosure of the correct class creates a certain degree of surprise. This implies that an informative new training sample should lead to a change in the student's representation of the problem, i.e. the classification boundary of the student should change.

Approaches to active learning in this context can coarsely be divided into two groups: (*i*) heuristic approaches and (*ii*) principled approaches based on the optimization of a suitably chosen objective function.

Another way of categorizing the approaches would be a grouping according to the manner in which the queries are obtained. The queries either are constructed independently from the input distribution of the samples or a filter technique for the selection of queries is used: random samples are generated according to the input distribution and the student decides whether or not to ask for the label.

3.2.1 Heuristic Approaches

The starting point for the heuristically motivated approaches to active learning is the intuition that the label of such a sample should be queried whose correct classification is uncertain. Such a training sample will carry information in any case. Thus the heuristic is to query for the labels of samples that lie on or near the student's current borderline of classification, since the student's classification of these samples is most likely to be error-prone.

Hence the task is to develop algorithms that have a direct access to the student's representation of the classification boundary in the input space and to select as a query a point from the set of points that constitute the classification boundary. This can be accomplished for various types of students:

Perceptrons. It is particularly simple to access the classification boundary of linear discriminant classifiers and one of the first proposals for active learning in the realm of neural networks was an approach for perceptrons by Kinzel and Ruján [26] [27].

A perceptron is defined by

$$s_{\mathbf{w}}(\mathbf{x}) = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x}) \; , \tag{1}$$

where $\mathbf{w}$ denotes the weight vector of the perceptron. Here the classification boundary consists of those points in input space that are perpendicular to the weight vector $\mathbf{w}$ with $\mathbf{w} \cdot \mathbf{x} = 0$. Such points can easily be constructed and used as queries. Kinouchi and Caticha [28] justify this algorithm on a theoretical basis by showing that these queries indeed minimize the objective function of expected reduction of the generalization error.

Multi-Layer Perceptrons. Along the same line of thought, Hwang et al. [29] propose a query selection algorithm for multi-layer perceptrons (MLP). In this case it is not as obvious as for the simple perceptron how to construct a query point on the classification boundary. The basic idea of the algorithm is to make use of the duality between the weights and the input variables in minimizing the mean squared error of the MLP by the standard training algorithm for MLP's, Backpropagation [30]. It is possible to invert the MLP by using the backpropagated error values not to change the weight values of the net – these are kept constant – but to change the input values of the net [31].

In this way, one can perform a gradient descent search in the input space for an input value that gives rise to a particular output value. If we assume that the activation function of the MLP is given by a sigmoidal function with the output values in the range of $[0, 1]$, where a value of zero represents class C_0 and a value of one represents class C_1, then samples that produce an output of 0.5 – lying midway between the two values that represent the classes – can be considered to form the borderline of classification.

Threshold Nets. A different constructive approach is described by Baum [32]. Here two layer threshold nets are considered and queries are used to construct the separating hyperplanes that are defined by the neurons in the hidden layer. The principle for query selection is to find a point on each of these separating hyperplanes by iterated interpolation and to query between two differently classified points. The weights of the output layer are then determined by passive learning.

Local Models. Basically the same idea of selecting a query from the borderline of the actual classification can be realized for a type of student that is completely different from the students discussed above. The query algorithm for local models proposed by Hasenjäger and Ritter [33] draws on the geometrical properties of these models that typically induce a Voronoi tessellation on the input space. The queries are then selected among the Voronoi vertices of this tessellation. This approach will be discussed in detail in Sec. 5.

3.2.2 Optimization Approaches

The second group of approaches treats active learning as an optimization problem. A measure for the utility of a query is derived and used to formulate a query algorithm that selects as a query the input value at which the utility function attains its global optimum or that selects one such value at random if there is more than one global optimum.

Suitable objective functions include

- the expected information gain, i.e. the reduction in the entropy of the posterior distribution of the student's parameters that is induced by the answer to the query,
- the expected reduction in generalization error.

While the first criterion aims at improving the accuracy with which the student's parameters can be determined, the second criterion optimizes the accuracy with which the outcome of future experiments can be predicted, which is the quantity one is usually interested in in learning. Note, that these measures are not equivalent in general. Since the expected generalization error cannot be calculated directly in a simple way there are only a few approaches that make use of this utility measure [34] [28], while the bulk of work concentrates on the expected information gain as a useful alternative.

A general probabilistic framework for query-based learning and its analysis with the tools of statistical physics is presented by Sollich [34], who considers the objective functions of information gain and generalization error. This approach is closely related to optimal Bayesian sequential design as advocated by MacKay [35], but differs from this Bayesian standpoint in a more general view of the relationship between student and teacher: the student need not be unbiased with respect to the teacher, i.e. the student need not represent the correct model of the teacher, an assumption that is made in nearly all Bayesian approaches.

Within a Bayesian framework of learning, MacKay considers several query criteria that are based on the maximization of the expected information gain for interpolation [35] as well as for classification tasks [36]. Within this framework, the informativeness of a query is measured in terms of the student's internal parameters $\mathbf{w}$. A measure of the expected information gain is then given by the expected reduction in entropy of the distribution of the parameters before and after the new training example is received – equivalently, the Kullback-Leibler divergence between these distributions can be considered.

Along the same line of thought, Cohn [37] presents a practical implementation of a query selection strategy that minimizes the expected output variance of a neural network learner and Belue et al. [38] suggest a query selection criterion for multi-layer perceptron classifiers in a multi-class setting.

An approach that is originally rooted in statistical physics but has been analyzed within a PAC like framework is “Query by Committee” [39] [40]. This algorithm makes use of a committee of students to estimate the expected information gain of a query and it can be shown that in this case there exists a relation between the two objective functions maximization of the expected information gain and minimization of the expected generalization error. This approach will be discussed in more detail in Sec. 4.

While “Query by Committee” is intended for classification tasks, Krogh and Vedelsby [41] propose an algorithm in the same spirit for regression tasks.

3.2.3 Pros and Cons

Now having reviewed the two basic approaches to active learning, let us summarize the principal assets and drawbacks of both methods.

An important disadvantage of the heuristic, constructive algorithms is that they cannot utilize information about the input distribution to the student. It is known from learning theory that access to the input distribution indeed is important for the student [42] [43]. In this respect, the constructive active student is at a disadvantage in comparison to the passive student, who receives this distributional information along with the randomly drawn training set.

A related problem, especially in high-dimensional input spaces, is that the constructed samples may be difficult to classify for the teacher – if they are samples from the true classification border – or that they may have no natural meaning at all. The last point is illustrated in a study by Baum and Lang [44] who applied their query algorithm [32] to optical character recognition and found that the questions posed by the algorithm were too difficult to be answered reliably by a human oracle, since many of the queries consisted of unrecognizable superpositions of characters. However, these problems are not insurmountable and application specific solutions can be found.

Another point that may be criticized is that the heuristic algorithms necessarily are model-specific solutions to the problem of detecting points on the classification boundary of the student. For each kind of student a special solution has to be found. On the other hand, we have seen that this is possible for important classes of students.

The indisputable advantage of the heuristic methods is that they are easy to implement and reasonably fast to compute.

The principled approaches are more satisfactory from the theoretical point of view: the optimization criterion for the selection of queries is explicit and this makes the algorithms amenable to analytical treatment on the one hand and on the other hand their strengths and weaknesses are easier to assess than those of heuristic approaches.

For example, most statistical approaches to active learning assume that the student is approximately unbiased – an exception being [45], where a biased student without variance is discussed – an assumption that will rarely apply in practice. So it is clear, that the quality of the queries will depend on the extent to which the assumption of unbiasedness is true.

Often the query criteria resulting from the principled approaches cannot be calculated exactly, although sometimes even analytical solutions are possible [46]. In such cases there is no need for a search in input space. However, these query criteria often are more difficult to implement than their heuristic counterparts and their evaluation possibly may turn out to be a computationally demanding task.

4 Query by Committee

In this section we study the query algorithm proposed by Seung et al. [39] in closer detail. “Query by Committee” (QBC) makes use of a committee of students in order to estimate the expected information gain of a query. A query is selected in such a way that the expected information gain is maximized.

The two basic ideas of QBC are (*i*) to use more than one student and (*ii*) to select a query from a stream of randomly presented unlabeled samples.

These principles were introduced to active learning by Cohn et al. [47]. They propose to query from a region of uncertainty that can be accessed by training a pair of students: a most specific student that is as restrictive as possible in classifying a sample as belonging to class C_0: except for the training samples belonging to C_0, as few other samples as possible are assigned to this class. The second student, however, is trained to assign as many samples as possible to C_0, this is the most general student. The region of uncertainty is defined to consist of those samples on whose classification the two students disagree.

The second important idea is to give the active student access to the input distribution. Unlabeled samples from this distribution are generated at random and the student decides whether or not to ask for the label of a particular sample. Thus the student filters an input stream of samples for the informative ones.

The major advantages of QBC are on the one hand its foundation on principles of information theory and on the other hand its generality and conceptual simplicity: the principle of QBC does not depend on the internal structure or architecture of the student and the algorithm is applicable to a wide variety of learning models.

We will review previous work on this query algorithm and use a toy example, the high-low game, to show the advantages of active learning and to discuss the algorithm. Our main focus will be on the following questions:

- How does active learning compare with passive learning?
- What role does the training algorithm play?
- How many members should the committee comprise?

4.1 The query algorithm

QBC was proposed [39] for active learning of binary classification tasks. The students should be parametric learning models with continuously varying weights, that are uniformly distributed in weight space prior to

Algorithm 1 Query by Committee

❶ Set up a committee of $2k$ students.

❷ Find an input vector from the set of possible input vectors that is classified as an example of class C_0 by k members of the committee, and as an example of class C_1 by the other k members. In other words, find the input vector that leads to maximum disagreement among the members of the committee.

❸ Ask for the correct classification of this sample and add the new labeled sample to the training set.

❹ Retrain the members of the committee on the new enlarged training set and proceed with ❷.

learning. A stochastic training procedure is considered: the Gibbs training algorithms selects a weight vector $\mathbf{w}$ *at random* from the version space, where the version space is defined to be the subset of weight space that is consistent with the training set. Students with weight vectors from the version space correctly classify the complete training set.

We consider an incremental active learning procedure: after a new training sample was queried, the student is retrained on the enlarged training set before the next query is selected.

The strategy of the QBC algorithm is then simply to train $2k$ students on the same training set and to select as query an input $\mathbf{x}$ that is classified as belonging to class C_0 by half of the committee and as belonging to class C_1 by the other half of the committee, cf. Algorithm 1. The existence of such a query cannot be guaranteed. In such cases, step ❷ of Algorithm 1 must be replaced by a sufficiently weakened condition.

The information content of a query. Let us explain why this strategy of maximizing the disagreement among the committee will lead to queries with a high information gain.

As already mentioned in Sec. 3.2.2, within the statistical framework of learning the information gain from the label of a new query can be measured in terms of the distribution of the student's parameters $\mathbf{w}$ in weight space. Let $P_m(\mathbf{w})$ denote the probability distribution on the weight space that is induced by training the student with a training set $\mathcal{T}_m$ with m training samples $\mathcal{T}_m = \{(\mathbf{x}_i, \mathbf{t}_i), i = 1 \ldots m\}$. A new training sample $(\mathbf{x}_{m+1}, \mathbf{t}_{m+1})$ adds an additional constraint on the weight space, which results in a new distribution $P_{m+1}(\mathbf{w})$. The information I_{m+1} that the student gains from the new training sample is then quantified by the reduction ΔS of the entropy of the parameter distribution on the weight space prior and posterior to the training on the enlarged training set,

$$I_{m+1} = -\Delta S = -(S_{m+1} - S_m) \ , \tag{2}$$

where S_m denotes the entropy of $P_m(\mathbf{w})$,

$$S_m = -\int d\mathbf{w} \, P_m(\mathbf{w}) \log P_m(\mathbf{w}) \ . \tag{3}$$

If additional training samples contain new information then the probability distribution on the weight space becomes narrower and narrower. That means, the volume of version space, i.e. the subset of weight space that is consistent with the training set, shrinks with each informative sample.

In our case of vanishing final training error and if we assume the prior distribution on the weight space to be flat, then the posterior distribution is uniform on the version space and vanishes outside [48]. This means that the distribution $P_m(\mathbf{w})$ on the weight space can be related to the volume V_m of the version space $\mathcal{W}_m$,

$$P_m(\mathbf{w}) = \begin{cases} 1/V_m & \text{if } \mathbf{w} \in \mathcal{W}_m, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In terms of the volume of version space, the information gain from the answer to the query $\mathbf{x}_{m+1}$ is then, cf. Eq. (2),

$$I_{m+1} = -\log\left(\frac{V_{m+1}}{V_m}\right) \ . \tag{5}$$

In query selection, the aim is to maximize the expected information gain of a query, i.e. I_{m+1} in Eq. (5) has to be averaged over all teacher weight vectors that are consistent with the training set.

In our case of binary classification, every new input $\mathbf{x}_{m+1}$ divides the version space $\mathcal{W}_m$ in two parts: a part $\mathcal{W}_0$ of volume V_0 that would label $\mathbf{x}_{m+1}$ as a member of class C_0 and a part $\mathcal{W}_1$ of volume V_1 that would label $\mathbf{x}_{m+1}$ as a member of class C_1. Performing the average over the possible teachers yields the expected information gain $E[I_{m+1}(\mathbf{x}_{m+1})]$ from the query $\mathbf{x}_{m+1}$,

$$E[I_{m+1}(x_{m+1})] = -\frac{V_0}{V_m} \log \frac{V_0}{V_m} - \frac{V_1}{V_m} \log \frac{V_1}{V_m} \ . \tag{6}$$

Clearly, the expected information gain attains its maximum value if $\mathbf{x}_{m+1}$ is chosen such that $V_0 = V_1$. In other words, a query with maximum expected information gain will divide the version space in two parts of equal size.

In general, however, the geometry of version space is quite complex, so that such a query can only rarely be found analytically. On the other hand, we have the opportunity to employ the student to inspect version space. Each student that is consistent with the training set represents a single point in version space. Since we need to take into account the whole version space, we try to evenly distribute a set of students in version space and approximate the necessary volumes V_0 and V_1 by taking a vote from this committee of students on each candidate for a query. Here we have the justification for Algorithm 1: since the committee acts as a sample of version space, an approximate bipartition can be achieved by finding a query that splits the committee in two groups of equal size and thus leads to maximum disagreement among the committee.

Note, however, that in general queries that maximize the expected information gain do not guarantee a rapid decrease of the generalization error. Freund et al. [40] suggest a variant of QBC that is restricted to a committee of two members for which they can establish a relation between the expected information gain and the expected generalization error. This algorithm is sketched in Algorithm 2.

In contrast to most other query algorithms, this variant of QBC comes with a termination criterion that is satisfied if a large number of consecutive samples are not found to be informative, i.e. if their labels are not queried from the teacher. It can be shown [40] that Algorithm 2 guarantees a fast decrease in the generalization error of QBC if the information

Algorithm 2 Query by Committee for a Committee of 2 Members

```
initializations
    choose the maximal tolerable prediction error $\epsilon > 0$
    choose the desired reliability $\delta > 0$
    $n \leftarrow 0$    where $n$ is the number of calls to the oracle
    set up a committee of 2 students
repeat
    ❶ draw a random unlabeled sample $\mathbf{x}$ from the input distribution
    if
        the two students agree in their predictions for the label of $\mathbf{x}$
    then
        ❷ reject the sample and go to ❶
    else
        ❸ ask for the label of $\mathbf{x}$
        ❹ $n \leftarrow n + 1$
        ❺ retrain the students with the new enlarged training set
until
    more than    $t_n = \frac{1}{\epsilon} \ln \frac{\pi^2 (n+1)^2}{3\delta}$    consecutive examples are rejected
end
```

gain of a query can be bounded from below. In particular this result applies to the class of linear discriminant classifiers.

4.2 The high-low game

In this section, we will illustrate QBC for a simple 1D toy problem, the high-low game. This problem is analytically tractable so that the performance of QBC can be compared to the theoretically predicted performance.

The task in the high-low game is to learn a binary threshold function in the unit interval, as illustrated in Fig. 4 (left),

$$t(x) = \begin{cases} 1 & \text{if } x > r, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

In active learning, the task is to learn the high-low game using as few

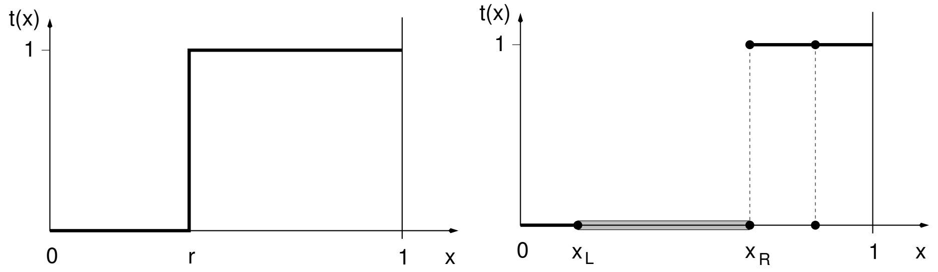


Figure 4. High-low game. The left figure shows the teacher function. The right figure shows the high-low game after three queries. The version space is given by the shaded interval $[x_L, x_R]$.

queries as possible. If the queries are selected according to the criterion of maximum information gain, we can easily derive an optimum query strategy.

Assume that the distribution on the weight space is uniform and that there are already m training samples (x_i, t_i), $i = 1 \dots m$ in the training set. Denote by x_L the largest input value from the training set that received the label $t(x) = 0$ and denote by x_R the smallest input value with $t(x) = 1$, cf. Fig. 4 (right). The version space is then given by the interval $[x_L, x_R]$. If we assume a uniform distribution of the teachers in the version space, we can now specify the probability $P_1(x)$ that a query x is answered with $t(x) = 1$,

$$P_1(x) = \begin{cases} 0 & \text{if } x \leq x_L, \\ \frac{x - x_L}{x_R - x_L} & \text{if } x_L < x < x_R, \\ 1 & \text{if } x \geq x_R. \end{cases} \tag{8}$$

With probability $P_0(x) = 1 - P_1(x)$ a query x will be answered with $t(x) = 0$. Maximization of the expected information gain then requires $P_0(x) = P_1(x) = 0.5$ so that the optimum query $x^\star$ is given by, cf. Eq. (8),

$$x^\star = \frac{x_R + x_L}{2}\,. \tag{9}$$

Thus the optimum strategy is well in accord with intuition: it always seeks to bisect the remaining interval $[x_L, x_R]$.

In the remaining part of this section, we will study empirically whether QBC will indeed pursue the same strategy with its queries. In particular, we will focus on two important parameters of the query algorithm. That is on the one hand the training algorithm that is used together with QBC and on the other hand the size of the committee.

We choose the students to be of the same functional form as the teacher, cf. Eq. (7), and denote the student's parameter with w. We consider the following training algorithms,

- **error correction rule**

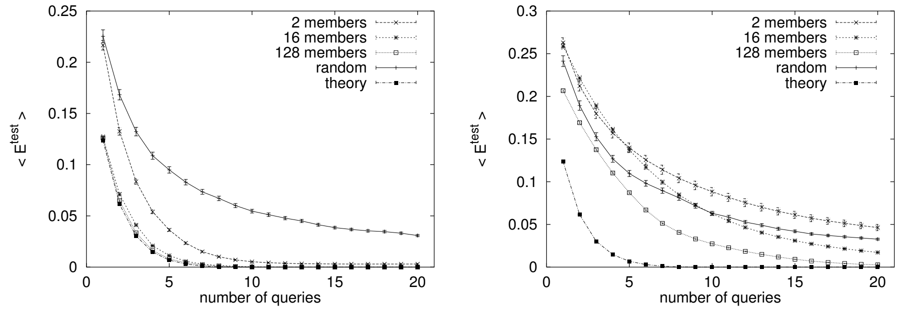
  $$\Delta w = -\eta[t(x) - s(x)]x \ , \tag{10}$$

  where t denotes the teacher function, Eq. (7), s denotes the student function and η is the learning rate parameter. This is the traditional type of learning rule in supervised learning that iteratively adjusts the student's parameter with the aim to reduce the training error. In our experiments the learning rate is $\eta = 0.1$. Training is stopped when the training error vanishes.

- **Gibbs rule**
  The Gibbs rule selects a parameter for the student at random from the version space. We realize the Gibbs rule by randomly drawing a parameter value w from the interval [0,1]. If w is consistent with the training set, i.e. a student with this parameter correctly classifies the complete training set, then w is accepted as a new weight value otherwise w is rejected and another weight value is randomly generated until a parameter w is found that is accepted.

We use a passive student, whose training samples are selected at random from the input distribution and active students, that select the training samples using QBC as described in Algorithm 1, where we use committees with 2, 16 and 128 members. The query algorithm chooses the query from the set $\{x|x = 0.001\,i\,,\ i = 0\ldots1000\}$. This set is used for test purposes as well. The test error E^{test} is calculated according to

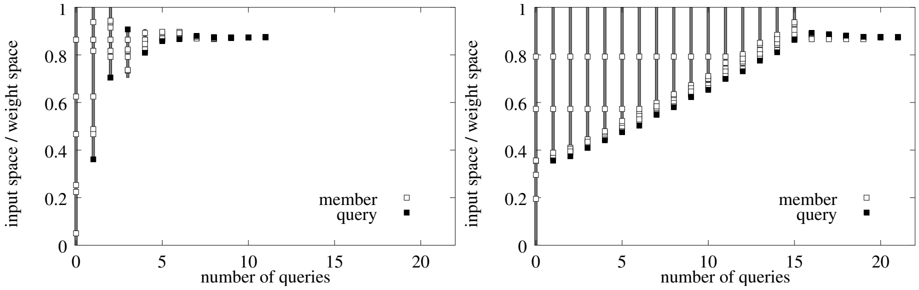
$$E^{\text{test}} = \frac{1}{M}\sum_{i=1}^{M}(t(x_i) - s(x_i))^2 \tag{11}$$



(a) Gibbs learner

(b) error correction learner

Figure 5. Average test error of active learning with QBC as a function of the number of queries for various sizes of the committee and two learning rules, the Gibbs rule (a) and the error correction rule (b). The error bars denote standard deviations about the mean.



(a) Gibbs learner

(b) error correction learner

Figure 6. Evolution of the version space during active learning with QBC for both the Gibbs learner (a) and the error correction learner (b). Both figures show the distribution of a committee of six learners (hollow squares) in the version space along with the actively selected queries (filled squares). The version space is shaded in gray. In this example, the threshold value was $r = 0.875$, which was identified after only 11 queries by the Gibbs learner, while the error correction learner needed 21 queries for the same task.

where M denotes the number of test samples. For the calculation of the test and training errors, a "representative opinion" of the committee is needed. This is established as a majority vote of the committee members. If this vote is undecided, each of the alternatives is chosen with probability 0.5.

The results of these experiments are shown in Figs. 5 and 6. The curves in Fig. 5 are averages of the results from 30 runs of the simulation for each of 30 different teacher functions.

Let us first discuss the results for the Gibbs committees. A comparison of the test performance of the passive student, the curve denoted with ‘random’ in Fig. 5 (a), with the active students – curves denoted with ‘2 members’, ‘16 members’, ‘128 members’ – shows a dramatic improvement by the use of active learning over the passive student. While randomly selected samples lead to a decay of the generalization error that is roughly proportional to the reciprocal value of the number of queries, the generalization error of the active students decays exponentially in the number of queries.

To study the effects of the size of the committee, we performed the learning experiments for a small committee of two members, a medium sized committee of 16 members and a large committee of 128 members. Additionally we trained a student with the theoretical optimum sequence of queries that are selected according to Eq. (9). The results of these simulations are shown in Fig. 5 (a). Already the small committee of size two leads to a clear improvement in generalization error over randomly chosen queries. With an increasing number of committee members, the theoretically optimum decay in generalization error is well approximated. This result exactly comes up to the expectations from theoretical considerations: since the committee represents a sample from version space that is used to estimate the probabilities for an unlabeled input to belong to one or the other class, it is not surprising that the quality of this estimation improves with increasing sample size.

In Fig. 6 (a), the evolution of the version space is plotted during an active learning session of QBC and a committee of six Gibbs students. The Gibbs learning algorithm ensures a distribution of the students (shown as hollow squares) that is almost uniform in version space. The queries (filled squares) show that the committee pursues a query strategy that is close to the optimum strategy of bisecting the version space.

The results turn out quite differently, if we do not use the stochastic Gibbs algorithm for training but the standard error correction rule, Eq. (10), as can be seen from Fig. 5 (b). In this case, the active student with a committee of two performs even worse than the student trained with random samples. It needs a large committee of 128 members to outperform the passive student, and even then the generalization performance of the active student is far from being optimal. The reason for this is illustrated in Fig. 6 (b). Here the version space is depicted during an active learning session of QBC with a committee of six students that are trained with the error correction rule Eq. (10). The error correction learning rule leads to an accumulation of the students near the boundary of version space. This is clear from the deterministic nature of the rule. The size of an adaptation step is limited by the product of the learning rate and the input value of an incorrectly classified sample, so that a student can only advance into version space to a limited extent. The consequences of this clustering of students for QBC are destructive: the committee's estimation of P_0 and P_1 is grossly wrong which leads to very suboptimal queries.

4.3 Summary

In this section, we discussed the "Query by committee" algorithm [39] for active learning in binary classification tasks. The algorithm aims at selecting queries according to the principle of maximizing the expected information gain that is induced by the knowledge of the answer. The estimation of the expected information gain is based on the simple principle of maximizing the disagreement in a committee of students. This is a very general principle that makes active learning amenable to a wide variety of learning architectures.

We used a simple toy problem to demonstrate the algorithm. For the high-low game active learning with QBC leads to a decay in generalization error that is exponential in the number of queries compared to a decay that is proportional to the reciprocal value of the number of queries in passive learning.

The most important ingredient for the success of the procedure is the use of the Gibbs learning procedure, that selects a parameter vector at random from the version space. This is the crucial point where the complexity of the procedure is hidden. It is necessary to train the students in such a way that they are well distributed in version space. This requirement is in general not satisfied by gradient-descent based learning rules, as we have seen in our little example. The principle of Gibbs learning which is a stochastic learning rule that is closely related to simulated annealing [49], however, often is difficult to implement. A coarse approximation to this learning rule can be obtained by using a deterministic learning rule, e.g. Backpropagation [30] in the case of multi-layer perceptrons, to find a point in the version space and then performing a random walk to distribute the students properly. For the case of perceptron students, Bachrach et al. [50] devise an efficient method to implement the Gibbs learning algorithm.

QBC was proposed for a noise-free relation between teacher and student, i.e. the labels that the student receives are assumed to be correct. They are not corrupted by noise and the teacher is assumed to be deterministic, i.e. a particular input is always labeled with the same class. But empirically the algorithm was shown to work successfully in noisy and probabilistic real world tasks, as for active learning in Hidden Markov Model classifiers that were trained for tagging words in natural language sentences [51], [52] and for text categorization where it was shown to reduce the number of labeled training samples by 1-2 orders of magnitude [53].

5 Active Learning with Local Models

In this section, we will present a heuristic approach to active learning in more detail. Remember that the reasoning in the heuristic approaches is to select queries whose answers are completely uncertain, because then the information obtained by the answer will be maximal. In classification tasks, the student should therefore select queries predominantly in the vicinity of its current representation of the classification border. These approaches always provide solutions that are specific to the type of student under consideration. Most heuristic algorithms for active learning were proposed for students that can be classified as ‘global’ models of the teacher function, such as perceptrons [26] [27], multi-layer perceptrons [29], and threshold nets [32]. In the remainder of this section, we will focus on another class of students: local models.

These students consist of a number of subunits that compete for predominance in a region of influence. Each unit of the student is competent only in a locally restricted region of the input space and only contributes to the student’s output if the input falls within its domain. We consider the simplest case of such a local model – a nearest neighbor classifier – and discuss a query strategy that is based on the geometrical properties of this model [33].

5.1 Nearest neighbor classification and Voronoi diagrams

A nearest neighbor classifier consists of a set of distinguished vectors in the input space, the reference vectors. Each point in the input space is assigned to the nearest neighbor among the reference vectors and shares the known class membership of the reference vector to which it is assigned.

Our aim is to construct a suitable set of reference vectors incrementally. We employ an active strategy for the selection of additional reference vectors by making use of the knowledge the student has accumulated so far. To this end let us take a look at the geometry of the classification boundary that is induced by the nearest neighbor classifier.

Since each point in the input space is assigned to its nearest neighbor among the N reference vectors $\mathbf{r}_i$, $i = 1 \ldots N$ the input space is divided into N regions that are defined by $V(\mathbf{r}_i) = \{\mathbf{x} | d(\mathbf{r}_i, \mathbf{x}) \leq d(\mathbf{r}_j, \mathbf{x}); \forall i \neq j\}$, where $d(\cdot, \cdot)$ denotes a distance measure. If d is the Euclidean distance, the regions are called Voronoi polyhedra. The set of points that cannot be assigned uniquely to one of the reference vectors forms the Voronoi diagram of the set of reference vectors.

Figure 7 (left) shows a Voronoi diagram of a set of 12 reference vectors in the two dimensional case. As can be seen, the Voronoi diagram contains another class of distinguished points, namely the points of intersection of three – or, in a p-dimensional space $(p + 1)$ – faces of adjacent Voronoi polyhedra. These points are called *Voronoi vertices* $\mathbf{v}_i$. Since they are equidistant from at least $(p + 1)$ reference vectors, they constitute the centers of circumcircles $C(\mathbf{v}_i)$ that pass through those reference vectors, cf. Fig. 7 (right).

The query algorithm makes use of these geometrical properties of the Voronoi diagram in the following way. Look at the nearest neighbor classifier from the perspective of competitive learners: the reference vec-

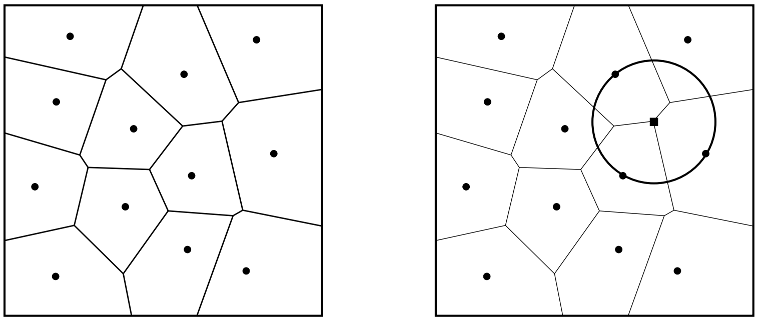


Figure 7. Voronoi diagram of a set of 12 reference vectors. The right figure shows the Voronoi diagram that is induced by the reference vectors. The left figure emphasizes the definition of a Voronoi vertex as the center of the circle that is specified by the vertex' next neighbors among the reference vectors.

tors are competing for regions of influence or competence; each reference vector is "responsible" for its Voronoi polyhedron. Now there are points that are especially "controversial" among the reference vectors: the Voronoi vertices of the set of reference vectors. To put this in terms of a classification problem: the classification of the Voronoi vertices is especially ambiguous, since they are at the same distance from a maximum number of reference vectors. This makes them promising candidates for the next query about the correct classification.

The choice of additional reference vectors among the Voronoi vertices restricts the number of possible candidates considerably but requires a further criterion for selecting a unique candidate. We consider adopting one of the following three strategies as a selection criterion:

A. The new query is selected randomly from the set of Voronoi vertices. Here we assume that the property of being a Voronoi vertex is a sufficient condition for the selection. Further distinguishing features do not matter (all Voronoi vertices are chosen with the same probability).

B. The Voronoi vertex $\mathbf{v}_i$ which is the center of the largest circumcircle $C(\mathbf{v}_i)$ is selected as the next query. This query strategy favors the exploration of the input space.

Strategies A. and B. exploit only information about the geometry of the set of input values for which the student has asked queries so far. The third strategy C. also utilizes the information that has been accumulated about the associated classifications.

C. Only the subset of Voronoi vertices whose nearest neighbors among the reference vectors belong to different classes are considered. These Voronoi vertices lie on the current classification border as defined by the NN-rule. Among these, the one is selected that is the center $\mathbf{v}_i$ of the largest circumcircle $C(\mathbf{v}_i)$. Such a query is guaranteed to yield information on the shape of the classification border, while still furthering the exploration of input space.

5.2 The Two Spirals problem

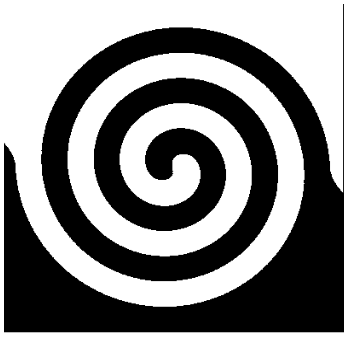


Figure 8. Two-spirals problem: decision regions.

We test the performance of this query algorithm on a well-known benchmark problem, the two-spiral classification problem [54] with the decision regions as indicated in Fig. 8. The performance of the vector quantizer is measured on a test set that consists of 10,000 i.i.d. samples from the input space.

To rank the performance of these different learning strategies, we compare the results of the actively acquired reference vectors to those of randomly selected reference vectors and an approximation of an “optimally” chosen set of reference vectors that is created as explained below.

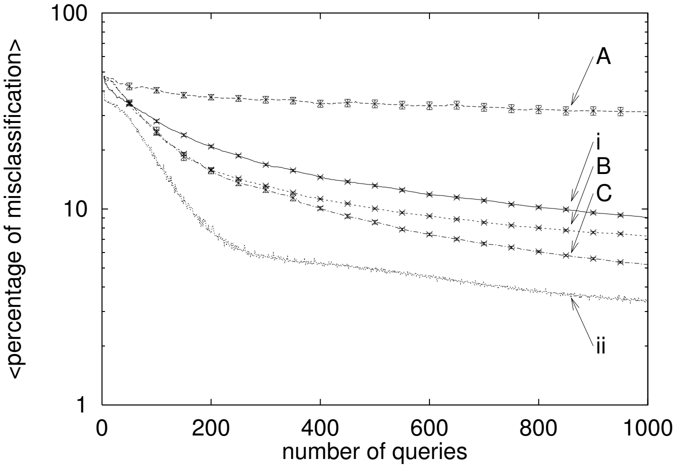


Figure 9. Two-spirals problem in 2D: the percentage of misclassification on the test set versus the number of queries for the different query strategies. A – random among Voronoi vertices, B – Voronoi vertex with maximum circumcircle, C – Voronoi vertex on the classification border with maximum circumcircle, i – random, ii – "optimal" (in this case, the abscissa denotes the number of the reference vectors; the number of queries was 50,000.).

For the random vector quantizer the reference vectors are i.i.d. values from the input space. In the active case the reference vectors are chosen according to the different query strategies as described above. The Voronoi data that are necessary for the active query strategies are calculated using the quickhull algorithm [55].

Since in this case, a theoretically optimum strategy cannot easily be devised, we approximate an optimally chosen set of reference vectors using the OLVQ algorithm [56] for learning vector quantization. This learning algorithm starts with a fixed number of randomly placed reference vectors and uses a training set of labeled samples to iteratively rearrange the reference vectors until a configuration of minimum training error is reached. We use a sample of 50,000 classified i.i.d. samples to create an omniscient student in this way.

Fig. 9 shows the simulation results for the two-spirals problem obtained by the query strategies A.-C. in comparison with (i) randomly selected reference vectors and (ii) the "omniscient" student that was trained on

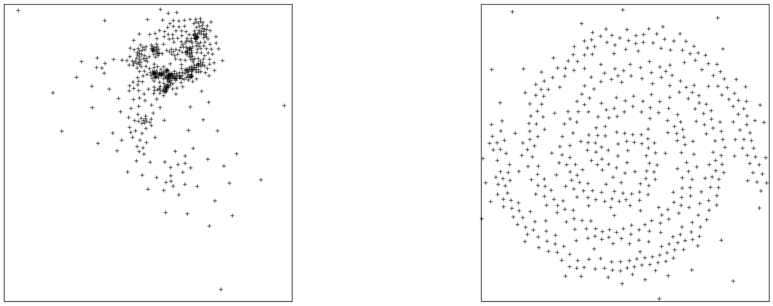


Figure 10. Typical set of reference vectors for query strategy A. (left) and query strategy C. (right).

50000 samples using the OLVQ algorithm [56] as described above. Each graph shows the average percentage of misclassification on the test set versus the number of reference vectors. All results are averages from 20 simulation runs.

Obviously, strategy A. is, despite its intuitive motivation, not very successful and below we will explain the reason for this failure. The remaining strategies B., C. and the passive learning strategy (i) are of a comparable performance during a short initial stage of learning. This is in line with the expectation that initially even randomly asked questions should earn informative answers; at the same time the active students have not yet gathered enough “knowledge” to optimize their queries significantly.

After learning has progressed for some time (after about 80 queries in Fig. 9), the advantage of the active strategies B. and C. over the passive student (i) becomes clearly visible. After 300 queries, the preference for a refinement of the classification boundary (curve C) in comparison to a more thorough exploration of the input space comes to fruition.

The generalization ability of the most successful query strategy C. tends to converge to that of an “omniscient” student. Note that for the positioning of its reference vectors, this “omniscient” student has the information from 50000 samples at its disposal.

Why did strategy A., that selects the reference vector at random among the Voronoi vertices, yield significantly poorer results than even a purely random selection strategy (curve (i))? An explanation for this can be found from Fig. 10 where we depict a typical set of reference vectors resulting from strategy A., Fig. 10 (left), and compare it with a set of reference vectors from strategy C., Fig. 10(right). Obviously, the student A. has concentrated most of its queries in a tiny subregion of the input space. Once such concentration has set in by some random fluctuation, the strategy A. favors the choice of further queries from the regions where the density is already higher. This “positive feedback” is not counteracted by an opposing requirement, such as the choice of the maximum circumcircle in strategies B. and C., and, as a consequence, the student becomes “trapped”.

5.3 Summary

In this section, we reviewed a heuristic query algorithm for local models that is based on the idea of selecting a query on the borderline of the current classification boundary.

For the implementation of this idea, we made use of the geometrical properties of these models that typically induce a Voronoi tessellation on the input space. We considered the Voronoi vertices as prospective query points and tested three different strategies for query selection among these Voronoi vertices on the two-spirals problem.

The active learning strategies needed a significantly smaller number of training samples to achieve the same generalization performance as the passive student. Among the active data selection strategies, the strategy that made best use of the available information in combination with a thorough exploration of the input space yielded the best results.

The query algorithm can easily be generalized to input spaces of dimensions higher than two but it should be kept in mind that the Voronoi diagram needs to be calculated explicitly. The complexity of this task grows exponentially with the dimension, limiting the application of the approach to dimensions in the range of about $1 \dots 8$.

6 Conclusion and Outlook

With active learning, we reviewed a learning technique that has become increasingly popular during the last few years. We concentrated mainly on active learning in classification tasks, thereby neglecting a number of approaches for active learning in regression, e.g. [57] [58] [59].

The aim of active learning is to minimize the cost of data acquisition. The basic assumption in active supervised learning is that unlabeled data are cheap but that it is expensive to obtain the labels of the data. In other words, in active learning the information that already is available from the data is exploited in an efficient way for the additional selection of data – instead of ignoring this information as is done in passive learning – with the expectation that the active student consequently needs fewer training samples than a passive student to achieve a fixed generalization ability.

The properties of active learning have been rigorously analyzed in the PAC framework for concept learning [42] [43] with the result that the set of PAC learnable concept classes is not enlarged by active learning but that in general fewer training samples are needed, i.e. the sample complexity can be reduced by active learning. This last point is confirmed by studies on active learning in more realistic learning scenarios. The reduction of the number of samples that is achieved by active learning is considerable, so that the additional computational cost that is incurred by the query algorithm is by far offset by the reduction in data acquisition costs.

While active learning techniques have been intensely studied in the domain of supervised learning, there have been only few approaches to incorporate active data selection into unsupervised learning [60] [61], although this class of learning algorithms offers a promising application field for active learning as well. Consider for example applications of image compression by vector quantization where it is known that part of the training data that can be obtained from an image or a series of images is redundant for the construction of an efficient codebook [62]. Here active learning could help to reduce the training costs. Another application area are unsupervised methods for the analysis of similarity data.

These methods, e.g. multidimensional scaling [63] or clustering methods for pairwise data [64] [65], generally scale with $\mathcal{O}(N^2)$ where N is the number of data items involved so that data acquisition for these algorithms quickly becomes an expensive task that can be alleviated by active data selection.

References

[1] Fahlmann, S. E. and Lebiere, C. (1990), "The cascade-correlation learning architecture", in D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, Vol. 2, pp. 524–532, Los Altos, CA. Morgan Kaufmann.

[2] LeCun, Y., Denker J. S., and Solla, S. A. (1990), "Optimal brain damage", in D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, Vol. 2, pp. 598–605, San Mateo, CA. Morgan Kaufmann.

[3] Littmann, E. and Ritter, H. (1996), "Learning and generalization in cascade network architectures", *Neural Computation*, Vol. 8, pp. 1521–1539.

[4] Riedmiller, M. (1994), "Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning techniques", *Computer Standards and Interfaces*, Vol. 16, pp. 265–278.

[5] Fedorov, V. V. (1972), *Theory of optimal experiments*. Academic Press, New York.

[6] Box, G. E. P. and Draper, N. R. (1987), *Empirical Model Building and Response Surfaces*. Wiley, New York.

[7] Atkinson, A. C. and Donev, A. N. (1992), *Optimum Experimental Designs*. Clarendon Press, Oxford.

[8] Valiant, L. G. (1984), "A theory of the learnable", *Communications of the ACM*, Vol. 27, pp. 1134–1142.

[9] Angluin, D. (1988), "Queries and concept learning", *Machine Learning*, Vol. 2, pp. 319–342.

[10] Angluin, D. (1987), "Learning regular sets from queries and counterexamples", *Information and Computation*, Vol. 75, pp. 87–106.

[11] Angluin, D. and Kharitonov, M. (1995), "When won't membership queries help?", *Journal of Computer and System Sciences*, Vol. 50, pp. 336–355.

[12] Plutowski, M. and White, H. (1993), "Selecting concise training sets from clean data", *IEEE Transactions on Neural Networks*, Vol. 4, pp. 305–318.

[13] Plutowski, M., Cottrell, G., and White, H. (1996), "Experience with selecting exemplars from clean data", *Neural Networks*, Vol. 9, pp. 273–294.

[14] Röbel, A. (1993), "The dynamic pattern selection algorithm: Effective training and controlled generalization of backpropagation neural networks", Technical Report 93–23, Technische Universität Berlin, Berlin.

[15] Cortes, C. and Vapnik, V. (1995), "Support-vector networks", *Machine Learning*, Vol. 20, 273–297.

[16] Guyon, I., Matić, N., and Vapnik, V. (1996), "Discovering informative patterns and data cleaning", in U. M. Fayyad, editor, *Advances in Knowledge Discovery and Data Mining*, pp. 181–20. AAI Press, Menlo Park, CA.

[17] Jung, G. and Opper, M. (1996), "Selection of examples for a linear classifier", *Journal of Physics A*, Vol. 29, 1367–1380.

[18] Munro, P. W. (1992), "Repeat until bored: A pattern selection strategy", in J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, Vol. 4, pp. 1001–1008, San Mateo, CA. Morgan Kaufmann.

[19] Cachin, C. (1994), "Pedagogical pattern selection strategies", *Neural Networks*, Vol. 7, pp. 175–181.

[20] Thrun, S. B. (1992), "The role of exploration in learning control", in D. A. White and D. A. Sofge, editors, *Handbook of Intelligent*

Control: Neural, Fuzzy and Adaptive Approaches, pp. 527–559. Van Nordstrand Reinhold, Florence, Kentucky.

[21] Thrun, S. (1995), “Exploration in active learning”, in M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pp. 381–384. MIT Press, Cambridge, MA.

[22] Ratsaby, J. (1998), “An incremental nearest neighbor algorithm with queries”, in M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Processing Systems*, Vol. 10, pp. 612–618, Cambridge, MA. MIT Press.

[23] Heskes, T. (1994), “The use of being stubborn and introspective”, in J. Dean, H. Cruse, and H. Ritter, editors, *Proceedings of the ZiF Conference on Adaptive Behavior and Learning*, pp. 55–65, Bielefeld.

[24] Leisch, F., Jain, L. C., and Hornik, K. (1998), “Cross-validation with active pattern selection for neural-network classifiers”, *IEEE Transactions on Neural Networks*, Vol. 9, pp. 35–41.

[25] van de Laar, P., Gielen, S., and Heskes, T. (1997), “Input selection with partial retraining”, in W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks – ICANN ’97*, pp. 469–474, Berlin. Springer.

[26] Kinzel, W. and Ruján, P. (1990), “Improving a network generalization ability by selecting examples”, *Europhysics Letters*, Vol. 13, pp. 473–477.

[27] Watkin, T. L. H. and Rau, A. (1992), “Selecting examples for perceptrons”, *Journal of Physics A: Mathematical and General*, Vol. 25, pp. 113–121.

[28] Kinouchi, O. and Caticha, N. (1992), “Optimal generalization in perceptrons”, *Journal of Physics A*, Vol. 25, pp. 6243–6250.

[29] Hwang, J.-N.,Choi, J. J., Oh, S., and Marks II, R. J. (1991), “Query-based learning applied to partially trained multilayer perceptrons”, *IEEE Transactions on Neural Networks*, Vol. 2, pp. 131–136.

[30] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988), "Learning internal representations by error propagation", in D. E. Rumelhart and J. L. MacClelland, editors, *Parallel Distributed Processing*, 7th ed., Vol. 1, chapter 8, pp. 318–362. MIT Pr., Cambridge, Mass.

[31] Linden, A. and Kindermann, J. (1989), "Inversion of multilayer nets", in *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, pp. 425–430, New York. IEEE Press.

[32] Baum, E. B. (1991), "Neural net algorithms that learn in polynomial time from examples and queries", *IEEE Transactions on Neural Networks*, Vol. 2, pp. 5–19.

[33] Hasenjäger, M. and Ritter, H. (1998), "Active learning with local models", *Neural Processing Letters*, Vol. 7, pp. 107–117.

[34] Sollich, P. (1994), "Query construction, entropy and generalization in neural network models", *Physical Review E*, Vol. 49, pp. 4637–4651.

[35] MacKay, D. J. C. (1992), "Information-based objective functions for active data selection", *Neural Computation*, Vol. 4, pp. 590–604.

[36] MacKay, D. J. C. (1992), "The evidence framework applied to classification networks", *Neural Computation*, Vol. 4, pp. 720–736.

[37] Cohn, D. A. (1996), "Neural network exploration using optimal experiment design", *Neural Networks*, Vol. 9, 1071–1083.

[38] Belue, L. M., Bauer Jr., K. W., and Ruck, D. W. (1997), "Selecting optimal experiments for multiple output multilayer perceptrons", *Neural Computation*, Vol. 9, pp. 161–183.

[39] Seung, H. S., Opper, M., and Sompolinsky, H. (1992), "Query by committee", in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 287–294, New York, NY. ACM Pr.

[40] Freund, Y., Seung, H. S., Shamir, E., and Tishby, N. (1997), "Selective sampling using the Query by Committee algorithm", *Machine Learning*, Vol. 28, pp. 133–168.

[41] Krogh, A. and Vedelsby, J. (1995), "Neural network ensembles, cross validation, and active learning", in G. Tesauro, D. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, Vol. 7, pp. 231–238, Cambridge, MA. MIT Pr.

[42] Eisenberg, B. and Rivest, R. L. (1990), "On the sample complexity of pac-learning using random and chosen examples", in M. Fulk and J. Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pp. 154–162, San Mateo, CA. Morgan Kaufmann.

[43] Kulkarni, S. R., Mitter, S. K., and Tsitsiklis, J. N. (1993), "Active learning using arbitrary binary valued queries", *Machine Learning*, Vol. 11, pp. 23–35.

[44] Baum, E. B. and Lang, K. (1992), "Query learning can work poorly when a human oracle is used", in *International Joint Conference on Neural Networks*, Beijing, China.

[45] Cohn, D. (1997), "Minimizing statistical bias with queries", in M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, Vol. 9, pp. 417–423, Cambridge, MA. MIT Pr.

[46] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996), "Active learning with statistical models", *Journal of Artificial Intelligence Research*, Vol. 4, pp. 129–145.

[47] Cohn, D., Atlas, L., and Ladner, R. (1994), "Improving generalization with active learning", *Machine Learning*, Vol. 15, pp. 201–221.

[48] Tishby, N., Levin, E., and Solla, S. (1989), "Consistent inference of probabilities in layered networks: Predictions and generalization", in *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, pp. 403–409, New York. IEEE Press.

[49] Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P. (1983), "Optimization by simulated annealing", *Science*, Vol. 220, pp. 671–680.

[50] Bachrach, R., Fine, S., and Shamir, E. (1998), "Query by Committee, linear separation and random walks", Accepted to *EuroColt-99*. Full version available at www.cs.huji.ac.il/labs/learning/Papers/qbc-main.ps.gz.

[51] Dagan, I. and Engelson, S. (1995), "Selective sampling in natural language learning", in *IJCAI-95 Workshop On New Approaches to Learning for Natural Language Processing*. Available at http://www.cs.biu.ac.il:8080/˜argamon/Access/ijcai-ml-nl95.ps.Z.

[52] Dagan, I. and Engelson, S. (1995), "Committee-based sampling for training probabilistic classifiers", in *Proceedings of the 12th International Conference on Machine Learning*, pp. 150–157, San Francisco, CA. Morgan Kaufmann.

[53] Liere, R. and Tadepalli, P. (1997) "Active learning with committees for text categorization", in *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence (AAAI–97/IAAI–97)*, pp. 591–596, Menlo Park, CA. AAAI Press.

[54] Lang, K. J. and Witbrock, M. J. (1989), "Learning to tell two spirals apart", in D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Summer School*, pp. 52–59, San Mateo, CA. Carnegie Mellon Univ., Morgan Kaufmann.

[55] Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996), "The quickhull algorithm for convex hulls", *ACM Transactions on Mathematical Software*, Vol. 22, pp. 469–483.

[56] Kohonen, T. (1995), *Self-Organizing Maps*, chapter 6, pp. 175–189. Springer, Berlin.

[57] Fukumizu, K. (1996), "Active learning in multilayer perceptrons", in D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, Vol. 8, pp. 295–301, Cambridge, MA. MIT Press.

[58] Paass, G. and Kindermann, J. (1995), "Bayesian query construction for neural network models", in G. Tesauro, D. Touretzky, and T. K. Leen, editors, *Advances in Neural Processing Systems*, Vol. 7, pp. 443–450, Cambridge, MA, 1995. MIT Pr.

[59] Sung, K. K. and Niyogi, P. (1995), "Active learning for function approximation", in G. Tesauro, D. Touretzky, and T. K. Leen, editors, *Advances in Neural Processing Systems*, Vol. 7, pp. 593–600, Cambridge, MA. MIT Pr.

[60] Hofmann, T. and Buhmann, J. M. (1998), "Active data clustering", in M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Processing Systems*, Vol. 10, pp. 528–534, Cambridge, MA. MIT Press.

[61] Hasenjäger, M., Ritter, H., and Obermayer, K. (1999), "Active learning in self-organizing maps", in E. Oja and S. Kaski, editors, *Kohonen Maps*, pp. 57–70. Elsevier, Amsterdam.

[62] Cohn, D., Riskin, E., and Ladner, R. (1994), "Theory and practice of vector quantizers trained on small training sets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 54–65.

[63] Borg, I. and Groenen, P. (1997), *Modern multidimensional scaling*. Springer, New York.

[64] Hofmann, T. and Buhmann, J. (1997), "Pairwise data clustering by deterministic annealing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, pp. 1–14.

[65] Graepel, T. and Obermayer, K. (1999), "A stochastic self-organizing map for proximity data", *Neural Computation*, Vol. 11, pp. 139–155.