# Query construction, entropy and generalization in Neural Network models

Peter Sollich

*Department of Physics, University of Edinburgh, Kings Buildings, Mayfield Road, Edinburgh EH9 3JZ, U.K.*

(To appear in Physical Review E)

**Abstract**

We study query construction algorithms, which aim at improving the generalization ability of systems that learn from examples by choosing optimal, non-redundant training sets. We set up a general probabilistic framework for deriving such algorithms from the requirement of optimizing a suitable objective function; specifically, we consider the objective functions entropy (or information gain) and generalization error. For two learning scenarios, the high-low game and the linear perceptron, we evaluate the generalization performance obtained by applying the corresponding query construction algorithms and compare it to training on random examples. We find qualitative differences between the two scenarios due to the different structure of the underlying rules (nonlinear and 'non-invertible' vs. linear); in particular, for the linear perceptron, random examples lead to the same generalization ability as a sequence of queries in the limit of an infinite number of examples. We also investigate learning algorithms which are ill-matched to the learning environment and find that in this case, minimum entropy queries can in fact yield a lower generalization ability than random examples. Finally, we study the efficiency of single queries and its dependence on the learning history, *i.e.* on whether the previous training examples were generated randomly or by querying, and the difference between globally and locally optimal query construction.

PACS numbers: 87.10.+e, 02.50.Ph, 02.50.Wp, 05.90.+m

## I. INTRODUCTION

In recent years, one of the main areas of research in the field of neural networks and machine learning has been the issue of *generalization*: Given a set of *training examples* generated by a *teacher* according to some underlying but unknown *rule*, one wants to generate, using a suitable *learning* or *training algorithm*, a *student* (*e.g.* a neural network) which can make intelligent guesses for previously unseen examples (for a review see *e.g.* [1] or the textbook [2]). The traditional approach has been to study the ability to generalize from *random examples*, where the input-output pairs which make up the training set are obtained by picking at random an input value according to some probability distribution, 'labeling' it with the corresponding output from the teacher and then possibly corrupting one or both of these values with some noise.

Recently, however, the alternative approach of *query learning* (or 'active data selection', 'experimental design' etc.) has attracted considerable interest. Here the inputs are not chosen at random, but rather by a *query selection algorithm* which, depending on the previously seen examples, selects an input value for the next input-output pair to be added to the training set. The motivation for query learning is that random examples often contain redundant information and that eliminating this redundancy must necessarily improve generalization performance or, equivalently, reduce the number of training examples necessary to attain a certain level of generalization performance. Query learning makes most sense in situations where labeling inputs is in some sense 'expensive', for example because the teacher output is actually the result of a complicated physical measurement, or where the cost of training itself increases strongly with the number of examples. There is of course a trade-off in so far as the query selection algorithm itself can be computationally expensive, possibly off-setting the savings due to the reduced number of training examples that are needed; for the sake of simplicity, we shall ignore this problem in our discussion.

Recent studies of query learning in the neural networks literature can be divided into groups according to the following two major distinctions: query selection algorithms can be *heuristic* or derived from optimization of an appropriate *objective function*, and they can either *construct* queries, *i.e.* calculate (maybe stochastically) the next input value to be queried, or they can *filter* queries from a source that provides a string of random input values.

Heuristic query construction has been studied in, for example, [3–6]. It has been pointed out before [7] that, while heuristic approaches can demonstrate the power of query learning in specific instances, they do not allow a systematic study of possible improvements of query selection algorithms, nor do their results generalize easily to learning problems other than those specifically considered. We shall therefore restrict our attention to query selection algorithms derived from optimization of objective functions. This approach has been used recently in [8,9] for query filtering and in [7,10] for construction. However, in these studies, only one objective function was considered, namely the *information gain* or entropy reduction per training example; also, the training algorithms were chosen in such a way that they directly reflected the a posteriori distribution of teachers and hence were optimally matched to the learning problem at hand. Only in [10] was the generalization error used as objective function for query construction, but only for one specific training algorithm.

We extend these considerations in the present paper by studying how the choice of objective function affects the performance of a query selection algorithm defined by optimizing it, and what the influence of a mismatch between training algorithm and a posteriori teacher distribution is. (This must be the typical real-world case, since of course it is never known in advance what the true a posteriori teacher distribution is.) By considering learning problems with both linear and nonlinear teacher rules we also study the effect of the nature of the rule on the efficiency of query learning. We restrict ourselves to rules which can be cast in the form of an input-output mapping, and focus on the case of query construction. Overall, our aim is not to provide practical query selection algorithms, but to study some of the basic properties of query learning.

We remark that the subject of query learning provides close links between the fields of neural networks, computational learning theory and statistics. In statistics in particular, the field of 'experimental design' has been studied in great depth. We cannot do justice to the vast body of literature published in this area and refer the interested reader to references [11–17] for some recent developments and reviews of older work. Our approach is closely related to the one used in 'optimal Bayesian sequential design' (see *e.g.* [18] and references therein); however, we allow for a distinction between a posteriori teacher distribution and learning algorithm as well as for different objective functions for query selection and performance evaluation.

The remainder of this paper is structured as follows: in section II we set up a general probabilistic framework for derivation of a query selection algorithm from a given objective function. We then apply this framework in sections III and IV. For a first pass at the problem of how the choice of objective function affects the performance of the corresponding query selection algorithms, we consider query construction based on optimization of the two objective functions *entropy* (or *information gain*) and *generalization error*. We study two specific learning problems, the 'high-low game' [9] and the 'linear perceptron'. These two examples allow us to gain some insight into the differences between query learning in linear and nonlinear systems. Since query selection is most effective when applied to all examples in the training set, *i.e.* when one allows the input of every new example which is added to the training set to be determined by the query construction algorithm, we consider the performance of the respective query construction algorithms when applied to generate *query sequences*, and compare the results to training on random examples. As performance measure we choose the generalization error (see section II for the precise definition) because generalization is after all what we want to improve by query selection. For the linear perceptron, we investigate the influence of a non-optimal learning algorithm which is poorly matched to the a posteriori teacher distribution. In section V we discuss some related issues: the efficiency of a single query and its dependence on the learning history, *i.e.* on whether the query is part of a query sequence or whether it is an *isolated query* after random examples; and the difference between *locally optimal* query selection, which builds up the training set step by step in a 'greedy' procedure, optimizing the given objective function at every step, and *globally optimal* query selection, which optimizes the whole query sequence for a given number of examples. We conclude in section VI with a summary and discussion of our results.

## II. THE FRAMEWORK

In this section we introduce a general probabilistic framework for the derivation of query selection algorithms based on optimization of a given objective function. The structure follows closely Wolpert's 'extended Bayesian framework' [19], but we adapt the notation to make it closer to that normally used in the Neural Networks community, see *e.g.* [1].

### Notation

We denote teachers by $\mathcal{V}$ and students by $\mathcal{N}$. Each teacher and each student implement a mapping from inputs $x$ (typically $\in \Re^N$) to outputs $y$ (often $\in \Re$). Let $\Theta^{(p)}$ denote a (ordered) training set consisting of $p$ examples $(x^\mu, y^\mu)$, $\mu = 1, \ldots, p$. We define the following probability distributions:

$P(y|x, \mathcal{V})$, the probability of, given input $x$, obtaining output $y$ from a teacher $\mathcal{V}$. This probability distribution specifies the input-output mapping implemented by the teacher $\mathcal{V}$, including possible corruption by noise. If $P(y|x, \mathcal{V})$ can be written in the form $\delta(y - f_\mathcal{V}(x))$, we call the teacher 'noise free', otherwise 'noisy'.

$P(x)$, the probability distribution of inputs when these are randomly selected, *i.e.* not queried. As commonly assumed, this distribution also governs the selection of test examples, from which the generalization error is calculated.

$P(\Theta^{(p)}|\mathcal{V})$, the probability of obtaining a specific training set from the teacher $\mathcal{V}$ (plus noise, possibly, which is always understood in the following). For randomly (and independently) selected examples, this can be written as

$$P(\Theta^{(p)}|\mathcal{V}) = \prod_{\mu=1}^{p} P(y^\mu|x^\mu, \mathcal{V}) P(x^\mu). \qquad (2.1)$$

$P(\mathcal{V})$, the a priori distribution of teachers [20].

$P(\mathcal{V}|\Theta^{(p)})$, the a posteriori teacher distribution, which can be calculated from $P(\Theta^{(p)}|\mathcal{V})$ and $P(\mathcal{V})$ using Bayes' Theorem

$$P(\mathcal{V}|\Theta^{(p)}) = \frac{P(\Theta^{(p)}|\mathcal{V})\ P(\mathcal{V})}{\int d\mathcal{V} P(\Theta^{(p)}|\mathcal{V})\ P(\mathcal{V})}. \qquad (2.2)$$

$P(\mathcal{N}|\Theta^{(p)})$, the 'post-training' distribution of students which specifies the learning algorithm in terms of the probability that training on the given training set $\Theta^{(p)}$ will yield the student $\mathcal{N}$. We shall assume that students are deterministic, *i.e.* that for each input $x$ a student $\mathcal{N}$ provides an output which can be written in the form $y = f_{\mathcal{N}}(x)$.

We emphasize that in a real-world learning problem, normally only $P(\mathcal{N}|\Theta^{(p)})$ and possibly $P(x)$ will be known, whereas all the probability distributions concerning the teachers $\mathcal{V}$ will be unknown. Quantitative analysis of a learning problem is only possible, however, if we make assumptions about these distributions, and indeed such assumptions have been made in all work on query learning to date. The necessity of such assumptions follows from the intuitively obvious result that, in the absence of any knowledge about the functional form and complexity of the teacher, generalization—and hence also its improvement by query selection—is impossible [19].

### Candidate objective functions

There are a variety of objective functions that one might want to be optimized by a query selection algorithm. We restrict our attention to two very common ones: Entropy (or information) and generalization error.

For a given training set $\Theta^{(p)}$, the entropy in teacher space can be defined as the entropy of the a posteriori distribution [21] $P(\mathcal{V}|\Theta^{(p)})$

$$S_{\mathcal{V}}(\Theta^{(p)}) = -\int d\mathcal{V}\, P(\mathcal{V}|\Theta^{(p)}) \ln P(\mathcal{V}|\Theta^{(p)}). \qquad (2.3)$$

The entropy in student space is defined similarly as a functional of the post-training distribution $P(\mathcal{N}|\Theta^{(p)})$ which depends on the learning algorithm that we are using. The information gain due to an additional example in either teacher or student space is defined as the decrease in the corresponding entropy.

We emphasize that student and teacher space entropy coincide *only* if $P(\mathcal{V}|\Theta^{(p)})$ and $P(\mathcal{N}|\Theta^{(p)})$ have exactly the same form. This is always the case in Bayesian analyses, where $\mathcal{V}$ and $\mathcal{N}$ are effectively identified (see *e.g.* [7]). In recent research on query learning [8,9] where the distinction between $\mathcal{V}$ and $\mathcal{N}$ was taken into account, the learning algorithm was nevertheless always chosen such that $P(\mathcal{V}|\Theta^{(p)})$ and $P(\mathcal{N}|\Theta^{(p)})$ were still identical. In the applications of our framework in the next section we shall see that new features can emerge if this is not the case.

The generalization error is probably the most commonly used measure of the performance of a student when trying to approximate a given teacher. It is defined starting from a specifically chosen error measure

$$e(y, x, \mathcal{N}) \qquad (2.4)$$

which determines how much the output of the student $\mathcal{N}$ for input $x$ is in error compared to the correct output $y$. Averaging this over all input/output pairs produced by a teacher $\mathcal{V}$, we obtain the generalization error, a measure of how closely $\mathcal{N}$ approximates $\mathcal{V}$:

$$\epsilon_{\mathrm{g}}(\mathcal{N}, \mathcal{V}) = \langle e(y, x, \mathcal{N}) \rangle_{P(y|x,\mathcal{V})P(x)}. \qquad (2.5)$$

In general, it has to be recognised that a decrease in entropy need not be correlated with a decrease in generalization error, *cf.* the discussion in [9].

For examples of a class of objective functions which have a somewhat intermediate character between entropy and generalization error, the 'prediction probabilities' and variants thereof, we refer the reader to [22,23].

### Derivation of query selection algorithms

We assume now that we are given an objective function, such as entropy or generalization error, which our query selection algorithm is supposed to optimize. We write this objective function generically in the form

$$\epsilon\,(\mathcal{N}, \mathcal{V}, \Theta^{(p)}).$$

We only consider query selection algorithms which are local in the sense that they work one example at a time, performing a greedy optimization of the given objective function at each query selection (see however section V, where we discuss what happens if this restriction is dropped). We also assume that after the new example is added to the training set, complete retraining takes place, *i.e.* the learning algorithm is re-applied to the enlarged training set $\Theta^{(p+1)}$. This excludes a dependence of query selection on the specific student (a representative of the distribution $P(\mathcal{N}|\Theta^{(p)})$ ) obtained after training on the existing training data as considered in [5,6]. Of course, a dependence on the actual—unknown—teacher $\mathcal{V}$ that generated the training data is not possible either, and so query selection can only be based on the existing training data $\Theta^{(p)}$. We therefore need to derive a function $\epsilon\,(x, \Theta^{(p)})$ which depends only on this existing training data and the next input, $x$. A query construction algorithm is then defined as picking, each time it is invoked, as the next query the value of $x$ at which $\epsilon\,(x, \Theta^{(p)})$ attains its global optimum (or randomly one such value if there is more than one global optimum). As pointed out above, we shall not be concerned with the actual implementation or computational complexity of this optimization process. In order to prevent the construction of nonsensical queries, we restrict the range of input values from which the query construction algorithm is allowed to choose to the support of $P(x)$, *i.e.* to values of $x$ which could also appear in a random training or test example. We remark in passing that the function $\epsilon\,(x, \Theta^{(p)})$, once obtained, can also be used to define a query filtering (as opposed to construction) algorithm which accepts a random input $x$ with a probability which is a function of the corresponding value of $\epsilon\,(x, \Theta^{(p)})$; however, we shall not consider this possibility further.

In order to obtain the function $\epsilon(x, \Theta^{(p)})$, which we do not want to depend on $\mathcal{N}$, we first average the given objective function over the post-training distribution:

$$\epsilon(\mathcal{V}, \Theta^{(p)}) = \langle \epsilon(\mathcal{N}, \mathcal{V}, \Theta^{(p)}) \rangle_{P(\mathcal{N}|\Theta^{(p)})}. \tag{2.6}$$

Averaging this over the a posteriori teacher distribution, we obtain an average objective function which depends on the training data only:

$$\epsilon(\Theta^{(p)}) = \langle \epsilon(\mathcal{V}, \Theta^{(p)}) \rangle_{P(\mathcal{V}|\Theta^{(p)})}, \tag{2.7}$$

We can now calculate the function defining the query construction algorithm by averaging (2.7), evaluated for the training data set $\Theta^{(p)} + (x, y)$, over the possible outputs $y$ that the teachers in the a posteriori distribution produce for the input $x$:

$$\epsilon(\Theta^{(p)}, x) = \langle \epsilon(\Theta^{(p)} + (x, y)) \rangle_{P(y|x,\Theta^{(p)})} \tag{2.8}$$

where $P(y|x, \Theta^{(p)})$ is given by

$$P(y|x, \Theta^{(p)}) = \int d\mathcal{V} P(y|x, \mathcal{V}) P(\mathcal{V}|\Theta^{(p)}). \tag{2.9}$$

It can be shown that the same result can be obtained by first evaluating $\epsilon(\mathcal{V}, \Theta^{(p)})$ for the training set $\Theta^{(p)} + (x, y)$, averaging over the outputs that $\mathcal{V}$ produces on $x$, and then averaging this over the a posteriori teacher distribution:

$$\epsilon(\Theta^{(p)}, x) = \langle \langle \epsilon(\mathcal{V}, \Theta^{(p)} + (x, y)) \rangle_{P(y|x,\mathcal{V})} \rangle_{P(\mathcal{V}|\Theta^{(p)})} \tag{2.10}$$

Equations (2.8) and (2.10) constitute the main result of this section and can be used interchangeably as definitions of the function $\epsilon(\Theta^{(p)}, x)$ which defines a query construction algorithm.

### Evaluation of performance of a query selection algorithm

The query construction algorithm as defined in the previous subsection yields a probability of querying $x$ if the existing training set is $\Theta^{(p)}$,

$$P_{\rm Q}(x|\Theta^{(p)}), \tag{2.11}$$

which is uniform over the set of all $x$ for which $\epsilon(\Theta^{(p)}, x)$ attains its global optimum and zero everywhere else. We shall evaluate the performance of this query construction algorithm when used to generate query sequences, using the generalization error as our performance measure. Starting from (2.5) we define first by analogy with (2.6) the average generalization error with respect to the post-training distribution:

$$\epsilon_{\rm g}(\mathcal{V}, \Theta^{(p)}) = \langle \epsilon_{\rm g}(\mathcal{N}, \mathcal{V}) \rangle_{P(\mathcal{N}|\Theta^{(p)})}. \tag{2.12}$$

We then define the average generalization error obtained after a sequence of $p$ queries when the true teacher is $\mathcal{V}$ as

$$\epsilon_{\rm g,Q}(\mathcal{V}) = \langle \epsilon_{\rm g}(\mathcal{V}, \Theta^{(p)}) \rangle_{P_{\rm Q}(\Theta^{(p)}|\mathcal{V})} \tag{2.13}$$

where the training sets are now generated according to the distribution

$$P_{\rm Q}(\Theta^{(p)}|\mathcal{V}) = \prod_{\mu=1}^{p} P(y^\mu|x^\mu, \mathcal{V}) P_{\rm Q}(x^\mu|\Theta^{(\mu-1)}) \tag{2.14}$$

in analogy to (2.1) which applies to the case of random examples. By averaging over the a priori teacher distribution, we obtain the generalization error for an average teacher:

$$\epsilon_{\rm g,Q} = \langle \epsilon_{\rm g,Q}(\mathcal{V}) \rangle_{P(\mathcal{V})} \tag{2.15}$$

In terms of $\epsilon_{\rm g}(\Theta^{(p)}) = \langle \epsilon_{\rm g}(\mathcal{V}, \Theta^{(p)}) \rangle_{P(\mathcal{V}|\Theta^{(p)})}$, this can be written as follows: $P(\mathcal{V}|\Theta^{(p)})$, the a posteriori distribution of teachers, can be shown to be the same for random examples and for query construction. This reflects the intuitively obvious fact that the way we generate data by querying or random sampling does not influence our inferences about the underlying rule [7]. Using Bayes' theorem, one can thus write

$$\begin{aligned}\epsilon_{\rm g,Q} &= \langle \langle \epsilon_{\rm g}(\mathcal{V}, \Theta^{(p)}) \rangle_{P_{\rm Q}(\Theta^{(p)}|\mathcal{V})} \rangle_{P(\mathcal{V})} \\ &= \langle \langle \epsilon_{\rm g}(\mathcal{V}, \Theta^{(p)}) \rangle_{P(\mathcal{V}|\Theta^{(p)})} \rangle_{P_{\rm Q}(\Theta^{(p)})} \\ &= \langle \epsilon_{\rm g}(\Theta^{(p)}) \rangle_{P_{\rm Q}(\Theta^{(p)})}.\end{aligned} \tag{2.16}$$

The distribution $P_{\rm Q}(\Theta^{(p)})$ can be written as

$$P_{\rm Q}(\Theta^{(p)}) = \prod_{\mu=1}^{p} P(y^\mu|x^\mu, \Theta^{(\mu-1)}) P_{\rm Q}(x^\mu|\Theta^{(\mu-1)}) \tag{2.17}$$

with $P(y^\mu|x^\mu, \Theta^{(\mu-1)})$ given by (2.9). For the case of training sets which are generated by a mixture of queries and random examples, one can still use (2.16) as long as in (2.17) the terms $P_{\rm Q}(x^\mu|\Theta^{(\mu-1)})$ are replaced by $P(x^\mu)$ for the examples $(x^\mu, y^\mu)$ that were generated randomly.

We remark that if one wants to know the average generalization error obtained by adding a query $x$ and the corresponding output $y$ to an existing fixed training set $\Theta^{(p)}$, all one needs to do is drop the average over $\Theta^{(p)}$ in (2.16), with the result

$$\begin{aligned}&\epsilon_{\rm g}(\Theta^{(p)} + 1 \text{ query}) \\ &= \langle \langle \epsilon_{\rm g}(\Theta^{(p)} + (x, y)) \rangle_{P(y|x,\Theta^{(p)})} \rangle_{P_{\rm Q}(x|\Theta^{(p)})} \\ &= \langle \epsilon_{\rm g}(\Theta^{(p)}, x) \rangle_{P_{\rm Q}(x|\Theta^{(p)})}.\end{aligned} \tag{2.18}$$

We have derived equations (2.13), (2.16) and (2.18) in order to show that for the evaluation of performance of a query selection algorithm, the same functions $\epsilon_{\rm g}(\mathcal{V}, \Theta^{(p)})$,

$\epsilon_g(\Theta^{(p)})$ and $\epsilon_g(\Theta^{(p)}, x)$ can be used that have to be calculated anyway for the derivation of minimum generalization error query construction. However, for the simple cases that we consider in the next sections, we shall avoid formal use of these results whenever more direct and intuitive derivations are possible.

## III. EXAMPLE: HIGH-LOW

In the next two sections we apply the framework set out in the preceding section to two specific learning problems, one nonlinear and one linear. We assume in both cases that the problem is learnable, *i.e.* that students and teacher have the same form. This is an abstraction from real-world problems—where unlearnable rules must occur frequently—which will have to be removed in further research.

In the present section we consider the 'high-low game' [8,9], which is an extremely simple example of a nonlinear rule with real input $x$ and binary output $y \in \{0,1\}$. The output is simply 1 or 0 depending on whether the input $x$ is greater or less than a certain preset threshold. Thus, for one-dimensional high-low, a noise free teacher is specified by a 'weight' $w_\mathcal{V}$ such that

$$P(y|x,\mathcal{V}) = \delta_{y,f_\mathcal{V}(x)}, \quad f_\mathcal{V}(x) = \Theta(x - w_\mathcal{V}) \tag{3.1}$$

where the Kronecker delta $\delta_{i,j}$ is equal to 1 if $i=j$ and 0 otherwise, and the step function $\Theta(x)$ is defined to be 1 if $x \geq 0$ and 0 otherwise. We assume that both inputs and teacher weights are taken from the unit interval $[0,1]$. An $N$-dimensional generalization of this can be defined as follows [9]: Inputs are now ordered pairs $(i,x)$, where $i \in \{1,2,\ldots,N\}$, $x \in [0,1]$, and a teacher $\mathcal{V}$ is defined in terms of an $N$-component vector $\mathbf{w}_\mathcal{V} = (w_{\mathcal{V},i})_{i=1,2,\ldots,N}$ and gives the output

$$f_\mathcal{V}(i,x) = \Theta(x - w_{\mathcal{V},i}). \tag{3.2}$$

As explained in [9], $N$-dimensional high-low is basically equivalent to $N$ concurrent one-dimensional high-low games.

As pointed out above we assume that the rule is learnable, *i.e.* that our students have the same functional form as the teachers, a student $\mathcal{N}$ being specified by an $N$-dimensional weight vector $\mathbf{w}_\mathcal{N}$. We assume the distribution of inputs to be $P(i,x) = P(i)P(x)$ with $P(i) = 1/N$, and $P(x)$ uniform on $[0,1]$ and zero everywhere else. We also assume the a priori teacher distribution $P(w_\mathcal{V})$ to be uniform on $[0,1]^N$. Under these assumptions, the a posteriori teacher distribution can easily be derived to be constant over the 'version space', *i.e.* the set of all teacher weight vectors which could have generated the training data, which is here simply a hypercube:

$$P(\mathcal{V}|\Theta^{(p)}) \propto \prod_{i=1}^{N} \Theta(w_{\mathcal{V},i} - x_{L,i})\Theta(x_{R,i} - w_{\mathcal{V},i}), \tag{3.3}$$

where we have denoted by $x_{L,i}$ and $x_{R,i}$ the largest and smallest $x$-value of inputs from the training set $\Theta^{(p)}$ with a given value of $i$ and output 0 and 1, respectively. The entropy in teacher space then follows from the definition (2.3) as

$$S_\mathcal{V}(\Theta^{(p)}) = \sum_{i=1}^{N} \ln(x_{R,i} - x_{L,i}). \tag{3.4}$$

For calculation of the generalization error, an obvious error measure is

$$e(y,(i,x),\mathcal{N}) = |y - f_\mathcal{N}(i,x)| \tag{3.5}$$

which is 0 if $y$ and $f_\mathcal{N}(i,x)$ agree and 1 otherwise, yielding

$$\epsilon_g(\mathcal{N},\mathcal{V}) = \frac{1}{N}\sum_{i=1}^{N} |w_{\mathcal{V},i} - w_{\mathcal{N},i}|. \tag{3.6}$$

We consider two learning algorithms: Zero temperature Gibbs learning, which is just given by

$$P_{\rm Gibbs}(\mathcal{N}|\Theta^{(p)}) = P(\mathcal{V}|\Theta^{(p)})\Big|_{\mathcal{V}=\mathcal{N}}; \tag{3.7}$$

and optimal learning in the sense of [24] for which

$$P_{\rm opt}(\mathcal{N}|\Theta^{(p)}) = \prod_{i=1}^{N} \delta(w_{\mathcal{N},i} - (x_{L,i} + x_{R,i})/2). \tag{3.8}$$

We remark that whereas for Gibbs learning the entropy in student space is identical to that in teacher space, the former is undefined for optimal learning, as is generally the case for deterministic learning algorithms.

For the generalization error averaged over the post-training distribution according to (2.6) and then over the a posteriori teacher distribution as in (2.7) one obtains

$$\epsilon_{\rm g,opt}(\Theta^{(p)}) = \frac{3}{4}\,\epsilon_{\rm g,Gibbs}(\Theta^{(p)}) = \frac{1}{4N}\sum_{i=1}^{N}(x_{R,i} - x_{L,i}). \tag{3.9}$$

Due to the proportionality between the two results we can restrict our attention to optimal learning in the following, dropping the subscript 'opt'.

Using (2.8), it is straightforward to calculate from (3.4) and (3.9) the defining functions for query construction for minimal teacher space entropy and minimal generalization error, respectively:

$$S_\mathcal{V}(\Theta^{(p)},(i,x)) = S_\mathcal{V}(\Theta^{(p)}) + q_i \ln q_i + (1-q_i)\ln(1-q_i) \tag{3.10}$$

$$\epsilon_g(\Theta^{(p)},(i,x)) = \epsilon_g(\Theta^{(p)}) - \frac{x_{R,i} - x_{L,i}}{2N}\, q_i(1-q_i), \tag{3.11}$$

where we have used the abbreviation

$$q_i = P(y=1|(i,x),\Theta^{(p)}) = \begin{cases} 0 & x \le x_{L,i} \\ (x-x_{L,i})/(x_{R,i}-x_{L,i}) & x_{L,i} < x < x_{R,i} \\ 1 & x \ge x_{R,i}. \end{cases} \tag{3.12}$$

Equations (3.10) and (3.11) are both minimized for $q_i = 1/2$, *i.e.* $x = (x_{L,i} + x_{R,i})/2$. This corresponds to the intuitively obvious method of bisecting a component of the version space. For $q_i = 1/2$ the value of $S_{\mathcal{V}}(\Theta^{(p)}, (i,x))$ is independent of $i$, so that query construction for minimal teacher space entropy selects randomly any of the $N$ possible values for $i$ and then $x = (x_{L,i}+x_{R,i})/2$. By contrast, query construction for minimal generalization error can only select from those $i$-values for which $x_{R,i} - x_{L,i}$ is maximal since only then will $\epsilon_{\mathrm{g}}(\Theta^{(p)}, (i,x))$ be minimized. Thus, query construction for minimal generalization error specifies along which component the version space should be bisected, a piece of information which cannot be obtained from the requirement of maximal information gain. In fact, as explained in [9], one can, simply by always bisecting the same component of the version space, construct a sequence of queries which at each step achieves the maximal entropy reduction but for which the generalization error never drops below a finite threshold.

The difference between the two objective functions, entropy and generalization error, is reflected in the average performance of the two query construction algorithms when they are used to generate query sequences: Query construction for minimal generalization error yields, after a sequence of $p = \alpha N = ([\alpha] + \Delta\alpha)N$ queries (where $[\alpha]$ denotes the integer part of $\alpha$ and $\Delta\alpha = \alpha - [\alpha]$ its non-integer part) and the corresponding outputs, a version space with $\Delta\alpha N$ components of length $(1/2)^{[\alpha]+1}$ and $(1-\Delta\alpha)N$ components of length $(1/2)^{[\alpha]}$ and hence from (3.9) a generalization error of

$$\epsilon_{\mathrm{g}}(\text{min. gen. error queries}) = \frac{1}{4}\left(\frac{1}{2}\right)^{[\alpha]}\left(1 - \frac{\Delta\alpha}{2}\right), \tag{3.13}$$

so that increasing $\alpha$ by one always reduces the generalization error by a factor of $1/2$. For minimal teacher space entropy, on the other hand, one obtains after a sequence of $p$ queries a version space with components of length $(1/2)^{p_1}, (1/2)^{p_2}, \ldots, (1/2)^{p_N}$ where $p_i$ is the number of times the $i$-th component of the version space has been bisected ($\sum_i p_i = p$); averaging over the distribution of the $p_i$ one obtains

$$\begin{aligned} &\epsilon_{\mathrm{g}}(\text{min. entropy queries}) \\ &= \frac{1}{4N}\sum_{\{p_i\}}\frac{p!}{N^p p_1!\cdots p_N!}\sum_{i=1}^{N}\left(\frac{1}{2}\right)^{p_i} \\ &= \frac{1}{4}\left[\left(1 - \frac{1}{2N}\right)^N\right]^\alpha. \end{aligned} \tag{3.14}$$

Comparing (3.13) and (3.14), we see that for $N = 1$, teacher space entropy and generalization error perform equally well as objective functions for query sequence construction, whereas for $N \ge 2$ a query sequence constructed for minimization of teacher space entropy needs to contain more examples than one constructed for minimization of generalization error in order to obtain the same generalization performance. As $N \to \infty$, $(1 - 1/2N)^N \to \exp(-1/2)$ and thus $-\ln(1/2)/(1/2) = \ln 4 \approx 1.39$ as many examples are needed.

We have seen that query construction both for minimal teacher space entropy and minimal generalization error yields a generalization error which decays exponentially with the number of examples normalized by the number of parameters of the high-low rule, $\alpha = p/N$, which is a drastic improvement over the case of random examples where the generalization error only decays algebraically with $\alpha$. This result for random examples has been given in [8] for $N = 1$ as $\epsilon_{\mathrm{g}}(\text{random examples}) = 1/2(p+2)$; for $N \ge 2$ it generalizes to

$$\begin{aligned} &\epsilon_{\mathrm{g}}(\text{random examples}) = \\ &= \sum_{\{p_i\}}\frac{p!}{N^p p_1!\cdots p_N!}\frac{1}{4N}\sum_{i=1}^{N}\frac{2}{p_i+2} \\ &= \frac{N}{2(p+1)}\left\{1 - \frac{N}{p+2}\left[1 - \left(1 - \frac{1}{N}\right)^{p+2}\right]\right\} \end{aligned} \tag{3.15}$$

which as $\alpha = p/N \to \infty$ gives a decay with $1/2\alpha + O(1/\alpha^2)$ from the inequalities

$$\begin{aligned} \frac{1}{2(\alpha+2)} &\le \epsilon_{\mathrm{g}}(\text{random examples}) \\ &\le \frac{1}{2\alpha}\left[1 - \frac{1}{\alpha}\left(1 - e^{-\alpha}\right)\right]. \end{aligned} \tag{3.16}$$

Our results in this section show that in the learning scenario considered, the teacher space entropy (or for the case of zero temperature Gibbs learning, the equivalent student space entropy) can serve as a useful guideline for query construction and does provide a large increase in generalization performance over random examples, but does not achieve quite as good a performance as query construction for minimum generalization error.

## IV. EXAMPLE: LINEAR PERCEPTRON

As a second application of the query learning framework set out in section II we now consider the linear perceptron. A teacher is specified by a vector $\mathbf{w}_{\mathcal{V}} \in \Re^N$ such that it yields (in the absence of noise) the output

$$f_{\mathcal{V}}(\mathbf{x}) = \frac{1}{\sqrt{N}}\mathbf{w}_{\mathcal{V}}^T\mathbf{x} \tag{4.1}$$

for the input $\mathbf{x}$ which is also an $N$-dimensional vector. Again, we take the problem to be learnable, and thus assume students to be of the same functional form, with weight vectors $\mathbf{w}_{\mathcal{N}}$. We will mainly be interested in the 'thermodynamic limit' $N \to \infty$, $p \to \infty$ at constant $\alpha = p/N$.

For convenience, we consider inputs $\mathbf{x}$ from a spherical distribution,

$$P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N\sigma_x^2), \tag{4.2}$$

and a Gaussian prior on teacher space

$$P(\mathcal{V}) = \mathcal{N}\left(\mathbf{0}, \sigma_{\mathcal{V}}^2 \mathbf{1}\right) \propto \exp(-\mathbf{w}_{\mathcal{V}}^2/2\sigma_{\mathcal{V}}^2). \tag{4.3}$$

Here we have used the notation $\mathcal{N}(\mu, \mathbf{\Sigma})$ for a multivariate Gaussian distribution with mean $\mu$ and covariance matrix $\mathbf{\Sigma}$, and denoted by $\mathbf{1}$ the $N$-dimensional unit matrix.

In order to fix $P(y|\mathbf{x}, \mathcal{V})$, we consider two forms of noise: Gaussian noise on the output of variance $1/\beta_{\mathcal{V}}$, *i.e.*

$$P(y|\mathbf{x}, \mathcal{V}) = \mathcal{N}\left(f_{\mathcal{V}}(\mathbf{x}), 1/\beta_{\mathcal{V}}\right) \tag{4.4}$$

and Gaussian noise on the teacher weights, yielding the output corresponding to a perturbed weight vector $\mathbf{w}'_{\mathcal{V}}$ distributed as $\mathcal{N}\left(\mathbf{w}_{\mathcal{V}}, \tilde{\sigma}_{\mathcal{V}}^2 \mathbf{1}\right)$:

$$P(y|\mathbf{x}, \mathcal{V}) = \langle \delta(y - f_{\mathcal{V}'}(\mathbf{x})) \rangle_{\mathbf{w}'_{\mathcal{V}}} \tag{4.5}$$

$$= \mathcal{N}\left(f_{\mathcal{V}}(\mathbf{x}), \tilde{\sigma}_{\mathcal{V}}^2 \mathbf{x}^2/N\right) \tag{4.6}$$

which under the spherical constraint for the inputs, (4.2), is of the same functional form as (4.4) and need not be considered separately in what follows; all results for noise on the output also hold for noise on the weights with the replacement $\beta_{\mathcal{V}} \to 1/\sigma_x^2 \tilde{\sigma}_{\mathcal{V}}^2$.

Combining (4.3) and (4.4) and using Bayes' formula one obtains that the a posteriori teacher distribution $P(V|\Theta^{(p)})$ is a Gaussian distribution $\mathcal{N}\left(\mathbf{M}_{\mathcal{V}}^{-1}\mathbf{a}, (\beta_{\mathcal{V}}\mathbf{M}_{\mathcal{V}})^{-1}\right)$ where we have set

$$\mathbf{M}_{\mathcal{V}} = \frac{1}{\beta_{\mathcal{V}} \sigma_{\mathcal{V}}^2} \mathbf{1} + \frac{1}{N} \sum_{\mu=1}^{p} \mathbf{x}^\mu (\mathbf{x}^\mu)^T \tag{4.7}$$

and

$$\mathbf{a} = \frac{1}{\sqrt{N}} \sum_{\mu=1}^{p} y^\mu \mathbf{x}^\mu. \tag{4.8}$$

The entropy in teacher space is thus simply

$$S_{\mathcal{V}}(\Theta^{(p)}) = -\frac{N}{2} \ln \beta_{\mathcal{V}} - \frac{1}{2} \ln |\mathbf{M}_{\mathcal{V}}| + \text{constant}. \tag{4.9}$$

Its independence of the outputs $y^\mu$ in the training set reflects the well known fact that in linear models information-based objective functions always lead to query selection algorithms or 'experimental designs' which can be expressed solely in terms of the input values of the training examples [7,11].

For calculation of the generalization error we start from the commonly used quadratic error measure

$$e(y, \mathbf{x}, \mathcal{N}) = \frac{1}{2}(y - f_{\mathcal{N}}(\mathbf{x}))^2 \tag{4.10}$$

which yields according to (2.5) the generalization error between student $\mathcal{N}$ and teacher $\mathcal{V}$

$$\epsilon_{\mathrm{g}}(\mathcal{N}, \mathcal{V}) = \frac{1}{2\beta_{\mathcal{V}}} + \frac{\sigma_x^2}{2N}(\mathbf{w}_{\mathcal{N}} - \mathbf{w}_{\mathcal{V}})^2. \tag{4.11}$$

The constant term which arises from the noise on the teacher alone will be omitted in the following.

For the learning algorithm, we take Gibbs learning with weight decay (see *e.g.* [25]), specified by a learning temperature $T = 1/\beta$ and weight decay parameter $\tilde{\lambda}$:

$$\begin{aligned} &P(\mathcal{N}|\Theta^{(p)}) \\ &\propto \exp\left[-\beta\left(\sum_{\mu=1}^{p} \frac{1}{2}(y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2 + \frac{\tilde{\lambda}}{2}\mathbf{w}_{\mathcal{N}}^2\right)\right] \\ &= \mathcal{N}\left(\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a}, (\beta\mathbf{M}_{\mathcal{N}})^{-1}\right), \end{aligned} \tag{4.12}$$

where we have introduced the matrix $\mathbf{M}_{\mathcal{N}}$, defined as

$$\mathbf{M}_{\mathcal{N}} = \tilde{\lambda}\mathbf{1} + \frac{1}{N}\sum_{\mu=1}^{p} \mathbf{x}^\mu (\mathbf{x}^\mu)^T \tag{4.13}$$

which only differs from $\mathbf{M}_{\mathcal{V}}$ by a multiple of the unit matrix. It follows from (4.13) that $\tilde{\lambda}/\sigma_x^2$ is a dimensionless quantity which we denote by

$$\lambda = \frac{\tilde{\lambda}}{\sigma_x^2} \tag{4.14}$$

and also simply refer to as the weight decay parameter. The student space entropy is from (4.12)

$$S_{\mathcal{N}}(\Theta^{(p)}) = -\frac{N}{2} \ln \beta - \frac{1}{2} \ln |\mathbf{M}_{\mathcal{N}}| + \text{constant}. \tag{4.15}$$

Averaging over the post-training distribution according to (2.6) and then over the a posteriori teacher distribution as in (2.7) we get for the average generalization error as a function of the training set

$$\begin{aligned} \epsilon_{\mathrm{g}}(\Theta^{(p)}) = \frac{\sigma_x^2}{2N} \Big[ &\left(\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a} - \mathbf{M}_{\mathcal{V}}^{-1}\mathbf{a}\right)^2 \\ &+ \frac{1}{\beta}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-1} + \frac{1}{\beta_{\mathcal{V}}}\mathrm{tr}\ \mathbf{M}_{\mathcal{V}}^{-1} \Big] \end{aligned} \tag{4.16}$$

Since a finite training temperature $T = 1/\beta$ only gives a positive definite additive contribution to the generalization error, we restrict ourselves to the case $T = 0$, *i.e.* $\beta \to \infty$ in the following [26]. We remark that optimal learning in the sense of [24] is obtained as a special case of Gibbs learning (at $T = 0$) by setting the weight decay parameter $\lambda$ to its optimal value

$$\lambda_{\mathcal{V}} = \frac{1}{\beta_{\mathcal{V}}\sigma_{\mathcal{V}}^2\sigma_x^2} = \frac{1}{s^2} \tag{4.17}$$

where

$$s = (\beta_{\mathcal{V}}\sigma_{\mathcal{V}}^2\sigma_x^2)^{1/2} = \left(\frac{\langle y^2\rangle_{P(\mathbf{x})P(\mathcal{V})} - 1/\beta_{\mathcal{V}}}{1/\beta_{\mathcal{V}}}\right)^{1/2} \tag{4.18}$$

is the root-mean-squared signal to noise ratio of the training examples. $\lambda_{\mathcal{V}} = 0$ thus corresponds to the limit of a noise free teacher, and a non-zero $\lambda_{\mathcal{V}}$ measures the typical amount of corruption of noise relative to the average uncorrupted signal; for $\lambda_{\mathcal{V}} = 1$ noise and signal levels are equal on average. In the special case of optimal weight decay, one has $\mathbf{M}_{\mathcal{V}} = \mathbf{M}_{\mathcal{N}}$ and hence the generalization error assumes the simple form

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}) = \frac{\sigma_x^2}{2N}\frac{1}{\beta_{\mathcal{V}}}\mathrm{tr}\ \mathbf{M}_{\mathcal{V}}^{-1}. \tag{4.19}$$

From (4.9) and (2.8) the defining function for query construction for minimal teacher space entropy follows immediately as

$$S_{\mathcal{V}}(\Theta^{(p)}, \mathbf{x}) = S_{\mathcal{V}}(\Theta^{(p)}) + \frac{1}{2}\ln|\mathbf{M}_{\mathcal{V}}| - \frac{1}{2}\ln|\mathbf{M}'_{\mathcal{V}}|, \tag{4.20}$$

where $\mathbf{M}'_{\mathcal{V}}$ is defined as the value of $\mathbf{M}_{\mathcal{V}}$ calculated for the training set $\Theta^{(p)}$ with the new example $(\mathbf{x}, y)$ added:

$$\mathbf{M}'_{\mathcal{V}} = \mathbf{M}_{\mathcal{V}} + \frac{1}{N}\mathbf{x}\mathbf{x}^T. \tag{4.21}$$

The analogous expression for the case of the student space entropy as objective function is obtained simply by replacing $\mathbf{M}_{\mathcal{V}}$ by $\mathbf{M}_{\mathcal{N}}$, whereas the corresponding result for the generalization error, which can be straightforwardly derived from (4.16) and (2.8), is:

$$\epsilon_{\mathrm{g}}(\Theta^{(p)}, \mathbf{x}) = \frac{\sigma_x^2}{2N}\left[\frac{1}{\beta_{\mathcal{V}}}\mathrm{tr}\ \mathbf{M}_{\mathcal{V}}'^{-1} + \left(\mathbf{M}_{\mathcal{N}}'^{-1}\mathbf{a}'_{\mathcal{V}} - \mathbf{M}_{\mathcal{V}}'^{-1}\mathbf{a}'_{\mathcal{V}}\right)^2 \right.$$
$$\left. + \frac{1}{N\beta_{\mathcal{V}}}\left(1 + \frac{1}{N}\mathbf{x}^T\mathbf{M}_{\mathcal{V}}^{-1}\mathbf{x}\right)\left(\mathbf{M}_{\mathcal{N}}'^{-1}\mathbf{x} - \mathbf{M}_{\mathcal{V}}'^{-1}\mathbf{x}\right)^2\right] \tag{4.22}$$

where

$$\mathbf{M}'_{\mathcal{N}} = \mathbf{M}_{\mathcal{N}} + \frac{1}{N}\mathbf{x}\mathbf{x}^T \tag{4.23}$$

and

$$\mathbf{a}'_{\mathcal{V}} = \mathbf{a} + \frac{1}{N}\mathbf{x}\mathbf{x}^T\mathbf{M}_{\mathcal{V}}^{-1}\mathbf{a}. \tag{4.24}$$

For the case of optimal weight decay this simplifies to

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}, \mathbf{x}) = \frac{\sigma_x^2}{2N}\frac{1}{\beta_{\mathcal{V}}}\mathrm{tr}\ \mathbf{M}_{\mathcal{V}}'^{-1}. \tag{4.25}$$

It is to this simpler case that we now turn.

**Optimal weight decay**

In the case of optimal weight decay, it is straightforward to derive that under the spherical constraint (4.2) the defining functions for query construction for minimal teacher space entropy, (4.20), student space entropy (which can be derived analogously from (4.15)), and generalization error, (4.25), are *all* optimized (*i.e.* minimized) by choosing the query $\mathbf{x}$ along the direction of an eigenvector of $\mathbf{M}_{\mathcal{V}}$ with minimal eigenvalue. For $p < N$, *i.e.* $\alpha < 1$ this amounts to choosing $\mathbf{x}$ to be perpendicular to the subspace spanned by the previous training inputs $\mathbf{x}^\mu$, $\mu = 1, \ldots, p$, an intuitively obvious result.

Applying this query construction algorithm to generate a sequence of queries, one sees that with each new query the lowest eigenvalue of $\mathbf{M}_{\mathcal{V}}$ is increased by $\sigma_x^2$. After $p = \alpha N$ queries $\mathbf{M}_{\mathcal{V}}$ thus has a $(\Delta\alpha N)$-fold eigenvalue $(\lambda_{\mathcal{V}} + [\alpha] + 1)\sigma_x^2$ and a $(1 - \Delta\alpha)N$-fold eigenvalue $(\lambda_{\mathcal{V}} + [\alpha])\sigma_x^2$ (we use the decomposition $\alpha = [\alpha] + \Delta\alpha$ introduced earlier). Thus from (4.19) one obtains

$$\epsilon_{\mathrm{g,opt}}(\text{optimal queries}) = \frac{1}{2\beta_{\mathcal{V}}}G_{\mathrm{Q}}(\lambda_{\mathcal{V}}) \tag{4.26}$$

with

$$\begin{aligned} G_{\mathrm{Q}}(\lambda_{\mathcal{V}}) &= \frac{\sigma_x^2}{N}\langle \mathrm{tr}\ \mathbf{M}_{\mathcal{V}}^{-1}\rangle_{P_{\mathrm{Q}}(\Theta^{(p)})} \\ &= \frac{\Delta\alpha}{\lambda_{\mathcal{V}} + [\alpha] + 1} + \frac{1 - \Delta\alpha}{\lambda_{\mathcal{V}} + [\alpha]}. \end{aligned} \tag{4.27}$$

This result can now be compared to the generalization error achieved by training on random examples. We use the results of Krogh *et al.* [27], who have calculated in the limit $N \to \infty$ the function [28]

$$\begin{aligned} G(\lambda_{\mathcal{V}}) &= \frac{\sigma_x^2}{N}\langle \mathrm{tr}\ \mathbf{M}_{\mathcal{V}}^{-1}\rangle_{P(\Theta^{(p)})} \\ &= \frac{1}{2\lambda_{\mathcal{V}}}\left(1 - \alpha - \lambda_{\mathcal{V}} + \sqrt{(1 - \alpha - \lambda_{\mathcal{V}})^2 + 4\lambda_{\mathcal{V}}}\right) \end{aligned} \tag{4.28}$$

which is the analogue of $G_{\mathrm{Q}}(\lambda_{\mathcal{V}})$ for random examples. Thus, for the average generalization error after training on $p$ random examples, one has

$$\epsilon_{\mathrm{g,opt}}(\text{random examples}) = \frac{1}{2\beta_{\mathcal{V}}}G(\lambda_{\mathcal{V}}). \tag{4.29}$$

The generalization error $\epsilon_{\mathrm{g,opt}}$ as a function of $\alpha$ is shown in figure 1 for various values of $\lambda_{\mathcal{V}} = 1/s^2$, both for random examples and for query sequences. Also shown is the relative reduction in generalization error due to query selection, *i.e.* the ratio of (4.29) and (4.26) which we denote by

$$\kappa(\alpha) = \frac{\epsilon_{\mathrm{g}}(\text{optimal queries})}{\epsilon_{\mathrm{g}}(\text{random examples})}. \tag{4.30}$$

For moderate noise levels (a numerical calculation yields $\lambda_\mathcal{V} \leq 0.92$), the maximum of $\kappa(\alpha)$ is reached at $\alpha = 1$; its height

$$\kappa(\alpha = 1) = \frac{1}{2}(1 + \lambda_\mathcal{V}) \left[ \left(1 + \frac{4}{\lambda_\mathcal{V}}\right)^{1/2} - 1 \right] \quad (4.31)$$

decreases monotonically with $\lambda_\mathcal{V}$—hence increases with the signal-to-noise ratio $s$—and is simply given by $(\lambda_\mathcal{V})^{-1/2} = s$ in the limit of small $\lambda_\mathcal{V}$. Query construction thus yields the greatest improvement of generalization performance for low noise levels. The fact that in the low noise regime the maximum of $\kappa(\alpha)$ is at $\alpha = 1$ can be understood from the fact that for random examples the average eigenvalue spectrum [29] of $\mathbf{M}_\mathcal{V}$ extends down to $(\lambda_\mathcal{V} + (1 - \sqrt{\alpha})^2)\sigma_x^2$ which tends to zero as $\lambda_\mathcal{V} \to 0$ and $\alpha \to 1$, making tr $\mathbf{M}_\mathcal{V}^{-1}$ much larger than for the case of query construction, where at $\alpha = 1$ all eigenvalues of $\mathbf{M}_\mathcal{V}$ are $(\lambda_\mathcal{V} + 1)\sigma_x^2$. For larger noise levels, the maximum of $\kappa(\alpha)$ shifts to larger integer values of $\alpha$ and has a height which can be bounded by $1 + 4/\lambda_\mathcal{V}$ and which thus tends to 1 in the limit of large noise levels, $\lambda_\mathcal{V} \to \infty$.

The plots in figure 1 suggest that independently of the value of $\lambda_\mathcal{V}$, $\kappa(\alpha)$ tends to 1 as $\alpha \to \infty$, which means that for a sufficiently large number of examples, the relative improvement in generalization error that can be obtained from optimal queries as compared to random examples tends to zero. This can be confirmed by an asymptotic expansion of $\kappa(\alpha)$ which yields

$$\kappa(\alpha) = 1 + \frac{1}{\alpha} + O\left(\frac{1}{\alpha^2}\right). \quad (4.32)$$

The above result is in stark contrast to the results for the high-low game obtained above and similar results for the binary perceptron [8,9], where the asymptotic behaviour of $\kappa(\alpha)$ for large $\alpha$ is

$$\kappa(\alpha) \propto \frac{1}{\alpha} \left(\exp(-c\alpha)\right)^{-1} \quad (4.33)$$

for some positive constant $c$, which clearly tends to infinity as $\alpha \to \infty$. A plausible explanation for this qualitative difference might be that in the limit of a noise free teacher $N$ examples are actually enough to specify a teacher linear perceptron completely, so that beyond $\alpha = 1$ one is trying to reduce generalization error due to noise; by contrast, for high-low or the binary perceptron, the teacher cannot be uniquely specified by any finite set of examples even in the noise free limit. In this sense, the high-low game and the binary perceptron are 'non-invertible' for any finite $\alpha$, and thus by querying the average amount of information about the teacher that can be gained from each new training example can be kept finite as $\alpha \to \infty$. This property was shown in [9] to be a sufficient condition for exponentially decaying generalization error, at least for the specific query filtering algorithm considered there. For the linear perceptron, on the other hand, the information available about the

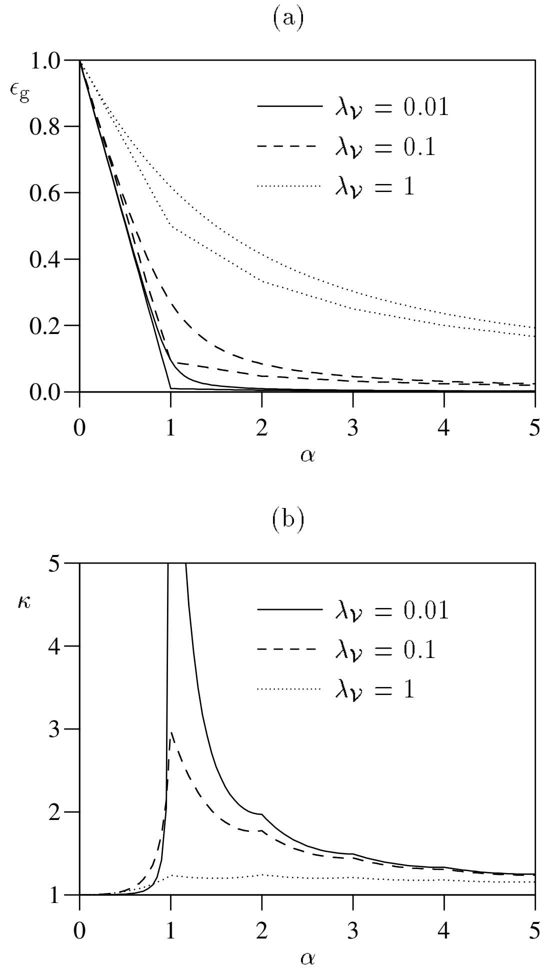


FIG. 1. (a) Generalization error $\epsilon_g(\alpha)$ achieved by training a linear perceptron on $\alpha N$ *random examples* (higher, 'smooth' curves) and on the same number of examples generated by *query sequences* (lower curves), in units of $\sigma_x^2 \sigma_\mathcal{V}^2/2$. The weight decay $\lambda$ is assumed to be set to its optimal value $\lambda_\mathcal{V}$; hence minimum generalization error and minimum entropy queries are identical. The values of the teacher noise level $\lambda_\mathcal{V} = 1/s^2$ are 0.01, 0.1 and 1. (b) Relative improvement in generalization error due to querying, $\kappa(\alpha)$, defined as the ratio of the values of $\epsilon_g$ for random examples and for query sequences.

teacher is, loosely speaking, 'exhausted' at $\alpha = 1$ and the information that can be gained from each new training example tends to zero as $\alpha \to \infty$.

### Non-optimal weight decay

We now turn to the case of non-optimal weight decay, where the weight decay parameter $\lambda$ in the learning algorithm is not set to the optimal value determined by the signal-to-noise ratio of the teacher as in (4.17).

We consider first query construction for minimization of the entropy in teacher space (4.9). Since this quantity is independent of $\lambda$, the query construction algorithm remains the same as for optimal weight decay. (The same conclusion holds for the case of minimization of the student space entropy.) Therefore, as in the preceding section, query construction only depends on the previous input $\mathbf{x}^\mu$ and not on the corresponding outputs. To calculate the average generalization error after a sequence of $p$ minimum entropy queries, *i.e.* the average of (4.16) over the training set distribution obtained by querying, $P_{\rm Q}(\Theta^{(p)})$, as defined in (2.16), one can therefore first perform the average over the $y^\mu$ to derive

$$\begin{aligned}&\epsilon_{\rm g}(\text{min. entropy queries})\\ &= \frac{\sigma_x^2\sigma_{\mathcal{V}}^2}{2}\left[\lambda_{\mathcal{V}} G_{\rm Q}(\lambda) + (\lambda_{\mathcal{V}} - \lambda)\frac{dG_{\rm Q}(\lambda)}{d\lambda}\right] \qquad (4.34)\end{aligned}$$

where the average over the $\mathbf{x}^\mu$ is taken care of in the definition of the function $G_{\rm Q}(\cdot)$ in (4.27). The analogous equation for the case of random examples as derived in [27] is obtained simply by replacing $G_{\rm Q}(\cdot)$ with $G(\cdot)$. The resulting values of $\kappa(\alpha)$ are plotted in figure 2 for various values of $\lambda$ and $\lambda_{\mathcal{V}}$. The most striking feature is that now $\kappa$ can actually assume values smaller than 1, implying that minimal entropy query construction leads to a *higher* generalization error than random examples, a seemingly counter-intuitive result. It can be checked numerically, however, that $\kappa < 1$ occurs only when $\lambda$ is smaller than the optimal value $\lambda_{\mathcal{V}}$, combined with high teacher noise levels $\lambda_{\mathcal{V}} \geq 2$ and values of $\alpha$ for which the underlying rule is only just beginning to be learnt, in that $\epsilon_{\rm g}$ is still more than over 80% of its value at $\alpha = 0$, *i.e.* before any training examples were presented. In these cases the learning algorithm is *overconfident* in that it underestimates the amount of noise in the training examples, making the entropy reduction or information gain a spurious indicator of an improvement in generalization ability. The correlation between reductions in entropy and generalization error is recovered as soon as $\alpha$ is large enough for the generalization error to be significantly smaller than at $\alpha = 0$; in the limit of an infinite number of training examples, one has

$$\begin{aligned}\kappa(\alpha) = 1 + \frac{1}{\alpha} + \frac{1}{\alpha^2}\Bigg[1 - 2\lambda_{\mathcal{V}} + 2\frac{(\lambda-\lambda_{\mathcal{V}})^2}{\lambda_{\mathcal{V}}}\\ - \Delta\alpha(1-\Delta\alpha)\Bigg] + O\left(\frac{1}{\alpha^3}\right) \qquad (4.35)\end{aligned}$$

so that $\kappa$ is again greater than one for large $\alpha$. The last result also shows that for fixed, large $\alpha$, $\kappa$ increases with increasing $(\lambda - \lambda_{\mathcal{V}})^2$, *i.e.* with the degree of mismatch between the learning algorithm and the actual learning problem at hand.

Now we consider for comparison the performance of query construction for minimal generalization error defined by minimizing (4.22). Since we have not been able to perform this minimization analytically for the general case, we restrict our attention to the special case in which $\mathbf{a}$ is an eigenvector of the matrix $\mathbf{M}_{\mathcal{V}}$, and to the limit of a noise free teacher, $\lambda_{\mathcal{V}} \to 0$. If we also assume that $\mathbf{M}_{\mathcal{V}}$ has full rank, *i.e.* that at least $N$ training examples with linearly independent input vectors have been presented, then only the second term in (4.22) survives:

$$\epsilon_{\rm g}(\Theta^{(p)}, \mathbf{x}) = \frac{\sigma_x^2}{2N}\left(\mathbf{M}_{\mathcal{N}}'^{-1}\mathbf{a}_{\mathcal{V}}' - \mathbf{M}_{\mathcal{V}}'^{-1}\mathbf{a}_{\mathcal{V}}'\right)^2 \qquad (4.36)$$

Setting

$$\mathbf{\Delta}' = \mathbf{M}_{\mathcal{N}}'^{-1}\mathbf{a}_{\mathcal{V}}' - \mathbf{M}_{\mathcal{V}}'^{-1}\mathbf{a}_{\mathcal{V}}' \qquad (4.37)$$

$$\mathbf{\Delta} = \mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a} - \mathbf{M}_{\mathcal{V}}^{-1}\mathbf{a} \qquad (4.38)$$

one can derive that

$$\mathbf{\Delta}' = \mathbf{\Delta} - \mathbf{M}_{\mathcal{N}}'^{-1}\frac{1}{N}\mathbf{x}\mathbf{x}^T\mathbf{\Delta} \qquad (4.39)$$

and under the above assumptions and the spherical constraint (4.2) one finds that $\epsilon_{\rm g}(\Theta^{(p)}, \mathbf{x})$ is minimized by choosing $\mathbf{x}$ along $\mathbf{\Delta}$ and hence along $\mathbf{a}$. This makes intuitive sense: Under our assumption that $\mathbf{a}$ is an eigenvector of $\mathbf{M}_{\mathcal{V}}$, $\mathbf{a}$ is proportional to $\mathbf{M}_{\mathcal{V}}^{-1}\mathbf{a}$ which is in fact the true teacher, $\mathbf{w}_{\mathcal{V}}$, due to the assumptions of full rank of $\mathbf{M}_{\mathcal{V}}$ and $\lambda_{\mathcal{V}} \to 0$, so that querying along $\mathbf{a}$ yields the largest possible signal $y = \mathbf{w}_{\mathcal{V}}^T\mathbf{x}/\sqrt{N} = \sigma_x|\mathbf{w}_{\mathcal{V}}|$ and hence reduces the generalization error (4.36) (which is due to the mismatch between $\lambda \neq 0$ and $\lambda_{\mathcal{V}} = 0$) most quickly. We remark that $\mathbf{x} \propto \mathbf{a}$ is a truly sequential query construction criterion since it involves, through $\mathbf{a}$, the previous outputs. This in contrast to query construction for minimum entropy where the optimal query $\mathbf{x}$ is determined solely by the preceding inputs as discussed above.

Let us now apply the query construction criterion $\mathbf{x} \propto \mathbf{a}$ to a simple case where the above assumptions are fulfilled. Namely, consider a noise free teacher $\mathbf{w}_{\mathcal{V}}$ and a training set of $N$ examples generated by minimum entropy query construction, *i.e.* containing $N$ mutually orthogonal input vectors and thus having $\mathbf{M}_{\mathcal{V}} = \sigma_x^2\mathbf{1}$ and $\mathbf{a} = \sigma_x^2\mathbf{w}_{\mathcal{V}}$. Querying at $\mathbf{x} = \mathbf{a}(N\sigma_x^2/\mathbf{a}^2)^{1/2}$ then yields a new matrix $\mathbf{M}_{\mathcal{V}}' = \mathbf{M}_{\mathcal{V}} + \mathbf{a}\mathbf{a}^T(\sigma_x^2/\mathbf{a}^2)$ and a new vector

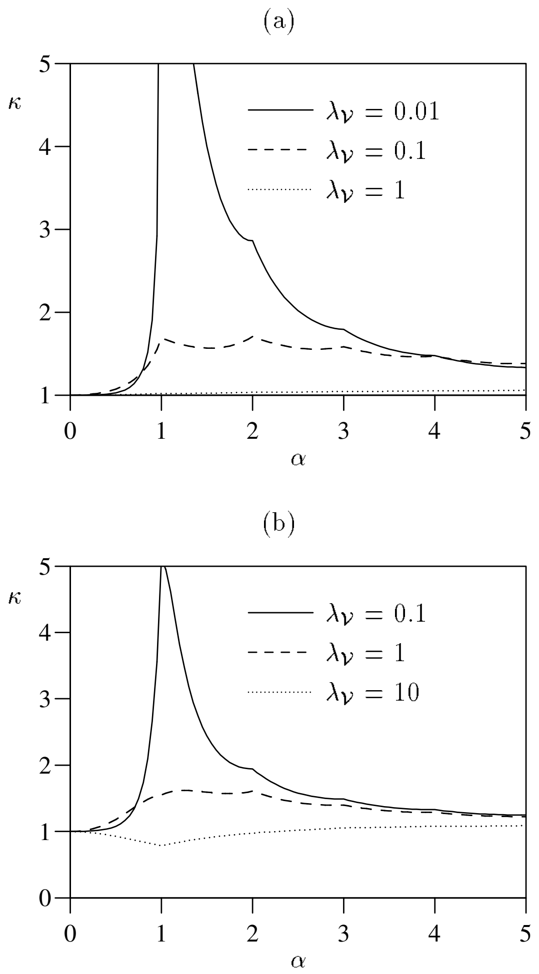


FIG. 2. $\kappa(\alpha)$ for minimum entropy queries for the case of non-optimal weight decay $\lambda \neq \lambda_{\mathcal{V}}$. (a) 'Under-confident', *i.e.* unnecessarily large weight decay $\lambda = 10\lambda_{\mathcal{V}}$, for $\lambda_{\mathcal{V}}$=0.01, 0.1 and 1. (b) 'Over-confident', *i.e.* inappropriately small weight decay $\lambda = \lambda_{\mathcal{V}}/10$, for $\lambda_{\mathcal{V}}$=0.1, 1 and 10. Notice that in the last case values of $\kappa(\alpha) < 1$ appear, *i.e.* that a sequence of minimum entropy queries can lead to higher generalization error than random examples.

$\mathbf{a}' = \mathbf{a} + y\mathbf{x}/\sqrt{N} = \mathbf{a} + \sigma_x^2 \mathbf{w}_{\mathcal{V}} = 2\mathbf{a}$; $\mathbf{a}'$ is thus again an eigenvector of $\mathbf{M}'_{\mathcal{V}}$ and hence the next minimum generalization error query will have to be selected along $\mathbf{a}'$, *i.e.* again along $\mathbf{a}$. $\delta p = N\delta\alpha$ such queries in sequence generate a matrix $\mathbf{M}_{\mathcal{V}}$ and a vector $\mathbf{a}$ with $\mathbf{M}_{\mathcal{V}}\mathbf{a} = (\delta p + 1)\sigma_x^2 \mathbf{a}$ and $\mathbf{a} = \sigma_x^2 \mathbf{w}_{\mathcal{V}}(1 + \delta p)$, thus leading to a generalization error (using (2.12) and (4.11))

$$\begin{aligned} \epsilon_{\mathrm{g}}(\Theta^{(N+\delta p)}, \mathcal{V}) &= \frac{\sigma_x^2}{2N}\left(\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a} - \mathbf{w}_{\mathcal{V}}\right)^2 \\ &= \frac{\sigma_x^2 \sigma_{\mathcal{V}}^2}{2} \frac{\mathbf{w}_{\mathcal{V}}^2}{N\sigma_{\mathcal{V}}^2} \frac{\lambda^2}{(\lambda + N\delta\alpha + 1)^2} \end{aligned} \tag{4.40}$$

This result contains the size of the perceptron, $N$, and for fixed $\delta\alpha$ converges to zero in the thermodynamic limit $N \to \infty$, implying that in this limit $\epsilon_{\mathrm{g}}$ expressed as a function of $\alpha$ has a step discontinuity at $\alpha = 1$. This result in itself, due to the limiting assumptions that we had to make, is probably less important than a more general conclusion which can be drawn: For query construction even in purely linear learning problems, maximizing information gain is not necessarily identical to minimizing generalization error, and to obtain the optimal generalization performance one will generally have to resort to truly sequential query selection.

## V. OTHER ISSUES

In the preceding sections we have focussed our attention on query construction when applied to generate query sequences. We now turn to two other interesting aspects of query construction: Single queries and locally vs. globally optimal query construction. We again investigate them for the two example learning scenarios considered above, confining ourselves to query construction for minimum entropy in the case of the linear perceptron in order to keep things analytically tractable.

### Single queries

We refer to a single query which is constructed on the basis of an existing training set of random examples as 'isolated'. It is then natural to ask the question: How does the improvement in generalization capability due to an isolated query, *i.e.* the decrease in generalization error, compare with that due to a query in a query sequence and that due to a random example? The first comparison concerns the question of how the performance of a single query depends on the previous learning history, *i.e.* on the method by which the previous training examples have been generated (randomly or by querying). It is not entirely obvious if for answering this question the relative or the absolute decrease in generalization error is the relevant quantity, and we shall consider both of these options below.

### 1. High-low

As derived in section II, the average generalization error after a single query can simply be calculated by averaging the function $\epsilon_{\rm g}(\Theta^{(p)}, x)$ over the respective query construction distribution $P_{\rm Q}(x|\Theta^{(p)})$. For the high-low game, we thus find from (3.11) that a single query constructed for minimum generalization error and teacher space entropy, respectively, reduces the generalization error by

$$\Delta\epsilon_{\rm g}(\text{1 min. gen. err. query}) = \frac{1}{8N}\max_i (x_{R,i} - x_{L,i}) \tag{5.1}$$

and

$$\begin{aligned}&\Delta\epsilon_{\rm g}(\text{1 min. entropy query})\\ &= \frac{1}{8N}\frac{1}{N}\sum_{i=1}^{N}(x_{R,i} - x_{L,i}) = \frac{1}{2N}\epsilon_{\rm g}(\Theta^{(p)}).\end{aligned} \tag{5.2}$$

Let us first consider the dependence of these results on the learning history. From (5.2), a minimum entropy query reduces the generalization error by an amount proportional to the generalization error before querying—which will therefore be large for previous training examples generated randomly and smaller if queries have been used—, making the relative improvement independent of the learning history. Comparing (5.1) and (5.2) one sees that a minimum generalization error query provides, as expected, a greater reduction (for $N \geq 2$; for $N = 1$ the two query construction algorithms are equivalent) in generalization error than a minimum entropy query, which is also more strongly dependent on the learning history. For previous training examples generated using minimum generalization error queries, the maximum in (5.1) is $(1/2)^{[\alpha]}$ as follows from the discussion before equation (3.13), giving an absolute decrease in generalization error decaying exponentially with the number of examples; from (3.13), the corresponding relative decrease is $(1-\frac{\Delta\alpha}{2})^{-1}/2N$ and thus between $1/2N$ (the value for a minimum entropy query) and $1/N$. The difference to the case of previous random training examples is most clearly exhibited for $N \to \infty$, because in this limit it follows from the well-known combinatorial 'collector's problem' (see *e.g.* [30]) that for any $\alpha$ there is with probability one at least one component of the version space for which no training examples exist at all, making the maximum in (5.1) equal to 1 and yielding an absolute decrease in generalization error of $1/8N$, independently of $\alpha$. From (3.16) the corresponding relative decrease [31] is $(\alpha + O(1))/4N$.

We now compare isolated queries to random examples. From (3.15), one finds that the absolute decrease in generalization error due to a random example after previous random training examples is given by $1/(2N\alpha^2) + O(1/N\alpha^3)$, yielding a relative decrease of $1/N\alpha + O(1/N\alpha^2)$. As $\alpha \to \infty$, this tends to zero, reflecting the fact that the information carried by new random examples becomes more and more redundant. By comparison, for an isolated minimum entropy query we found above that the relative decrease in generalization error is $1/2N$, showing that minimum entropy query construction successfully avoids this redundancy. For a minimum generalization error query and in the limit $N \to \infty$, the relative generalization error decrease of $(\alpha + O(1))/4N$ is still larger, by a factor of $\alpha/2 + O(1)$, than the relative decrease achieved by a minimum entropy query, implying that minimum generalization error query construction selects among all queries providing non-redundant information the one with the greatest potential for improving generalization.

### 2. Linear perceptron

We now turn to the case of the linear perceptron. As pointed out before, we consider only the case of query construction for minimum (teacher or student space) entropy. In this case the reduction in generalization error due to a single query is particularly easy to calculate, since only the change in $G(\lambda)$ and $dG(\lambda)/d\lambda$ (or $G_{\rm Q}(\lambda)$ and $dG_{\rm Q}(\lambda)/d\lambda$, respectively) needs to be worked out. One obtains a result which in general depends on the learning history through the minimal eigenvalue of $\mathbf{M}_{\mathcal{N}}$, which we write as $(\lambda + \lambda_{\min})\sigma_x^2$. For $\alpha < 1$, however, there is no such dependence since one always has $\lambda_{\min} = 0$ because the correlation matrix $\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^T$ does not have full rank. In the case of previous random examples [29] one obtains, using the fact that in the thermodynamic limit $N \to \infty$ the eigenspectrum of $\mathbf{M}_{\mathcal{N}}$ is self-averaging,

$$\lambda_{\min} = \begin{cases} 0 & \text{for } \alpha \leq 1 \\ (\sqrt{\alpha} - 1)^2 & \text{for } \alpha > 1. \end{cases} \tag{5.3}$$

For a query in a query sequence, one simply has $\lambda_{\min} = [\alpha]$ as discussed in section IV. Using these values of $\lambda_{\min}$, one finds that almost always, a query in a sequence leads to an absolute reduction in generalization error less or equal to that due to an isolated query. The exception is the case of overconfidence and high noise level, where at finite $\alpha$ a query in a sequence can reduce the generalization error by a larger amount (or increase it by a smaller amount) than an isolated query. Asymptotically, a query in a query sequence reduces the generalization error by $(1/2\beta_{\mathcal{V}}N)\alpha^{-2}(1 + O(\alpha^{-1}))$, which corresponds to a relative decrease of $1/N\alpha + O(1/N\alpha^2))$, whereas for an isolated query both the absolute and relative reductions are bigger by a factor of $1 + 4\alpha^{-1/2} + O(\alpha^{-1})$.

Now let us compare isolated queries to random examples. The reduction in generalization error due to a single random example can be straightforwardly obtained by differentiating the analogue of (4.34) for random examples with respect to $N\alpha$, and is shown in figure 3 along with the corresponding results for isolated queries. It can be seen that the trend of the comparison between query sequences and sequences of random examples discussed in section IV is mirrored in the result for isolated queries and single random examples: For optimal weight decay, an isolated query always performs better than a random example (maximally, it reduces the generalization error by 5 times as much as a random example, which is achieved at $\alpha = 9/4$ in the limit $\lambda_\mathcal{V} \to 0$) whereas for non-optimal, over-confident weight decay and small $\alpha$ it can perform worse. Asymptotically, the reduction due to an isolated query is greater by a factor of $1+4\alpha^{-1/2}+O(\alpha^{-1})$ than that due to an additional random example.

To summarize our discussion of single minimum entropy queries for the linear perceptron, we have found quite a different behaviour than for the high-low game, as would have been expected from the significant differences between the two systems regarding the efficiency of query sequences. Whereas minimum entropy queries in the high-low game—whether isolated or in a query sequence—lead to a relative improvement in generalization error which remains finite as $\alpha \to \infty$, the relative improvements in the linear perceptron decay towards zero roughly as $1/\alpha$ and to lowest order in $1/\sqrt{\alpha}$ are identical to those obtained from random examples. Again, we argue that the reason for this qualitative difference is that for large $\alpha$ learning in the linear perceptron is mainly learning against noise, for which queries are not significantly more useful than random examples.

We found for both high-low and the linear perceptron with optimal weight decay that the *absolute* reduction in generalization error is always larger for an isolated query than for one in a query sequence, whether we consider minimum generalization error or minimum entropy queries. This result makes intuitive sense because, if the previous training examples have already been generated by queries, one expects there to be less scope for reducing the generalization error by another query. We speculate that this might be more generally valid in learning problems where the training algorithm is well-matched to the learning environment, *i.e.* the a posteriori teacher distribution. For the linear perceptron with non-optimal weight decay, *i.e.* a poorly matched training algorithm, we find that the above does hold at least asymptotically (as $\alpha \to \infty$) for minimum entropy queries, but not necessarily for finite $\alpha$. In terms of the *relative* reduction in generalization error, we observe that for large $\alpha$ an isolated query still performs better than one in a query sequence, whereas for small $\alpha$ it can be shown that one can also have the reverse relationship between the two.

### Locally vs. globally optimal query construction

All our considerations so far have been based on the assumption that query construction can be viewed as a 'greedy' optimization of some appropriate objective function. If one is looking for query construction algorithms

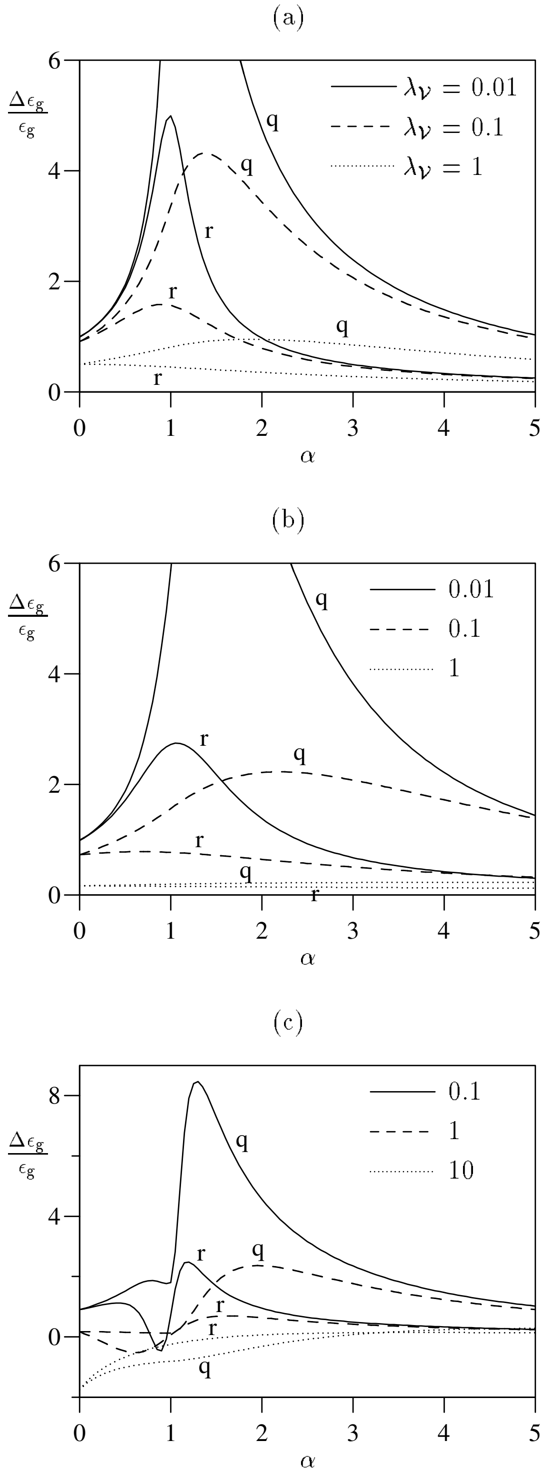


FIG. 3. Relative reduction in generalization error (in units of $1/N$) due to an isolated minimum entropy query and a random example, respectively. (a) Optimal weight decay $\lambda = \lambda_\mathcal{V}$; $\lambda_\mathcal{V}$=0.01, 0.1, 1. (b) 'Under-confident' weight decay $\lambda = 10\lambda_\mathcal{V}$; $\lambda_\mathcal{V}$=0.01, 0.1, 1. (c) 'Over-confident' weight decay $\lambda = \lambda_\mathcal{V}/10$; $\lambda_\mathcal{V}$=0.1, 1, 10. Notice that in the last case for high teacher noise level ($\lambda_\mathcal{V}$=10) a minimum entropy query can reduce the generalization error less than an additional random example.

which are applicable independently of the total number of queries that will eventually be used in the learning process, this approach, which we shall call 'locally optimal query construction', is perfectly reasonable. If the total number of allowed queries were known, one might want to optimize the query construction algorithm 'globally' in order to achieve the optimum of the relevant objective function after learning from the specified number of queries and the corresponding outputs (see *e.g.* [32]). It is the aim of the present section to compare the performance of globally and locally optimal query construction, with the goal of assessing the loss in performance that one incurs if one restricts oneself to locally optimal query construction. We emphasize that what we mean by globally optimal query construction is not identical to what is normally referred to as 'statistical' (or 'exact') design in the statistics literature, where all queries are chosen before any outputs are received; globally optimal query construction shares with this approach the fact that the total number of training examples is fixed, but sequentially selects each new query on the basis of all preceding training examples, inputs and outputs alike. We also stress that the major disadvantage of globally optimal query construction is that it is tied to the specific number $p$ of queries that is considered; in fact, one must expect that a globally optimal sequence of $p$ queries cannot be augmented by more queries later without leading to suboptimal generalization performance.

We shall first consider the question of possible equivalence of locally and globally optimal query construction in terms of the final value of the relevant objective function that they achieve. Intuitively, one expects that if a globally optimal sequence of $p$ queries can always be augmented by another query to give a globally optimal sequence of $p+1$ queries, then globally optimal query sequences can be constructed using a local, *i.e.* step-by-step approach. This criterion can be formalized and one can check that it does indeed hold for the high-low game, whether generalization error or entropy is used as the objective function for query construction; thus locally and globally optimal query construction perform equally well. For the linear perceptron, however, the situation is different, as we now show. Consider the case of optimal weight decay, where the generalization error is given by (4.19). From the convexity inequality

$$\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{V}}^{-1} \geq \left(\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{V}}\right)^{-1} = (\sigma_x^2(\lambda_{\mathcal{V}}+\alpha))^{-1} \qquad (5.4)$$

one has the bound

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}) \geq \frac{1}{2\beta_{\mathcal{V}}}\frac{1}{\lambda_{\mathcal{V}}+\alpha}. \qquad (5.5)$$

For $\alpha < 1$, this bound can be tightened using the fact that $\mathbf{M}_{\mathcal{V}}$ must have at least $N(1-\alpha)$ eigenvalues of size $\lambda_{\mathcal{V}}\sigma_x^2$:

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}) \geq \frac{1}{2\beta_{\mathcal{V}}}\left(\frac{\alpha}{\lambda_{\mathcal{V}}+1}+\frac{1-\alpha}{\lambda_{\mathcal{V}}}\right) \qquad (5.6)$$

A result from [33] shows that the above inequalities can be made into equalities by appropriate choice of the $\mathbf{x}^\mu$, so that globally optimal query construction for minimum generalization error saturates the bounds (5.5),(5.6). Comparing this with the result (4.26),(4.27) for locally optimal query construction, one sees that the two achieve identical performance for $\alpha \leq 1$ and for the integer values $\alpha = 2, 3, \ldots$, but that for all other values of $\alpha$ locally optimal query construction performs worse. This can also be read off from figure 4 which shows the ratio $\rho$ of the generalization error achieved by globally and locally optimal query selection as a function of $\alpha$, for different values of $\lambda_{\mathcal{V}}$. This ratio attains its minimum of 8/9 at $\alpha = 3/2$ for $\lambda_{\mathcal{V}} \to 0$, and is for large $\alpha$ given by $1-\Delta\alpha(1-\Delta\alpha)/\alpha^2 + O(\alpha^{-3})$, showing that although locally optimal query construction in general performs worse for finite $\alpha$, it 'catches up' again with globally optimal query construction asymptotically.

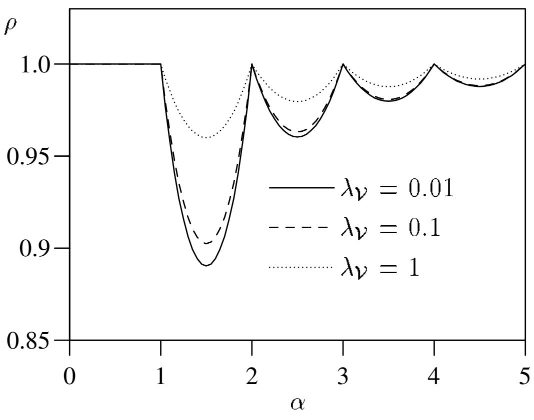


FIG. 4. Ratio $\rho$ of generalization error achieved by globally vs. locally optimal query sequences, for the linear perceptron with optimal weight decay $\lambda = \lambda_{\mathcal{V}}$. Values of $\lambda_{\mathcal{V}}$ are 0.01, 0.1, 1. The globally optimal query sequence leads to a generalization error which is at most smaller by a factor of 8/9 (at $\alpha = 3/2$ and for $\lambda_{\mathcal{V}} \to 0$) than that of the locally optimal query sequence.

To illustrate the reason for the difference between locally and globally optimal query selection, we consider briefly the case $N = 2$, $p = \alpha N = 3$. The locally optimally query construction algorithm selects the first two queries $\mathbf{x}^1$ and $\mathbf{x}^2$ orthogonal to each other and the third one randomly, leading to a $(2\times 2)$ correlation matrix $(1/N)\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^T$ with eigenvalues $\sigma_x^2$ and $2\sigma_x^2$. Globally optimal query selection selects the three queries $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ at angles of 120° to each other, making the eigenvalues of the correlation matrix both equal to $3/2\sigma_x^2$ and thus saturating the bound (5.5). This example also illustrates another point which was mentioned above: for an unknown total number of training examples globally optimal query construction is not normally a good idea. If, after having chosen the globally optimal queries for $p = 3$, we were allowed an additional query, we would end up with a correlation matrix with eigenvalues $3/2\sigma_x^2$, $5/2\sigma_x^2$ which does not saturate the bound (5.5), whereas the locally optimal query construction algorithm would select the fourth query orthogonal to $\mathbf{x}^3$, yielding the optimal correlation matrix with two eigenvalues of $2\sigma_x^2$.

Summarizing, we have found that in general locally optimal query construction will perform worse than its globally optimal equivalent, but that, at least for the two learning problems we have considered, the differences in performance, if they exist, become negligibly small for large values of $\alpha$. Overall, the advantage of locally optimal query construction algorithms, namely their applicability whatever the total number of training examples is, thus seems to compensate well for the loss in performance compared to globally optimal query construction. It remains a matter of further research to establish how general this result is.

## VI. CONCLUSION

In the present paper, we have considered query construction algorithms derived by optimization of appropriate objective functions. After setting up a general probabilistic framework for this problem in section II, we have explored in sections III and IV the differences between the objective functions entropy and generalization error in two learning scenarios, the high-low game and the linear perceptron, by evaluating the average generalization ability obtained after training on examples generated by a sequence of queries. We have found that there are strong qualitative differences which are due to the different structure of the underlying rule in the two scenarios: In the high-low game with its nonlinear and 'non-invertible' rule, the generalization error decays exponentially with $\alpha$, the number of examples normalized by the number of parameters in the system, which is a dramatic improvement over the asymptotic decay with $1/\alpha$ for random examples. For the linear perceptron with its purely linear rule, on the other hand, we have found that the relative reduction in generalization error due to querying is much less pronounced and indeed is given by a reduction factor $\kappa(\alpha)$ as small as $1 + 1/\alpha$ for large $\alpha$. We have related this qualitative difference to the fact that in the high-low game query construction can realize a finite information gain per training example as $\alpha \to \infty$, whereas for the linear perceptron the maximal information gain per example tends to zero in this limit, the available information essentially being 'exhausted' at $\alpha = 1$.

As to the difference between entropy and generalization error as objective functions for query construction we have found that most of the time the entropy can serve as a useful guideline for query construction, but does not achieve the optimal performance obtained by query construction for minimum generalization error. For the case of the linear perceptron, we have observed that if the learning algorithm is ill-matched to the details of the learning problem at hand (although the rule was still assumed to be perfectly learnable), minimum entropy queries can actually lead to a higher generalization error than random examples, but only if the teacher is very noisy, the learning algorithm is over-confident (*i.e.* underestimates the noise level) and the number of training examples is so low that the rule is only just beginning to be learnt.

In section V, we have considered the performance of isolated queries, *i.e.* queries which follow a training set of random examples, and compared them to single queries in a query sequence and single random examples. We have observed in our two example learning scenarios that for large $\alpha$ an isolated query leads to a greater (absolute) reduction in generalization error than a query in a query sequence and speculate that this result, as well as its analogue for the relative reduction in generalization error, might hold more generally. We have also investigated how much one could improve on the approach we have adopted in this paper, namely locally optimal query construction, *i.e.* 'greedy' optimization of the objective function at each step, by allowing global optimization of the query construction algorithm for a fixed total number of queries. We have found that the two methods will not in general be equivalent, but we expect from the results for our two example systems that the difference in performance will be small for many learning problems, especially for large numbers of training examples.

It should be clear from the above that much remains to be done in the field of query learning. In particular, more complicated rules need to be analysed and scenarios with unlearnable rules considered. Also the extension to classification problems where an entropic cost function [34–36] might be a more appropriate performance measure than the generalization error would be desirable.

In conclusion, we would like to stress that the emphasis of our work was not on finding practical query construction algorithms, but rather on exploring some of the more basic capabilities and limitations of query construction. In particular, our framework allows different characteristics of teacher and student space and thus makes an analysis of ill-matched learning algorithms and unlearnable rules possible, in contrast to, for example, a Bayesian approach. The drawback of this method is that in principle the query construction algorithms that we derive are influenced by our assumed knowledge about the teacher space, making them unlikely candidates for real-world applications. Of course this is nothing new—Bayesian analysis, for examples, makes even more stringent assumptions on the structure of the teacher space by assuming it to be equivalent to the student space. The search for query selection methods and corresponding objective functions which are robust against uncertainties in the structure of the teacher space is still open.

## ACKNOWLEDGEMENTS

I would like to thank D J Wallace and D Saad for helpful discussions and careful reading of the manuscript.

---

[1] T. L. H. Watkin, A. Rau, and M. Biehl, Reviews of Modern Physics **65**, 499 (1993).
[2] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, California, 1991).
[3] E. Baum, IEEE Transactions on Neural Networks **2**, 5 (1991).
[4] J.-N. Hwang, J. J. Choi, S. Oh, and R. Marks II, IEEE Transactions on Neural Networks **2**, 131 (1991).
[5] W. Kinzel and P. Rujan, Europhysics Letters **13**, 473 (1990).
[6] T. L. H. Watkin and A. Rau, Journal of Physics A **25**, 113 (1992).
[7] D. J. C. MacKay, Neural Computation **4**, 590 (1992).
[8] H. S. Seung, M. Opper, and H. Sompolinsky, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (ACM, New York, 1992), pp. 287–294.
[9] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby (unpublished).
[10] O. Kinouchi and N. Caticha, Journal of Physics A **25**, 6243 (1992).
[11] V. V. Fedorov, *Theory of Optimal Experiments* (Academic Press, New York, 1972).
[12] S. D. Silvey, *Optimal design* (Chapman and Hall, London, 1980).
[13] A. C. Atkinson, International Statistical Review **50**, 161 (1982).
[14] I. Ford, D. M. Titterington, and C. F. J. Wu, Biometrika **72**, 545 (1985).
[15] J. Berger and L. M. Berliner, Annals of Statistics **14**, 461 (1986).
[16] I. Ford, D. M. Titterington, and C. P. Kitsos, Technometrics **31**, 49 (1989).
[17] P. Chaudhuri and P. A. Mykland, Journal of the American Statistical Association **88**, 538 (1993).
[18] J. Pilz, *Bayesian Estimation and Experimental Design in Linear Regression Models*, 2nd ed. (John Wiley, Chichester, 1991).
[19] D. H. Wolpert, Complex Systems **6**, 47 (1992).
[20] As a notational shorthand, we assume that in all probability distributions in which $\Theta^{(p)}$ appears, the number of examples $p$ is held fixed, without writing this explicitly. Thus, for example, $P(\Theta^{(p)}|\mathcal{V})$ should strictly be written as $P(\Theta^{(p)}|\mathcal{V},p)$; hence it is normalized to one when integrating over all possible training sets of size $p$. To make this convention consistent with the use of Bayes' Theorem as in (2.2), we also make the natural assumption that the number of training examples is independent of the teacher rule that we are trying to learn. Thus, $P(p|\mathcal{V}) = P(p)$ and hence $P(\mathcal{V}|p) = P(\mathcal{V})$, so that we only need one a priori teacher distribution for all values of $p$.
[21] If there is a continuum of teachers $\mathcal{V}$, $P(\mathcal{V}|\Theta^{(p)})$ is a probability density which has the dimension of the inverse of $\mathcal{V}$. Strictly speaking, a dimensional normalizing constant is then necessary to make the argument of the logarithm in (2.3) dimensionless, but we shall not write this explicitly since it cancels from the entropy differences we will be concerned with.
[22] S. Amari and N. Murata, Neural Computation **5**, 140 (1993).
[23] E. Levin, N. Tishby, and S. A. Solla, Proceedings of the IEEE **78**, 1568 (1990).
[24] T. L. H. Watkin, Europhysics Letters **21**, 871 (1993).
[25] A. P. Dunmur and D. J. Wallace, Journal of Physics A **26**, 5767 (1993).
[26] The divergence as $T \to 0$ of the term $(N/2)\ln T$ in the student space entropy (4.15) does not present a problem here since we will only be concerned with entropy differences for which this term is irrelevant.
[27] A. Krogh and J. A. Hertz, Journal of Physics A **25**, 1135 (1992).
[28] Strictly speaking Krogh *et al.* [27] consider a Gaussian distribution for the inputs instead of the spherical distribution (4.2), but in the limit $N \to \infty$ these produce identical results, as can be checked by a direct calculation of the average eigenvalue spectrum of $\mathbf{M}_{\mathcal{V}}$ along the lines of [29].
[29] W. Kinzel and M. Opper, in *Models of Neural Networks*, edited by E. Domany, J. L. van Hemmen, and S. K (Springer, Berlin, 1991), pp. 149–171.
[30] W. Feller, *Introduction to Probability Theory and Its Applications*, 3rd ed. (John Wiley, New York, 1970), Vol. 1, (1st edition 1950).
[31] For finite but large $N$, this expression can be estimated to be valid for values of $\alpha$ much smaller than $\ln N$, from results for the mean waiting time in the 'collector's problem' (see *e.g.* [30]); this ensures that the relative decrease $(\alpha + O(1))/4N$ is always smaller than one as it has to be.
[32] M. H. DeGroot, Annals of Mathematical Statistics **33**, 404 (1962).
[33] J. Gladitz and J. Pilz, Math. Operationsforschung und Statistik, Series Statistics **13**, 491 (1982).
[34] J. J. Hopfield, Proc. Natl. Acad. Sci. USA **84**, 8429 (1987).
[35] S. A. Solla, E. Levin, and M. Fleisher, Complex Systems **2**, 625 (1988).
[36] J. S. Bridle, in *Neuro-Computing: algorithms, architectures and applications*, edited by F. Fougelman-Soulie and J. Hérault (Springer, Berlin, 1989), pp. 227–236.