Phase transition for parameter learning of hidden Markov models

Nikita Rau,[1] Jörg Lücke,[2] and Alexander K. Hartmann[1]

[1]*Institut für Physik, Universität Oldenburg, D-26111 Oldenburg, Germany*
[2]*Department of Medical Physics and Acoustics, Universität Oldenburg, D-26111 Oldenburg, Germany*

(Received 5 July 2021; accepted 15 September 2021; published 5 October 2021)

We study a phase transition in parameter learning of hidden Markov models (HMMs). We do this by generating sequences of observed symbols from given discrete HMMs with uniformly distributed transition probabilities and a noise level encoded in the output probabilities. We apply the Baum-Welch (BW) algorithm, an expectation-maximization algorithm from the field of machine learning. By using the BW algorithm we then try to estimate the parameters of each investigated realization of an HMM. We study HMMs with $n = 4,\ 8$, and 16 states. By changing the amount of accessible learning data and the noise level, we observe a phase-transition-like change in the performance of the learning algorithm. For bigger HMMs and more learning data, the learning behavior improves tremendously below a certain threshold in the noise strength. For a noise level above the threshold, learning is not possible. Furthermore, we use an overlap parameter applied to the results of a maximum *a posteriori* (Viterbi) algorithm to investigate the accuracy of the hidden state estimation around the phase transition.

DOI: 10.1103/PhysRevE.104.044105

I. INTRODUCTION

Phase transitions [1,2] are phenomena of central interest in physics and, in particular, statistical physics and thermodynamics. Classically, phase transitions are studied for actual physical system like liquid-vapor transitions of gases, ferromagnetic transitions of magnets, or the super conduction phase transition of metals. The behavior of phase transitions becomes more interesting if (quenched) disordered systems are studied. Well-known examples are the percolation transition, the spin glass-paramagnet transition of spin glasses, or the localization transition of disordered Bose systems. For decades, phase transitions in "nonphysical" systems have also been studied, e.g., the jamming transition in transport models like the Nagel-Schreckenberg model [3], the transition to an epidemic state in disease spreading [4], "easy-hard" phase transitions in optimization problems [5], or the transition to synchronicity of brain activity as described by the Kuramoto model [6]. Also, information-theoretic phase transitions with respect to analyzing (large) sets of data have become a field of interest, e.g., when finding communities in networks [7–10], analyzing the complexity of data generated by random systems [11], learning of patterns in neural networks [12], and detecting causality in Bayesian networks [13]. Many of these information-theoretic phase transitions seek to distinguish between phases where the desired information, can be obtained from the given data, and for phases where this is not possible.

Investigating these phase transitions allows one to understand the fundamental limitations of learning and extracting information from the data in general and in dependence of the used models and algorithms. This is a fundamental way to look at many problems and approaches which are considered in the field of *machine learning* [14,15]. It has become in recent years of major interest not only for ubiquitous applications but also for fundamental scientific studies. Note that, interestingly, machine-learning models like neural networks have been used also as tools to extract phase transitions in different system like Ising systems [16–18].

Nevertheless, in this work we are interested in the first mentioned connection between phase transitions and data analysis. In particular, we address the question whether there exists a transition between a phase where the fundamental parameters of a model can be extracted from the given data, and a phase where not. Specifically, we study the behavior of learning of parameters of elementary hidden Markov models (HMMs) [19,20] by computer simulations [21]. HMMs are widespread in data analysis and modeling, e.g., speech-recognition [19], biological sequence analysis [22], or analysis of gestures [23].

HMMs have been used often as tools, also in physical contexts, e.g., to treat data in experimental physics [24] or analyze phase transitions in physical systems [25]. However, they have, to the best of our knowledge, only rarely been the object of interest in a physical study, in particular, with respect to phase transitions occurring in the HMMs. For example, the entropy of a binary HMM was calculated [26] by a mapping to a one-dimensional Ising model. Lathouwers and Bechhoefer have investigated [27] transitions with respect to whether the reconstruction of a hidden sequence is possible or not depending on whether data can be kept in memory. Allahverdyan and Galstyan have investigated [28,29] the maximum *a posteriori* (Viterbi) sequence as a function of a noise parameter and found transitions between regions where almost full sequence reconstruction is possible and regions where it is not.

In contrast to these previous works, as mentioned, we are not interested in analyzing the properties or performance of a given HMM, with known parameters, with respect to the given data, but we are interested whether it is possible to learn the unknown parameters of an HMM from the given data. Specifically, we will numerically generate data for an HMM with given "ground-truth" parameter set. We control some noise via the emission probabilities. Subsequently we try to learn the parameters again using the BW algorithm [20,30]. We analyze the learning of the transition and emission probabilities specifying an HMM. We are interested whether there exists a sharp transition between a "learning phase" and a phase where the determination of the parameters fails. As we will see below, this is indeed the case.

The remainder of the paper is organized as follows: In Sec. II, we present the definition of an HMM and state the ensemble of random HMMs we have used. In Sec. III, we explain the algorithms we applied to simulate HMMs, to calculate posterior probabilities and to learn the parameters from the data. In Sec. IV, we define the measurable quantities we have recorded. In Sec. V, we present our simulation results. We finish with a summary and discussion in Sec. VI.

II. DEFINITIONS

Here we present the definitions we use in the present work, in particular, of the hidden Markov model. HMMs consist of a finite set of n (hidden) states and a finite or infinite set of emission symbols. A chain generated by an HMM starts in some initial state, which is randomly chosen with probabilities given by a vector $\overrightarrow{A^0} = (A_1^0, \dots A_n^0)$. Transitions between states i, j occur at discrete steps with probabilities A_{ij}, which is the probability to go into state j in the next step if the HMM is in state i in the current step. These probabilities are collected in an $n \times n$ transition matrix $\mathbf{A}$. Since the probability to be in a certain state depends only on the previous state, the sequence of states, denoted by $\overrightarrow{x} = (x_1, \dots, x_L)$ (L: length of sequence), forms a Markov chain. Nevertheless, the states are hidden, i.e., cannot be observed. Instead, at each state a randomly draw symbol is emitted, creating a sequence $\overrightarrow{y} = (y_1, \dots, y_L)$. Here we consider the discrete case where each time one symbol from an m letter alphabet is emitted. Let B_{ik} denote the probability to emit symbol k in state i. These probabilities are collected in the $n \times m$ matrix $\mathbf{B}$. The regular conditions for probabilities apply, i.e., all entries are nonnegative and the entries are normalized:

$$\begin{aligned} &\sum_{i=1}^{n} A_i^0 = 1, \\ &\sum_{j=1}^{n} A_{ij} = 1 \; \forall i \in \{1, ..., n\}, \\ &\sum_{k=1}^{m} B_{ik} = 1 \; \forall i \in \{1, ..., n\}. \end{aligned} \tag{1}$$

Here and in the following, we will always use letter i, j to indicate states and the letter k to indicate an emission symbol. In summary, each HMM is characterized by the set of parameters $\theta = (\overrightarrow{A^0}, \mathbf{A}, \mathbf{B})$. In this work, the HMMs are chosen in a way that there are as many emission symbols as states, i.e., $m = n$. Furthermore, $\mathbf{A}$ and $\mathbf{B}$ are chosen to have a specific structure. For the transition matrices, we consider a *full* ensemble of quenched disorder matrices which all have the form

$$A_{ij} = \begin{cases} p_T & \text{if } i = j, \\ \frac{1-p_T}{n-1} & \text{otherwise.} \end{cases} \tag{2}$$

Furthermore, we consider more sparse ensembles. For the *symmetric* ensemble, we allow only two transitions out of any state to its two neighbouring states.

$$A_{ij} = \begin{cases} p_T & \text{if } i = j, \\ \frac{1-p_T}{2} & \text{if } |i-j| = 1 \mod n, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Third, the *asymmetric* ensemble, only a transition to one neigbouring state is possible:

$$A_{ij} = \begin{cases} p_T & \text{if } i = j, \\ 1 - p_T & \text{if } i - j = 1 \mod n, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

For each disorder realization matrix we usually draw, if not mentioned otherwise, p_T uniformly from the interval [0.85,1]. Thus, for each matrix p_T is the probability for remaining in the current hidden state. For big values of p_T, the transition into other hidden states different from the current state is less probable, i.e., the hidden state chain will exhibit less fluctuations. We have restricted the values to $p_T \geqslant 0.85$ to reduce the fluctuations which makes our simulations less demanding in terms of statistics. Nevertheless, for some test cases, we also allow smaller values of p_T which leads to harder learning problems. Together with small data sets it may even turn out to be impossible to provide reasonable estimates of HMM parameters through learning, as we elaborate in the results section.

Furthermore, for each disorder realization the vector of initial-state probabilities consist of a set of $U(0, 1)$ uniformly drawn random numbers. We require that the sum of these numbers is normalized to one, i.e.,

$$A_i^0 = r_i / \sum_{i=1}^{n} r_i \quad \text{with } r_i \sim U(0, 1). \tag{5}$$

Here one could also consider other choices. However, a nonuniform distribution would presumably be even slightly easier to estimate. As the vector $\vec{A}^0$ only makes up a small subset of the parameters Θ, no strong impact of the precise choice of the initialization can be expected, though. Thus, we do not consider variants here.

The emission matrices have the form

$$B_{ik} = \begin{cases} p_{\mathrm{E}} & \text{if } i = k, \\ \frac{1-p_{\mathrm{E}}}{m-1} & \text{otherwise,} \end{cases} \tag{6}$$

where $p_{\mathrm{E}} \in [1/m, 1]$ is a fixed (external) parameter that controls the output noise level of the HMM. The case when p_{E} has the value 1 corresponds to an HMM with no noise at all. In this case each emission symbol of a hidden state corresponds to the hidden state itself and no other symbols are emitted. The lower bound $p_{\mathrm{E}} = \frac{1}{m}$ represents an HMM with a maximum noise level. In this case columns of $\mathbf{B}$ are all the same, therefore the hidden states can not be distinguished by their emission probabilities.

For each given HMM, we generate Markov chains of states and corresponding sequences of emitted symbols. Note that for each HMM, and correspondingly each sequence, the parameters actually used are called *ground-truth* parameters. The aim of our work is to see how well the learned parameters agree with the ground truth, see below for our measurable quantities. All results will be averages over a suitable number of matrices drawn from this ensemble. Note that the ensembles we study are inspired by the nondisordered, i.e., fixed matrices which were considered previously for the smallest possible case of $n = 2$ states [28] and for larger systems [27].

For all our work we consider different values of p_{E}, but all averages over different transition matrices $\mathbf{A}$ will be performed for fixed values of p_{E}, i.e., the same matrix $\mathbf{B}$. All results are then analyzed as a function of the parameter value p_{E}. We expect that in the limit $p_{\mathrm{E}} \to 1$ it will be much easier for any algorithm to learn the parameters from the sequence of visible symbols, while for the limit $p_{\mathrm{E}} \to \frac{1}{m}$ it will become impossible. In particular, we are interested in whether between these limiting values there exists a transition from an "easy" learning phase, at large values of p_{E} to a "hard" learning phase for small values of p_{E}.

III. ALGORITHMS

In this work, the parameter learning is executed by the Baum-Welch algorithm [30]. The algorithm is an expectation-maximization algorithm which seeks parameters θ^* that maximize the data likelihood $P(\overrightarrow{y}^1, ..., \overrightarrow{y}^N|\theta)$ of a given training data set $\{\overrightarrow{y}^1, \dots, \overrightarrow{y}^N\}$, i.e.,

$$\theta^* = \mathrm{argmax}_\theta P(\overrightarrow{y}^1, ..., \overrightarrow{y}^N|\theta). \tag{7}$$

Given an HMM model and initial parameters Θ, the training data set can be considered the input to the algorithm, and the parameters Θ^* can be considered the output. The BW algorithm is, like EM algorithms in general, an iterative procedure that can converge to local optima of the data likelihood. The BW algorithm is the very standard choice and preferable, e.g., to Viterbi training [20], because the latter one does not necessarily improve the likelihood in each iteration.

For comprehensiveness, we outline the algorithm here, details can be found in the literature [20]. One starts with first-guess initializations $\theta = (\overrightarrow{A^0}, \mathbf{A}, \mathbf{B})$, which are uniformly drawn here within the members of the ensemble, respectively, i.e., obeying the same constraints like normalization, etc. In each iteration, the algorithm proceeds in two steps which will be presented in more detail in the following sections: In the *expectation step*, the E-step, the BW algorithm calculates the expected times of transitions between two hidden states, the expected times of symbol emissions by hidden states and the expected number of times a sequence starts with a certain hidden state. Based on these calculations in the *maximization step*, the M-step, the new parameters are calculated. The algorithm guarantees a step-wise decrease of the Kullback-Leibler distance between the probability distributions over symbol sequences of the data and the model [31], i.e., the data likelihood increases monotonously.

A. E step

This step requires a sum over all hidden paths $\overrightarrow{x} = (x_1, \dots, x_L)$ which are compatible with one given observation $\overrightarrow{y}$ (taken from $\overrightarrow{y}^1, ..., \overrightarrow{y}^N$). For this purpose, so-called forward-variables $f_i(l)$ and backward-variables $b_i(l)$ in Eq. (8) are calculated:

$$\begin{aligned} f_i(l) &= P(y_1, ..., y_l, x_l = i), \text{ with } l \in \{1, ..., L\}, \\ b_i(l) &= P(y_{l+1}, ..., y_L|x_l = i), \text{ with } l \in \{1, ..., L-1\}. \end{aligned} \tag{8}$$

The forward variable $f_i(l)$ describes the joint probability that the hidden state i occurs at the lth position of a sequence and the first l observations were emitted. The backward variable $b_i(l)$ expresses the conditional probability that the last $L-l$ observations occur conditioned on the lth hidden state x_l being i [20]. There are recursive calculation rules that enable one to get forward and backward variables for every position within a sequence, see [19]. Combining the product rule for probabilities $P(X,Y) = P(X|Y)P(Y)$ with the definitions of $\mathbf{A}$, $\mathbf{B}$ and Eq. (8), one obtains [20]

$$P(x_l = i, x_{l+1} = j|\overrightarrow{y}, \theta) = \frac{f_i(l) A_{ij} B_{jy_{l+1}} b_j(l+1)}{P(\overrightarrow{y})}. \tag{9}$$

Equation (9) represents the conditional probability for getting the two consecutive hidden states i and j at the positions l and $l+1$ under the condition that the whole observation sequence $\overrightarrow{y}$ is known. $P(\overrightarrow{y})$ can be calculated by using the Forward-variables for $l = L$ as

$$P(\overrightarrow{y}) = \sum_{i=1}^{n} f_i(L). \tag{10}$$

Using Eqs. (9) and (10) and by averaging over the data set expected counts (denoted by an over bar) for the transition, emission and initial-state probabilities can be obtained as

$$\overline{A_{ij}} = \sum_{n=1}^{N} \frac{1}{P(\overrightarrow{y}^n)} \sum_{l=1}^{L-1} f_i^n(l) A_{ij} B_{jy^n_{l+1}} b_j^n(l+1), \tag{11}$$

$$\overline{B_{ik}} = \sum_{n=1}^{N} \frac{1}{P(\overrightarrow{y}^n)} \sum_{\{l=1|y^n_l=k\}}^{L} f_i^n(l) b_i^n(l), \tag{12}$$

$$\overline{A_i^0} = \sum_{n=1}^{N} \frac{1}{P(\overrightarrow{y}^n)} A_i^0 B_{iy_1^n} b_i^n(1). \tag{13}$$

B. M step

In the M step, the parameters are updated similar to the case of maximum-likelihood estimation for Gaussian distributions [20]. Here, the parameter updates are given by normalizing the expected counts Eqs. (11)–(13):

$$A_{ij}^{\text{new}} = \frac{\overline{A_{ij}}}{\sum_{j'=1}^{n} \overline{A_{ij'}}}, \tag{14}$$

$$B_{ik}^{\text{new}} = \frac{\overline{B_{ik}}}{\sum_{i'=1}^{m} \overline{B_{i'k}}}, \tag{15}$$

$$\left(A_i^0\right)^{\text{new}} = \frac{\overline{A_i^0}}{\sum_{i'=1}^{n} \overline{A_{i'}^0}}. \tag{16}$$

E and M steps are repeated until convergence to a, possibly local, likelihood optimum. In this work for convergence we consider the relative change of the data likelihood $P(\overrightarrow{y}^1, ..., \overrightarrow{y}^N|\theta)$ before and after the parameter update from Eqs. (14)–(16). When the relative change is smaller than a threshold ϵ, the BW algorithm is terminated.

As we will see below, to which set of parameters the BW algorithm converges depends on the choice of the initial parameter set. Therefore, as we will detail below, we use BW with ten random restarts and select from the 10 outcomes the "best" one, i.e., that one with highest posterior probability.

C. Viterbi algorithm

For some of our simulations, we also computed the maximum *a posteriori* (MAP) path, i.e., the (hidden) path $\overrightarrow{x^*}$ of states that maximizes for each observation $\overrightarrow{y}$ and given HMM parameters θ the path probability $P(\overrightarrow{x^*}|\overrightarrow{y}, \theta)$. This can be done by the Viterbi algorithm [32]. Similar to the forward-backward algorithm, it computes iteratively the Viterbi-variable $v_i(l)$. It is describing the probability of the most probable l steps path conditioned to it ends in state l and conditioned to the first l letters of the observed sequence. The hidden state sequence $\overrightarrow{x^*}$ can be obtained by backtraceing.

IV. SETUP, PARAMETERS, AND MEASURABLE QUANTITIES

We applied the BW algorithm to ensembles of HMMs, as described by Eqs. (2)–(6), for three different HMM sizes $n \in \{4; 8; 16\}$. We have tested several values for the convergence parameter ϵ. We show results for $\epsilon = 10^{-7}$ because for higher values the convergence was a bit worse and for even smaller values the results do not change substantially. To see the influence of an increase of the available data, we have performed all numerical experiments for six different sizes (N, L) of the learning sets, for each HMM size, respectively.

The convergence of the BW algorithm depends on the initial parameter set. Thus, we have, for each given realization of an HMM under consideration, run the BW algorithm 10 times with independently drawn initial parameters, each resulting in a locally optimum estimate θ^*_r $(r = 1, \dots, 10)$. To select the best parameter set $\tilde{\theta}$ among the 10 outcomes of the 10 runs, we have used, to avoid over-fitting effects, a second data set $\overrightarrow{z}^1, ..., \overrightarrow{z}^N$ always of the size $N = 200, L = 100$. Note that in practical applications one can always split the available data into two halves. The final best estimate θ' is the one that exhibits the maximum joint data probability $P(\overrightarrow{z}^1, ..., \overrightarrow{z}^N|\theta^*_r)$ $(r = 1, \dots, 10)$. For practical reasons, we consider log likelihood when possible, as usual. For technical convenience, when we add up probabilities, we always first scale them by the smallest probability, then perform the sum, and then rescale, as it is often done [20].

All results presented below, for each considered value of $p_{\rm E}$, we have performed an average over different realizations from the ensemble of HMMs. For $n = 4$, we considered 1000 realizations, for $n = 8$ we studied 600 realizations, and for $n = 16$ we found 200 realizations to be sufficient.

Note that during the learning process it is assumed that the generating HMM-parameters and the hidden state sequences $\overrightarrow{x}^1, ..., \overrightarrow{x}^N$ of the learning data are unknown. Since we use artificially generated data they are nevertheless available to us and we can use them as ground truth for comparison and evaluation of the learning process. For our purposes, we measure the total error $E_{\rm tot}$ [Eq. (17)], which is the sum of the absolute differences between actual and estimated parameters:

$$E_{\rm tot} = \sum_{i=1}^{n} \left|A_i^0 - \tilde{A}_i^0\right| + \sum_{i=1}^{n}\sum_{j=1}^{n} |A_{ij} - \tilde{A}_{ij}| + \sum_{i=1}^{m}\sum_{k=1}^{n} |B_{ik} - \tilde{B}_{ik}|. \quad (17)$$

This quantity consists of differences for probabilities where certain probabilities sum up to one due to the normalization. For two sets $\{C_1, \dots, C_l\}$ and $\{\tilde{C}_1, \dots, \tilde{C}_l\}$ of probabilities with $\sum_{i=1}^{l} C_i = 1$ and $\sum_{i=1}^{l} \tilde{C}_i = 1$, the sum $\sum_i |C_i - \tilde{C}_i|$ can at most attain the value 2, for example, for the case $C_1 = 1, C_i = 0$ $(i > 1)$ and $\tilde{C}_2 = 1, \tilde{C}_i = 0$ $(i \neq 2)$. If, e.g., one reduces C_1 a bit, $C_1 = C_1 - \delta$ and adds this, e.g., at $C_3 = C_3 + \delta$, then the total sum of the differences will be still 2. Thus, one cannot exceed 2. Hence, the error $E_{\rm tot}$ can at most reach $2(1 + n + m)$, i.e., is, in particular, linear in the number of states for $n = m$ with $E_{\rm tot} \leqslant 4n + 2$. In general, also estimated parameters very different from the ground-truth parameters can make up a successful run (e.g., in degenerated cases when a model's likelihood is invariant under certain parameter permutations). In our case and for the way we choose the generating parameters in Eqs. (2)–(6), we found Eq. (17) to measure the degree of success of a given run usally well. Nevertheless, it may happen that the BW algorithm permutes the states. For example, it may attribute the typical, i.e., high-probability, output "1" to state 3, output "2" to state 4, output "3" to state 1, and output "4" to state 2. Thus, the algorithm will obtain corresponding transition permuted rates, e.g., the estimated transition probability for A_{34} corresponds actually to the transition between states 1 and 2. Hence, even if all parameters are estimated very well, a one-to-one comparison using Eq. (17) will indicate a large error, due to this "misnaming" of the states. To avoid this, for calculating the final error, we always take the minimum error over all permutations of the states.

Another way to test the estimated HMMs is to obtain, for each training sequence $\overrightarrow{y}^n$ $(n = 1, \dots, N)$, the most-likely hidden path $x_1^{*n}, \dots x_L^{*n}$ by applying the Viterbi algorithm to an HMM with the estimated parameters $\tilde{\theta}$. This can be compared to the actual paths $x_1^i, \dots x_L^i$. The fraction of agreeing hidden states is given by the so-called overlap q, which is a frequently used quantity in the physics of disordered systems:

$$q = \frac{\sum_{n=1}^{N}\sum_{l=1}^{L} \delta\left(x_l^n, x_l^{*n}\right)}{N L}. \quad (18)$$

Here $\delta(x_j^n, x_j^{*n}) = 1$ if the hidden state of the learning set x_j^i is equal to the hidden state of the Viterbi sequence x_j^{*i} and otherwise zero. This means $q \in [0; 1]$ with $q = 1$ corresponds to a 100% reconstruction of the hidden sequences $\overrightarrow{x}^1, ..., \overrightarrow{x}^N$. Note that $q = 1$ is not common because even when using the true parameters of an HMM, the Viterbi path is only the most-likely one, but very often not the actually generated one.

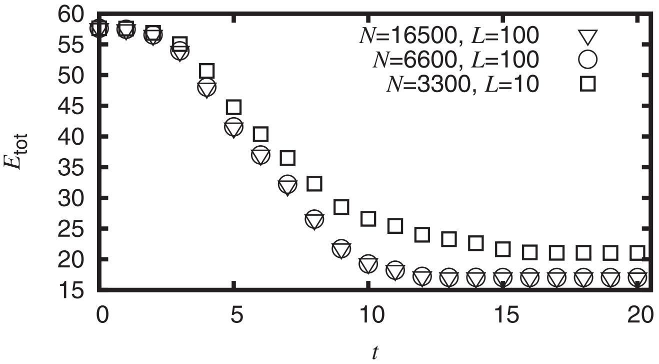

FIG. 1. The evolution of the total error E_{tot} as a function of the BW step for an HMM with $n = m = 16$ and three differently data set sizes, but the same initial parameters. Note that the initial value $E_{\text{tot}}(t = 0)$ is consistent with the upper bound $4n + 2 = 66$.

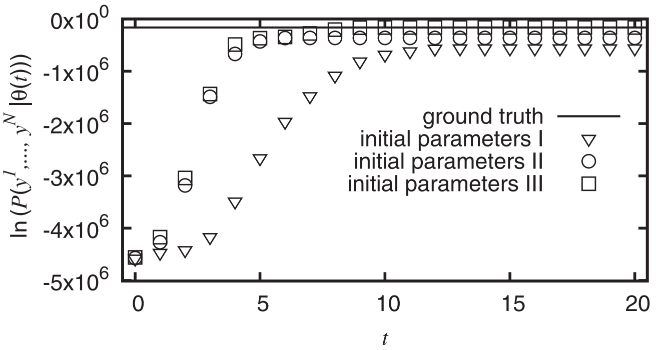

FIG. 2. The evolution of the log-likelihood as a function of the BW step for an HMM of size $n = m = 16$ with three differently initial parameter sets θ^{I}, θ^{II}, and θ^{III} as starting point of the BW algorithm. The log-likelihood was calculated for the learning set. The solid line shows the ground truth. The corresponding total errors after leveling off are: $E_{\text{tot,I}} = 17.005$, $E_{\text{tot,II}} = 4.362$, and $E_{\text{tot,III}} = 0.050$.

V. RESULTS

First, we study the behavior of the BW algorithm. In Fig. 1 the evolution of the total error E_{tot} is shown for one realization of the *full* ensemble as a function of the step t of the BW algorithm. We consider three different learning data set sizes, but in all three cases with the same set of starting parameters θ where each parameter was drawn uniformly from [0,1]. Initially, the parameter set is very different $E_{\text{tot}} \approx 57$ from the ground-truth parameters of the original HMM, but during its evolution, the error is decreased until it levels off at parameter values still different from the ground-truth ones, i.e., $E_{\text{tot}} \gg 0$. One also can see that for larger data set sizes, the error is decreased more and faster, but once a certain size is reached, no more improvement is obtained. Below we will see that the combined size NL acts like a system size in the theory of phase transitions, which allows us to extrapolate the phase transition point from “hard” to the “easy” learning phase.

Since for the previous example the algorithm was not able to recover the ground-truth parameters, we next study what influence the initial parameter set has. For three different random initial parameter sets the log-likelihood $\ln\{P[\overrightarrow{y}^{1}, ..., \overrightarrow{y}^{N}|\theta(t)]\}$ of the learning data for the current parameter set $\theta(t)$ is shown as function of the iteration t. One can observe in Fig. 2 that initially the growth in log-likelihood is fast, similar to the improvement seen for E_{tot} in Fig. 1. After some iterations also these values level more or less off. One sees that indeed for different initial parameters different final log-likelihoods are reached. The same is true for the error (not shown here). This illustrates the usefulness of repeated BW runs, from which the one with the highest log-likelihood for the test data set is chosen. Therefore, the influence of “unfortunate” choices for the initial parameters is reduced. Actually for case (III) the log-likelihood is nearly equal to the ground truth, which was used to generate the data. This is an indication that indeed a very good estimate of the parameters was obtained [33], which is supported by the fact that for this case the error in parameters is $E_{\text{tot,III}} = 0.050$, i.e., very small.

A. Total error of the full ensemble

In Fig. 3 the total error E_{tot} is shown as a function of rescaled probability $p_{\text{E}}m$ for $m = n = 4$ and six different learning set sizes (N, L). It is visible that for each learning set size the largest E_{tot}-value, i.e., the worst results, can be found for $p_{\text{E}}m = 1 \leftrightarrow p_{\text{E}} = 0.25$. This meets the expectations since $p_{\text{E}} = 0.25$ corresponds to the largest noise level. For increasing $p_{\text{E}}m$, the total error decreases and reaches a minimum for $p_{\text{E}} \to 1$. This corresponds to a nonexisting noise level, where the full information about the states can be obtained from the data. Hence, Fig. 3 shows clearly that the parameter-learning improves if the noise level is decreased. Furthermore, it is visible that the learning gets better by increasing the size of the learning data set. The behavior for $N = 1125$ and $L = 100$ shows that the biggest improvement occurs roughly in the interval $p_{\text{E}}m \in [1.3; 1.4]$. The very steep decrease of the curve indicates a sharp change from a nonlearning to a learning behavior, i.e., a phase transition in the information theoretic sense. A similar drop of E_{tot} is observable for the learning sets $N = 450$, $L = 100$ and $N = 225$, $L = 100$, which shows that the learning set sizes are large enough to observe the limiting

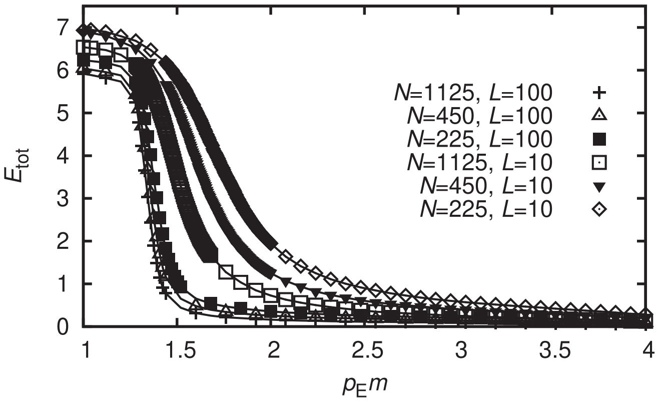

FIG. 3. The total error E_{tot} as a function of the noise parameter p_{E}, multiplied by the number of symbols $m = 4$ for six different learning set sizes. Each data point is the average result over 1000 simulations. The error bars are smaller than symbol size. For the three largest learning set sizes, the curves differ only slightly, which indicates that the results for the thermodynamic limit will look similar.

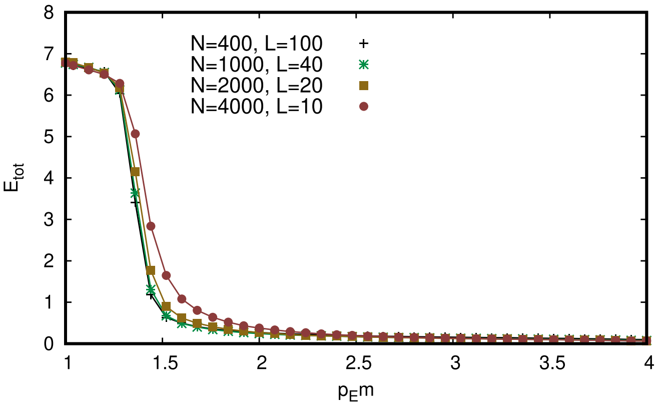

FIG. 4. The total error E_{tot} as a function of the noise parameter p_{E}, multiplied by the number of symbols $m = 4$ for a large learning set $NL = 40\,000$ but different values of L and N. Each data point is the average result over 1000 simulations. The error bars are smaller than symbol size.

behavior. The three smallest learning sets show a less steeper decrease, indicating stronger finite-size effects. It can also be seen that the decline shifts to the left with increasing size of the learning data set, which we will below use to determine a phase-transition point.

To verify whether the choice of the learning-set size as system size is justified, we have performed additional simulations for a fixed value of $NL = 40\,000$ while varying both values. As visible in Fig. 4, the results for different partitions (N, L) of the learning set do not differ much. Only the case $L = 10$ of very short runs is just slightly harder to estimate. This is probably because initial states provide less information, in particular, for the present uniform distribution. Thus, if the fraction of initial states in the data set is high, then the error for estimating the transition $\{A_{ij}\}$ parameters will also be higher. Only estimating the initial parameters $\{A_i^0\}$ will be more accurate, but this somehow is outweighed by the fact that there are more transition parameters than parameters for the initial states. Anyway, the difference for the $L = 10$ case is really small. We have obtained a very similar result for $NL = 10\,000$ (not shown here). We conclude our data for $L = 100$ should represent well the limit of "large enough" length of the individual runs, but for completeness, we also show below sometimes results for $L = 10$ as well.

We have studied the behavior of the total error also for other HMM sizes. Figures 5 and 6 show E_{tot} for $m = n = 8$ and $m = n = 16$. For each system size, the parameter learning behaves qualitatively the same as for $n = m = 4$. But one observes that in comparison to the case $n = m = 4$ the decrease as a function of $p_{\text{E}}m$ seems to become even steeper and its position shifts slightly to larger parameter values, i.e., away from the point $p_{\text{E}}m = 1$ of no information. This means, the size of the no-learning phase becomes bigger on the rescaled p_{E} axis, indicating that for even larger HMM sizes the phase transition from no learning to learning will still persist.

B. Finite-size scaling for full ensemble

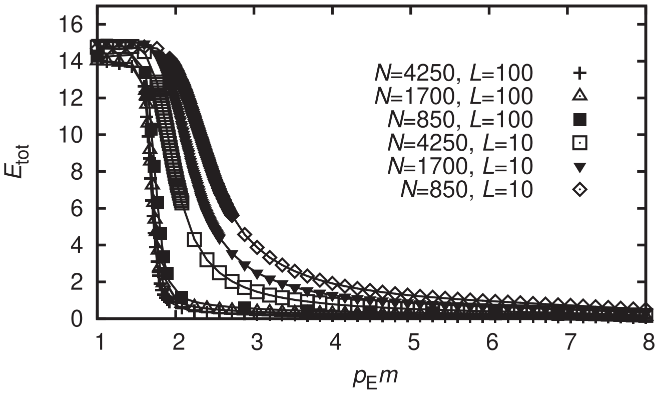

FIG. 5. The total error E_{tot} as a function of the noise parameter p_{E}, multiplied by the number of symbols $m = 8$ for a better comparison of the different HMM sizes and for six different learning set sizes. Each data point is the average result of 600 simulations. The error bars are smaller than symbol size. For the three largest learning set sizes, the curves differ only slightly, which indicates that the results for the thermodynamic limit will look similar.

As mentioned, a left shift of the decline of E_{tot} is observable for all HMM system sizes in Figs. 3–6 when increasing the learning data set sizes. This allows us to determine the phase transition point, which is the position of the steepest point of decline in the thermodynamic limit. To determine this we consider the variances σ^2 of the total error as a function of $p_{\text{E}}m$, which exhibits peaks at the points of steepest decrease of E_{tot}. To obtain estimates for the peak positions P_{peak} (on the $\tilde{p} = p_{\text{E}}m$ scale) we performed Gaussian fits to the peaks. Figure 7 shows the data used for the fits and the fit results for $n = 4$. A shift to the left upon increasing NL is clearly observable. In addition, the Gaussian fits, i.e., the transition regions, become narrower for larger learning data sets which is often observed in standard finite-size scaling theory [34]. By using the peak positions of all learning data sets, we extrapolate the dependence of $P_{\text{peak}}(N, L)$ to large learning sets. For that, we used the standard finite-size scaling power-law ansatz for the finite-size dependence of the phase transition

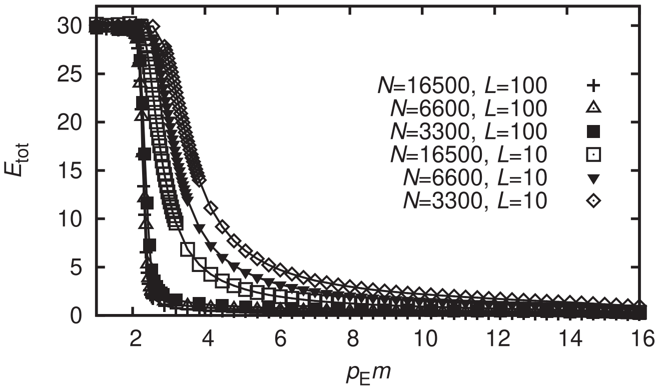

FIG. 6. The total error E_{tot} as a function of the noise parameter p_{E}, multiplied by the number of symbols $m = 16$ for six different learning set sizes. Each data point is the average result over 200 simulations. The error bars are smaller than symbol size. For the three largest learning set sizes, the curves differ only slightly, which indicates that the results for the thermodynamic limit will look similar.

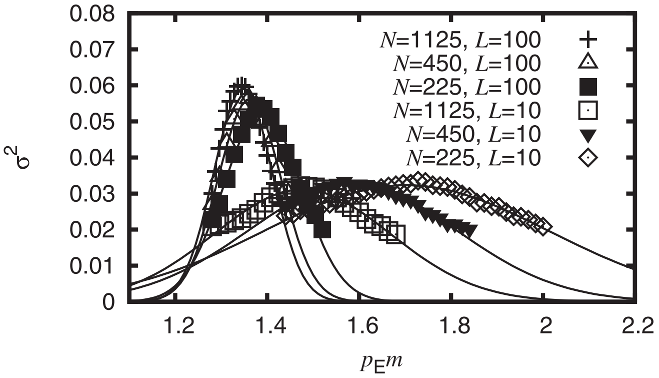

FIG. 7. The variance σ^2 of the total error E_{tot} for the system size $m = n = 4$ as function of $p_E m$, calculated by using the results of the 1000 HMM realizations. Data is shown only near the transition regions, respectively. The lines show the fits to the Gaussians, which where used to determine the peak positions.

positions for second order phase transitions:

$$P_{\text{peak}}(N, L) = \tilde{p}_\infty + a(NL)^{-1/\nu}. \tag{19}$$

Here, $\tilde{p}_\infty$ denotes the phase transition point in the thermodynamic limit $NL \to \infty$ and a is a nonuniversal fit parameter. ν denotes the exponent governing the finite-size corrections and describes in the standard theory of continuous phase transitions the growth of the correlations when approaching the phase transition point.

The data for the peak positions together with the fit according to Eq. (19) for the learning sets of the system size $n = 4$ is shown in Fig. 8. One can observe that the fits match reasonably well given the rather large fluctuations of the critical-point estimates. Such fluctuations notwithstanding, the overall behavior allows the phase transition to be well seen as a continuous phase transition. The resulting fit parameters, also for the other HMM sizes (which were analyzed in a similar way, not shown as figures), are collected in Table I. We observe that the rescaled critical point moves to the right with increasing HMM size $n = m$. The critical exponents ν carry rather large error bars, which is often the case when numerically studying phase transitions. Hence, whether they are the same for all HMM sizes, i.e., whether the hard-easy learning transition is universal with respect to HMM size, cannot be concluded from the data.

TABLE I. Rescaled critical points $\tilde{p}_\infty$ (second column) and critical exponents ν (third column) for the different HMM sizes m (1st column) as obtained from a fit to Eq. (19). For the largest HMM sizes the corrections to scaling were large such that the peak position for small data size was omitted from the fit.

m	$\tilde{p}_\infty$	ν
4	1.2(1)	2.5(10)
8	1.5(1)	2.3(7)
16	2.0(4)	2.1(14)

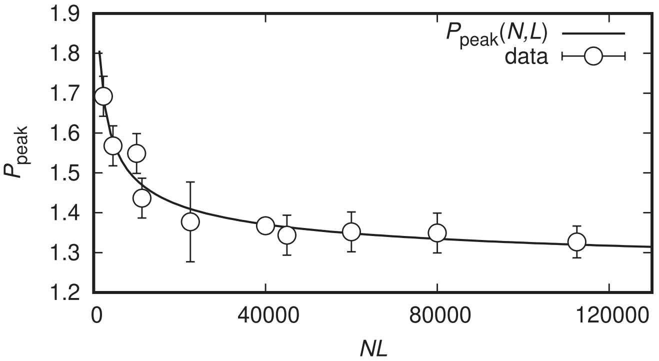

FIG. 8. Dependence of P_{peak} as a function of the size of learning data set, indicated by the product NL. The symbols show the positions of the steepest point of decline which were obtained by the peak positions of the Gaussian fits from Fig. 7. Error bars are at most of order symbol size. The line indicates the result of a fit to Eq. (19).

C. Changing range of diagonal entry p_T

Most of our results are for the case that the probability to stay in a state, i.e., the diagonal entries of the transition matrix **A**. is rather large, i.e., $p_T \in [0.85, 1]$. Here we consider two cases with smaller diagonal values. The smaller the diagonal values are, the faster the HMM will change state, thus, making estimates, in particular, of parameters, more difficult.

For the enlarged range $p_T \in [0.5, 1]$, the results do not look much different, see Fig. 9, only we observe that the error is slightly larger in the limit $p_E \to 1$ compared to the first case.

The result for very small probabilities to stay in a state, taken in a range [0.2,0.5], is shown in Fig. 10. Thus, the states in the Markov chain change extremely rapidly. Here, for small data set sizes NL the BW algorithm fails to find the true parameters basically everywhere. Only for higher amounts of learning data, the BW algorithm again can somehow be successful and one observes that the error E_{tot} somehow decreases when increasing p_E. Nevertheless, even for large values of p_E the full true parameters cannot be obtained exactly, only a pleateau with $E_{\text{tot}} \approx 1$ is reached. Only a trace of a true phase transition behavior is visible. Presumable only for an extreme size of the data set better results might be achievable, if at all, but this is beyond the scope of our study.

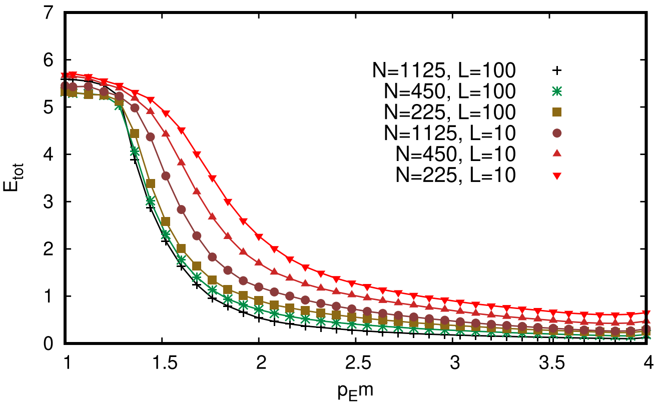

FIG. 9. The total error E_{tot} as a function of the scaled noise parameter $p_E m$ for $n = m = 4$ case of the *full* ensemble and a larger range $p_T \in [0.5, 1.0]$ of the diagonal transition matrix entries.

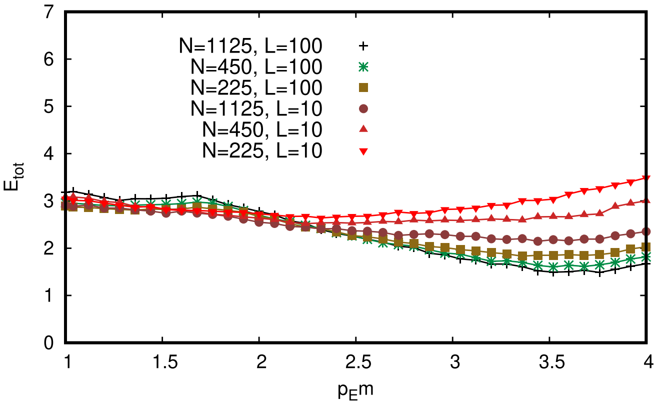

FIG. 10. The total error E_{tot} as a function of the scaled noise parameter $p_{\text{E}}m$ for $n = m = 4$ case of the *full* ensemble and a shifted range $p_{\text{T}} \in [0.2, 0.5]$ of the diagonal transition matrix entries.

Since for the medium values of p_{T} the transition near $p_{\text{E}}m \approx 1.4$ looks very similar to the above case $p_{\text{T}} \in [0.85, 1]$, we do not go into the details here and do not provide another finite-size scaling analysis.

D. Other ensembles

We have also studied two other ensembles of transitions matrices which are much sparser, the *symmetric* and the *asymmetric* ensemble as defined in Eqs. (3) and (4), respectively. Since we are interested only in the general behavior, we restrict the study to the case $m = n = 4$. This is justified, because, as visible in Fig. 11, the behavior is nearly indistinguishable from the *full* ensemble. Only the finite-size effects near $p_{\text{E}}m = 1$ are smaller here. This means the BW algorithm does not profit from the fact that most entries of the transition matrix are zero and the learning behavior is somehow insensitive to the structure of the matrix.

We have performed a finite-size scaling analysis in the same way as it was done for the *full* ensemble in Sec. V B.

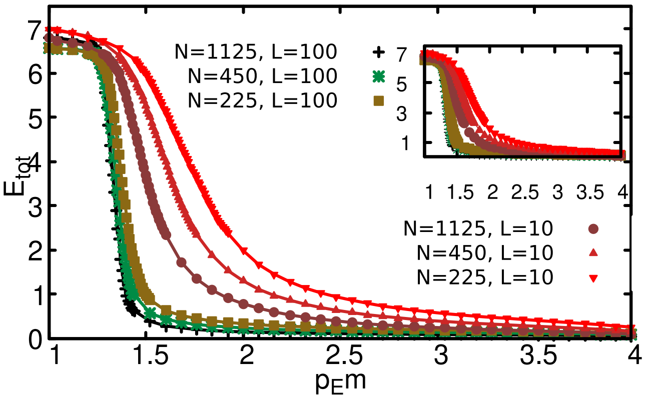

FIG. 11. The total error E_{tot} as a function of the scaled noise parameter $p_{\text{E}}m$. The main plot shows the *asymmetric* ensemble, while in the inset the *symmetric* ensemble is shown. Displayed are results for six different learning set sizes, respectively. Each data point is the average result over 1000 simulations. The error bars are smaller than symbol size.

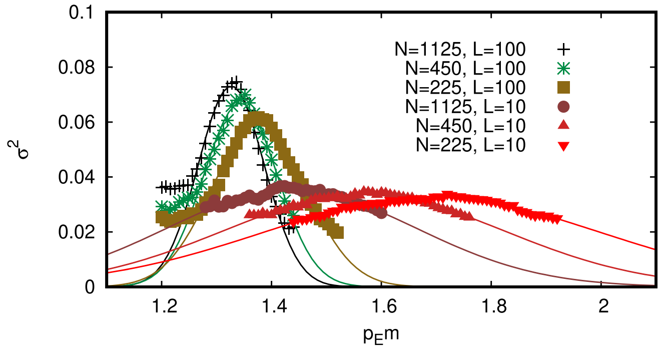

FIG. 12. The variance σ^2 of the total error E_{tot} for the *symmetric* ensemble of a system size $m = n = 4$ as function of $p_{\text{E}}m$, calculated by using the results of the 1000 HMM realizations. Data is shown only near the transition regions, respectively. The lines show the fits to the Gaussians, which where used to determine the peak positions.

Since the results look very similar, we only show in Fig. 12 as an example the bevhavior of the variance σ^2 of the total error E_{tot} for the *symmetric* ensemble as function of $p_{\text{E}}m$. We have also here performed finite-size scaling fits by fitting peak positions to Eq. (19). We obtained for the *symmetric* ensemble $\tilde{p}_\infty = 1.16(14)$ and $\nu = 3.1(1.1)$ For the asymmetric ensemble the values $\tilde{p}_\infty = 1.11(18)$ and $\nu = 3.6(1.4)$. Interestingly, the values are compatible with the values found for the $m = n = 4$ case of the *full* ensemble, but the error bars, in particular, for the critical exponent ν are quite large which prohibits an accurate comparison. The reason is probably that we have used, for convenience, a smaller number of different data sizes NL here. Nevertheless, the results indicate that the transition between the phase where the parameters can not be learned and the phase where it is possible is at least possibly *universal* in the statistical mechanics sense, i.e., does not depend on details of the model.

E. Overlap

The behavior of the overlap parameter q as a function of $p_{\text{E}}m$ for the *full* ensemble and $n = m = 8$ is shown in Fig. 13. When decreasing the noise, i.e., increasing $p_{\text{E}}m$, the estimated MAP paths become more and more similar to

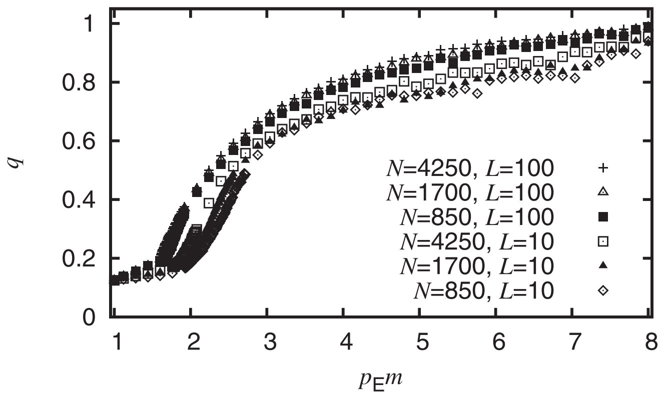

FIG. 13. The overlap parameter q for HMMs with the size $m = n = 8$ and the same learning sets as in Fig. 5.

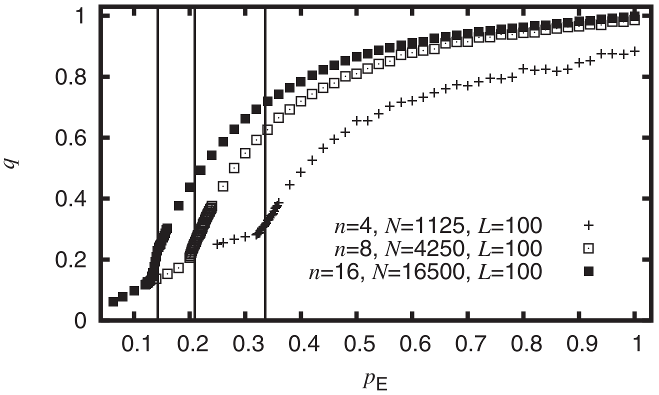

FIG. 14. The overlap parameter q for HMMs with the size $n = 4$; $n = 8; n = 16$ and the largest investigated learning set size as a function of the noise parameter p_E. p_E is shown for the interval $[\frac{1}{16} : 1]$ because its lower bound corresponds to the strongest noise level for HMMs with $m = n = 16$. The vertical lines indicate the extrapolated transition points obtained from the fit of Eq. (19), and suitably rescaled, i.e., $\tilde{p}_\infty/m$.

the actual (ground-truth) paths. Since even with the correct parameters estimating the MAP path often does not lead to the actual path, the behavior is very smooth. Interesting, near the phase transition point, the curve exhibits a strong kink, which indicates the phase transition is visible for q as well, but less clearly. Note that unlike to the behavior observed previously [28], there is no alternation between several sharp kinks and monotonously ascents of q over the whole range $p_E m \in [1 : 8]$. With respect to the finite-size effects of our results, it can be seen that for growing learning data set size the transition appears slightly sharper and occurs for smaller values of $p_E m$, i.e., exhibits the same principle finite-size behavior as the total error. The results of q for the two largest sizes are almost indistinguishable, thus can be taken to be very similar to the result for the thermodynamic limit $NL \to \infty$.

Thus, to compare the behavior for different HMM sizes, we take always the result obtained for the largest learning data set. In Fig. 14 a comparison of the overlap parameter q for the different sized HMMs is shown here as a function of p_E only, because in this way the different curves can be better distinguished. Note that for smallest HMM even the largest learning data set used is rather small, because this was sufficient to estimate the parameters with high accuracy, i.e., for a small value of E_{tot}. Nevertheless, here, for the overlap, this results in stronger fluctuations as compared to the larger HMMs. Anyway, one observes that the kinks for $q(p_E)$ indeed are very close to the extrapolated transition points (shown as vertical lines in the figure). Thus, the hard-to-easy learning phase transition is not only visible in the total error for the parameters but also in the underlying behavior of the HMMs, as exhibited by the MAP hidden paths.

VI. SUMMARY AND DISCUSSION

In this work we have not used HMMs as mere tools to analyze the data originating from physical and other systems. Here, we made HMMs the actual subject of interest with a physics perspective. This has already been done in a few previous papers, but we have here addressed a very different research question. We have analyzed an ensemble of elementary HMMs with n states and m output symbols with respect to learning HMM parameters from data. We have restricted ourselves to $m = n$. For learning we have used the Baum-Welch algorithm to estimate the maximum likelihood parameters. However, we believe that many aspects of our results also apply for other combinations of n and m and other algorithms for parameter estimation.

We have varied a noise parameter p_E which controls how much the visible output symbols convey information about the visited hidden states. In the limit of $p_E \to 1$ no noise exists and perfect learning is possible, while for $p_E \to 1/m$ the output is completely random and no learning is possible. From analyzing the error E_{tot} of the learned to the actually used parameters, from its variance and from the overlap parameter q, we obtain clear evidence for the existence of a nontrivial phase transition between a "hard" learning phase and a "easy" learning phase. Note that at $p_E = 1/m$ clearly no learning is possible at all. But one could expect that for any $p_E > 1/m$, if the amount of available data is only large enough, the algorithm could exploit the bias to finally get the true parameters. For restricting the number of restarts to 10, this is not the case, the phase transition point is clearly different from the trivial limit $1/m$. The transition seems to persist in the limit of large HMM sizes n, since the critical point moves even to the right on the $\tilde{p}_E \equiv p_E m$ scale with increasing HMM size $n = m$. Note that it is still possible that in the "hard" phase, the number of local minima is exponential in the number of states, thus, maybe there is a range of values of p_E where by using a very large number of restarts one can still find the ground-truth parameters. To investigate this issue we have, for $m = 8$ and the largest data sets available, performed some test runs in the "hard" region where we started the BW algorithm always with the ground-truth parameters. Indeed, close to the phase transition, the BW algorithm always stayed close to the ground-truth parameters, which means that here the phase is only "hard" but not "impossible." Nevertheless, close to $p_E = 1/m$, where the states of the HMM are indistinguishable, the BW algorithm always iterated away from the ground-truth parameters, thus, here learning is indeed "impossible."

When decreasing the diagonal entry p_T of the transition matrix, the problem to determine the parameters using the BW algorithm becomes harder, but if enough data is provided, the results look very similar to the case of p_T close to unity. Also for variants of the ensembles, when the transitions matrices become more sparse as for the *symmetric* and the *asymmetric* ensemble, the appearance of phase transitions was observed, with very similar curves.

From the finite-size dependence of the critical points, we have determined the critical exponent ν for the different HMM sizes. The value seems to be universal with respect to HMM size and near 2.3, but with a rather large error bar. Thus, there exists an information-theoretic phase transition in the learning of the investigated HMMs, similar to transitions observed for neural networks [12], community detection [7–10] or optimization algorithms [5]. Analyzing this phase transition for HMMs will allow for a better understanding of the limits of learning. For example, with more numerical effort, one could rerun the BW algorithm many times and study the distribution of local minima and investigate whether they tend to be very close to each other in parameter space. Or they could turn out to be organized hierarchically in clusters, similar to the "replica-symmetry breaking" of spin-glasses. Such hierarchical organizations were also found numerically in the solution landscape of optimization problems [35,36]. But such studies about the behavior of HMMs and parameter learning might also be useful for practitioners, to optimize algorithms and to get to know meaningful application ranges.

From a fundamental point of view, it would be certainly worthwhile to consider to use mathematical (mean-field) methods to analytically perform the disorder average and investigate such phase transitions occurring in HMMs more thoroughly, expanding previous work on two-state HMMs which were tackled by mapping it to Ising systems [28,29] or a direct analysis of the estimation probabilities [27]. Nevertheless, the previous works are for the case of the estimation of the most probable path, while the present work is for the estimation of the parameters, which is arguably a significantly more difficult problem to analyze even for much simpler algorithms, e.g., used in combinatorial optimization [37,38]. Thus, whether an analytical analysis is possible appears uncertain to us.

Clearly, we have analyzed only a small set of specific ensembles of HMMs. We expect, however, that many aspects of our results, in particular, the existence of one (or more) nontrivial phase transitions as observed here, hold more generally. Nevertheless, it would be certainly of interest to study other HMMs and other types of probabilistic data models to study and understand phase transitions of their learning algorithms. Also it would be worth investigating different types of parameter-estimation algorithms, e.g., to investigate how much the location of the phase transition depends on the algorithm itself. Nevertheless, due to universality often observed in physical systems, we expect critical exponents (ν in our case), describing the growth of correlations when approaching continuous phase transitions, to also be universal here. Furthermore, we expect that there are fundamental limits of learning, like those observed for community detection [9,10], where there is a phase which exhibits statistically significant differences but which provably cannot be exploited by any algorithm.

ACKNOWLEDGMENTS

The simulations were performed at the HPC cluster CARL, located at the University of Oldenburg (Germany) and funded by the DFG through its Major Research Instrumentation Program (INST 184/157-1 FUGG) and the Ministry of Science and Culture (MWK) of the Lower Saxony State.

[1] H. E. Stanley, *An Introduction to Phase Transitions and Critical Phenomena* (Oxford University Press, Oxford, 1971).
[2] J. M. Yeomans, *Statistical Mechanics of Phase Transitions* (Oxford Science Publications, Oxford, 2000).
[3] K. Nagel and M. Schreckenberg, J. Phys. I (France) **2**, 2221 (1992).
[4] M. E. J. Newman, Phys. Rev. E **66**, 016128 (2002).
[5] A. K. Hartmann and M. Weigt, *Phase Transitions in Combinatorial Optimization Problems* (Wiley-VCH, Weinheim, 2005).
[6] M. G. Kitzbichler, M. L. Smith, S. R. Christensen, and E. T. Bullmore, PLOS Comput. Biol. **5**, e1000314 (2009).
[7] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[8] P. Ronhovde and Z. Nussinov, Phys. Rev. E **81**, 046114 (2010).
[9] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. E **84**, 066106 (2011).
[10] D. Hu, P. Ronhovde, and Z. Nussinov, Philos. Mag. **92**, 406 (2012).
[11] O. Melchert and A. K. Hartmann, Phys. Rev. E **87**, 022107 (2013).
[12] H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992).
[13] A. K. Hartmann and G. Nuel, PLoS One **12**, e0170514 (2017).
[14] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).
[15] A. C. Faul, *A Concise Introduction to Machine Learning* (Chapman & Hall/CRC, Boca Raton, 2020).
[16] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[17] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[18] K. Kashiwa, Y. Kikuchi, and A. Tomiya, Prog. Theor. Exp. Phys. **2019**, 083A04 (2019).
[19] L. Rabiner, Proc. IEEE **77**, 257 (1989).
[20] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis* (Cambridge University Press, New York, 1998).
[21] A. K. Hartmann, *Big Practical Guide to Computer Simulations* (World Scientific, Singapore, 2015).
[22] K.-J. Won, A. Prügel-Bennett, and A. Krogh, Bioinformatics **20**, 3613 (2004).
[23] A. F. Wilson, A. D.; Bobick, Internat. J. Patt. Recog. Artif. Intell. **15**, 123 (2001).
[24] I. Kanter, A. Frydman, and A. Ater, Europhys. Lett. **69**, 798 (2005).
[25] J. Bechhoefer, New J. Phys. **17**, 075003 (2015).
[26] O. Zuk, I. Kanter, and E. Domany, J. Stat. Phys. **121**, 343 (2005).
[27] E. Lathouwers and J. Bechhoefer, Phys. Rev. E **95**, 062144 (2017).
[28] A. E. Allahverdyan and A. Galstyan, in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)* (AUAI Press, Arlington, VA, 2009).
[29] A. E. Allahverdyan and A. Galstyan, J. Stat. Phys. **161**, 452 (2015).
[30] T. Baum, Leonard E.; Petrie, Annals Math. Stat. **37**, 1554 (1966).
[31] V. Breuer and G. Radons, Phys. Rev. E **53**, 3982 (1996).
[32] A. J. Viterbi, IEEE Trans. Inform. Theory **13**, 260 (1967).
[33] J. Lücke and D. Forster, Patt. Recog. Lett. **125**, 349 (2019).
[34] J. Cardy, *Finite-size Scaling* (Elsevier, Amsterdam, 1988).
[35] W. Barthel and A. K. Hartmann, Phys. Rev. E **70**, 066120 (2004).
[36] A. Mann and A. K. Hartmann, Phys. Rev. E **82**, 056702 (2010).
[37] S. Cocco and R. Monasson, Phys. Rev. Lett. **86**, 1654 (2001).
[38] M. Mézard and A. Montanari, *Information, Physics and Computation* (Oxford University Press, Oxford, UK, 2009).