Trabajo de Fin de Grado de Física

Clasificación de transiciones de fase cúanticas mediante aprendizaje automático cuántico

Teresa Sancho Lorente

Director:

David Zueco Láinez

Departamento de Física de la Materia Condensada
Facultad de Ciencias, Universidad de Zaragoza

Junio de 2021

Undergraduate Dissertation in Physics

Classification of quantum phase transitions using quantum machine learning

Teresa Sancho Lorente

Director:

David Zueco Láinez

Universidad Zaragoza

Facultad de Ciencias
Universidad Zaragoza

Department of Condensed Matter Physics
Faculty of Science, University of Zaragoza

June 2021

I would like to express my deep gratitude to my family, from whom I have always received nothing but support and affection. To all my colleagues who have made this four-year journey an unbeatable experience.

I would also like to thank David and Juan, for devoting their time to teach me, not only physics, but also the importance of enjoying the learning process. They have been a true example of patience, humanity and hard work.

Contents

Introduction 1

Objectives and outline 2

1 Quantum Phase Transitions 3

1.1 Quantum Ising model 5

2 Study of phase transitions through fidelity 7

2.1 Finite size scaling 8

3 Support vector machine (SVM) 10

3.1 Kernel trick 13

3.2 Quantum Kernel(s) 14

4 Results 15

4.1 Fidelity based Kernel ($K^{(F)}$) 15

4.2 Fidelity-per-site based Kernel ($K^{(\lambda)}$) 18

4.3 Kernel Alignment 19

5 Conclusions 22

6 References 23

A Code 24

Introduction

Both machine learning and quantum computing are the cradle of a large number of research projects developed in the last decade. The reason is simple, the suggestive idea of finding new tools which allow the computational treatment of larger amounts of data is, not only a difficult task for experts in the field, but also the gateway to breaking the computational frontiers known so far. Undoubtedly, this is very attractive for the global technology industry. The achievement of this goal seems to be based on the constructive combination of two disciplines: quantum physics and artificial intelligence. Nevertheless, it is not clear to date whether running quantum machine learning (QML), *i.e.* ML algorithms in a quantum computer, will be practical.

The difficulty, as well as the possible advantage this computation presents, lies in the enormous dimensions of the Hilbert spaces with which these computers work. To help us understand this concept, consider that quantum computers with 50-100 qubits are able to perform tasks which surpass the capabilities of today's classical digital computers. These computers are already available and they are called Noise Intermediate Size Quantum (NISQ) [1] computers. Nevertheless, they are seriously limited by noise and decoherence.
In parallel to the evolution of this technology, the boom in artificial intelligence has led to the development of tools to implement algorithms into quantum computers [2] to carry out both quantum and classical tasks with the hope of dealing with intractable problems or improve the efficiency of those already solved. As a result: quantum optimization, quantum neural network models, quantum decision trees, finding ground and excited states of many body problems or quantum circuit optimization have already been reported. These algorithms are based on encoding the data into quantum states, *i.e.* $|\psi(\mathbf{x}_j)\rangle$, with $\mathbf{x}_j$ the input data. The encoding is followed by the processing of the state in a q-computer $U_\theta|\psi(\mathbf{x}_j)\rangle$. Similarly to classical machine learning , the solution of the problem lies in the optimization of a cost function $f = \langle M\rangle_{\theta,j}$ in order to find the optimal U_θ which transforms the initial state into the expected outcome.
Among the different classical algorithms that can be *quantized* in this way are Kernel methods. In them, the principal object is the Kernel: the inner product of the input data ($\mathbf{x}_j$). They are, for example, used in classification tasks. The quantum counterpart of Kernel methods uses the encoding $\mathbf{x}_j \to |\psi(\mathbf{x}_j)\rangle$, so $K = |\langle\psi(\mathbf{x}_j)|\psi(\mathbf{x}_{j'})\rangle|$, *i.e.* the Kernel is nothing but the Fidelity between two quantum states. We can particularly think of adapting the classical support vector machine (SVM) to classify datasets. In this framework, the quantum advantage (if any) occurs if quantum mapping is necessary to solve the problem or, at least, shows better performance than the classical techniques. So far, a quantum speed up has been rigorously shown for classifying data inspired in the discrete logarithmic problem [3].

In this Undergraduate Dissertation we want to explore this possible quantum advantage in quantum phase transitions classification using quantum kernels.

Objectives and outline

This dissertation has as its principal objective to answer one of the main questions currently being discussed about quantum computing: Are quantum computers capable of carrying out artificial intelligence (AI) tasks in a better or faster way than classical computers do? Or expressed differently: Can we find examples where quantum computing supremacy is plausible?

The pursuit of the answer to these questions lead us to the comprehension of the advantages and drawbacks of current quantum computers in order to develop intuition to look for a concrete problem which allows us to achieve our goal.

Inspired by the impossibility of classical computers to process data from the Hilbert space due to its large dimensions, we chose a quantum task and study whether quantum machine learning could solve it or not. Particularly, we discuss the performance of quantum Kernels in the detection of quantum phase transitions (QPT) through the study of the one dimensional quantum Ising model. As a direct implication, our work begins by understanding support vector machines (the algorithm with which we will develop the idea) and quantum phase transitions. As benchmark, we reproduce a study of these transitions through the fidelity of the ground states of the system. Since we do not have reliable quantum computers yet, we use the Jordan Wigner transformation to solve the quantum Ising model and construct the quantum Kernel. The data that we handle is quantum by nature. Thus, by showing that this quantum machine learning algorithm is capable of finding the transition boundary accurately, supremacy is confirmed and the method can be extrapolated to solve non-exactly-solvable models.

Accordingly, this Undergraduate Dissertation is structured as follows: in Section 1 we briefly explain the main traits of quantum phase transitions and particularize them introducing, in Section 1.1, the quantum Ising model. To continue, in Section 2, we reproduce a study of the transition in our model using the fidelity between quantum states. We show how the critical parameter scales with the number of spins in the considered chain and use these results to compare with the ones predicted by our algorithm. We also present Support Vector Machines (SVM) and their operation as the algorithm used to carry out our analysis of the transitions, in Section 3. Particularly, in sub-Section 3.1 we show the Kernel Trick and the quantum kernels we will analyze. Finally, the results are presented and discussed in Section 4. We finish with some conclusions and include an Appendix with a link to a public repository where all the code, originally developed for this project, can be found.

1 Quantum Phase Transitions

Their importance in nature makes them one of the topics which have attracted most the attention from physicists in the last decades. Phase transitions appeared to be everywhere, from the boiling of water to the complex phenomenon of superconductivity, and their applications jump from physics to mathematics, information science, biology and many other scientific domains. In order to take full advantage of these processes and use them in different applications it is of the utmost importance to characterize them. This can be translated into answering two simple questions: when does the phase transition occur, and what does it signify?

When speaking of classical phase transitions, the combination of thermodynamics, statistics and condensed matter physics with experiments in different systems, have allowed us to understand their operating principles. In these kind of transitions, two states of matter are clearly separated by the abrupt change of some properties which help us to determine whether the system is in one phase or another. To fix ideas, let us begin with some important basic concepts on classical phase transitions.

In particular, we might think of the classical Ising model, which is perfect to exemplify these concepts. Here, thermal fluctuations compete, to drive the system to disorder, with the spin-spin interaction, which tends to correlate the spins and thus, order the system. This competition yields to a critical temperature T_c, such that for $T > T_c$ the spins fluctuate without any correlation. For $T < T_c$ the spins in the system are ordered, since the interactions between the spins are larger than the thermal fluctuations. The phase transition, in this case, is characterized by looking at the magnetization. In the disordered phase the spins are randomly oriented, so the magnetization is zero. In the ordered phase, a macroscopic magnetization emerges. This observable that changes from zero to different from zero at the transition is the order parameter, that can be used to fully characterize the critical properties of the system under study.

We summarise two main important concepts discussed with the example above. Firstly, we introduced the driver of transitions in classical statistical mechanics: thermal fluctuations. Secondly, we have defined the concept of order parameter that witnesses the transition: magnetization in the above example. To end with our description of classical transitions, we will distinguish two types of phase transitions. First order transitions, when the first derivative of the order parameter is discontinuous, and second order transitions, also called continuous, when the discontinuity occurs in the second derivative of the order parameter.

A universal feature of second order phase transitions is the divergence of the correlation length (ξ)[4] at the critical point [1] In the Ising model, this magnitude is defined as the typical size of clusters of aligned spins and diverges in the critical point as:

$$\xi \, \alpha \left| \frac{T - T_c}{Tc} \right|^{-\nu} \tag{1}$$

[1] First order transitions do not present this divergence of the correlation length in general.

where ν is the thermal critical exponent, dependent on the dimensions and symmetries of the system and its Hamiltonian. This characteristic is specially interesting since it reflects a universal behaviour in continuous phase transitions, independent of microscopic traits of the systems [4]. Since the focus of this Thesis are quantum phase transitions, let us jump into this phenomena. When we deal with microscopic systems at sufficiently low temperatures, quantum fluctuations become important and, in some cases, they can even drive a transition. When these transitions occur at zero temperature, in the absence of thermal fluctuations, they are called *quantum phase transitions (QPT)*.

Quantum phase transitions are defined as non-analyticity points in the ground state energy of the system [5]. Since the temperature is fixed and null, there must be a parameter g in the Hamiltonian which controls the transition, such that for $g = g_c$ this non-analyticity occurs. Without loss of generality, let us write the Hamiltonian as the sum of two terms and a g-parameter:

$$H(g) = H_0 + gH_1 \,. \tag{2}$$

If both terms in the Hamiltonian commute, $[H_0, H_1] = 0$, they present common eigenvalues and a level crossing may occur, as it is shown in Fig. 1(a). At this level crossing, the ground state energy will be non-analytical. This is a first order quantum phase transition. Notice that this can happen in finite systems (no thermodynamic limit here). In what follows we will focus on another type of QPT, second order or continuous QPTs. When $[H_0, H_1] \neq 0$, an anti crossing emerges leading to a QPT at the thermodynamic limit, where the gap closes (see Fig.1b)). For any finite-size system a continuous transition will be described as an avoided level crossing in the ground state.

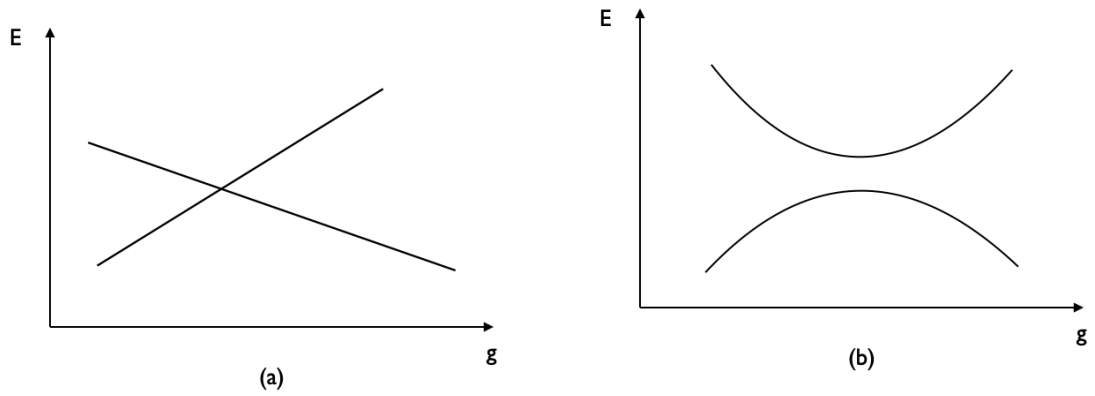

Fig. 1: (a) Example of a crossing (First order QPT). (b) Example of an anti crossing, the seed for a second order QPT

Pretty much like in the classical case, in second order phase transitions the correlation length also diverges, in this case as a function of the g-parameter [6].

$$\xi \; \alpha \; |g - g_c|^{-\nu} \tag{3}$$

where ν is the critical exponent, Cf. Eq. (1).

Historically, quantum phase transitions have been studied following the patterns used in the classical approach. Again, we look for an driving parameter whose non-null value characterizes,

in an infinite system, a break on its symmetry and thus, a phase transition. Nevertheless, the difference between two ground states, $|\phi\rangle$ and $|\psi\rangle$, in terms of this driving parameter is based in the quantum measurement of the observable associated to this parameter. This distinguishability of states is, in some way, equivalent to the condition of orthogonality between states [7]. We can say that, if two states $|\phi\rangle$ and $|\psi\rangle$ are orthogonal, then there must necessarily exist an observable, i.e. a parameter, capable of distinguishing them. In the limit case, this observable will be the projection of both states, $\langle\psi|\phi\rangle$.
Knowing that orthogonal states can be reliably distinguished, one can think of using this equivalence shown above to characterize phase transitions without resorting to an order parameter. This modern point of view inherited from quantum information theory focuses its attention on the ground state wave function, instead of looking at the Hamiltonian [8]. It is based in the fidelity measure between quantum states.

$$F(\rho, \sigma) = \sqrt{\rho\sigma\rho} \tag{4}$$

where ρ and σ are the density matrix associated to any two states of the system in study. If we work with pure states, this expression can be rewritten as:

$$F(\psi, \phi) = |\langle\psi|\phi\rangle| \tag{5}$$

This fidelity is a natural measure of closeness between quantum states. Assuming all the wave functions are normalized:

$$F(\psi, \phi) \in [0, 1] \tag{6}$$

such that if $\phi = \psi \rightarrow F = 1$, while orthogonal states give $F = 0$.

Since we are considering two different phases, using the argument stated above, we expect ground states at both sides of the transition to be orthogonal. In Section 2, we study the characterization of phase transitions using this fidelity measure which is intrinsic to the concept of quantum states, instead of searching for the adequate order parameter for the system in study. This different approach to the problem enables us to identify phase transitions without information of the Hamiltonian used to describe our system, only using the ground states for different values of the coupling parameter g. To carry out with our purpose, we present the results through a particular example, chosen for its importance in many scientific fields; the quantum Ising model.

1.1 Quantum Ising model

As described above, the one dimensional Ising model explains the behaviour of the spins in a chain. Its quantum version considers the exchange interactions between neighbour spins and the force created applying a magnetic field. These two interactions compete to align the spins in two transverse directions as it can be seen in the Hamiltonian:

$$H_I = -J \sum_{i=1}^{N} \hat{\sigma}_i^x \hat{\sigma}_{i+1}^x + \sum_{i=1}^{N} \hat{\sigma}_i^z \tag{7}$$

Note that, throughout this work, $\hbar = 1$. Here, σ_α, $\alpha = x, y, z$ are Pauli matrices representing spin-1/2. The parameter J is called the coupling constant. The first term in the equation

represents the exchange interactions between spins that occur in form of quantum fluctuations and the second term corresponds to the interaction of the spins with the transverse magnetic field applied. As we have explained above, both terms compete to align the spins in two transverse directions.

If we consider a N-site 1D lattice, where each spin has 2 dimensions, the total Hilbert space has dimension 2^N. Each function in the basis of this space will be constructed as the tensor product extended to all the spins in the chain. The elements in the basis have the form:

$$|\phi\rangle = |\uparrow\rangle_1 \otimes ... |\downarrow\rangle_j \otimes ... \otimes |\uparrow\rangle_N \tag{8}$$

This model has been historically important for its many applications in physics and it is always used as a good framework for all those interested in the study of phase transitions since we can easily intuit what will happen in both phases as a competition between the exchange (J) and the magnetic interaction.

With the purpose of developing this intuition, we present a more visual description of both paramagnetic and ferromagnetic states. The paramagnetic phase presents all the spins aligned with the z-axis. This phase appears when the parameter J present values below the critical point ($J < J_C$).

$$|\uparrow\uparrow\uparrow\uparrow ... \uparrow\uparrow\rangle$$

The ferromagnetic phase [9] appears when $J > J_C$ and corresponds to situations where the exchange interaction between spins dominates and the spins are aligned with the x-axis. Note that there are two degenerate ferromagnetic ground states, parallel and anti-parallel to the x-axis. In fact, this is a consequence of a discrete $\mathbb{Z}_2$ symmetry. At the phase transition, the symmetry is broken and one of the possible states emerges, thus the x-magnetization (order parameter) becomes different from zero. For example, the g.s. is

$$|\rightarrow\rightarrow\rightarrow\rightarrow ... \rightarrow\rightarrow\rangle$$

In this work we set $J > 0$, *i.e.* the system is ferromagnetic.

A final and important advantage of using the Ising model lies in the fact that it is analytically solvable using the Jordan-Wigner transformation [10, 11] which maps 1/2-spins into spinless fermions enabling us to reduce the computational cost of calculating the ground state functions and, thus, to simulate with larger chains. Due to its classical importance, this model has already been resolved and it is known that the transition between both ferromagnetic and paramagnetic states occurs at $J_c = 1$. Also, the universal critical exponent describing the divergence of the correlation length (see Eq.(3)) is $\nu = 1$. All this previous knowledge makes it possible to evaluate the quality of the results obtained for the methods we present to solve this transition task by comparing them with these previously known results.

After presenting the model, we will characterize the phase transition using the fidelity. This concept, already mentioned above, is based on the orthogonality of the ground states from both phases and has been extensively developed by other scientists. [11, 12]. We reproduce below the results of these studies.

2 Study of phase transitions through fidelity

In this section we pursue the comprehension of the fidelity and its utility to characterise quantum phase transitions such as the one studied in this research project. For the Ising model the fidelity can be expressed [11, 10]:

$$F(J, J') = \langle \psi_0(J) \mid \psi_0(J') \rangle = \prod_k \cos\Big(\theta_k(J) - \theta_k(J\prime)\Big) \,, \tag{9}$$

here, $|\psi_0(J)\rangle$ is the ground state of (7) for a given coupling strength J. And we define:

$$\cos 2\theta_k(J) = \frac{1 + 2J\cos k}{\sqrt{1 + 4J\cos k + 4J^2}} \,. \tag{10}$$

To define the values of k we impose periodic boundary conditions and obtain, considering N-even chains:

$$k = \frac{(2n-1)\pi}{N} \tag{11}$$

with $n = 1, ..., N/2$.

Following [11] we compute the fidelity between two close states $F(J, J-\delta)$ to detect the phase transition. It is expected that if J and $J-\delta$, with δ small enough, are in the same phase then $F(J, J-\delta) \cong 1$. Only when the phase boundary is crossed, the fidelity drops. To compute the plots, we work with:

$$J \in \ [0.25 : 1.75]$$

and

$$\delta_J = 0.0002 \,.$$

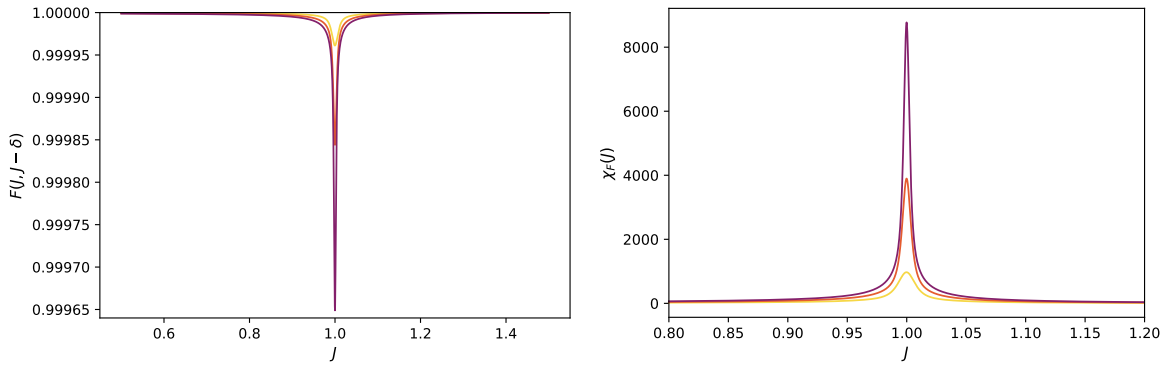

Fig. 2: (a) Fidelity between adjacent states for N=250, 500, 750. (b) Susceptibility for N=250, 500, 750. We recall that $\delta = 0.0002$

In Fig. 2a) we see, as expected, that far from the transition the fidelity between each state and its adjacent is almost one. Since they are subject to very similar coupling parameters and we are far from the thermodynamic limit, they seem to be quasi-identical states. In the phase boundary, no matter how close two adjacent g.s. are in the state space, they belong to different phases and thus they are different. As a conclusion, the fidelity presents a minimum in the transition. Another interesting quantity to compute is the susceptibility, defined after the expansion,

$$F(J, J-\delta) \cong 1 - \chi\delta^2 \,.$$

Thus:

$$\chi_F(J) = \frac{1 - F(J, J-\delta)}{\delta^2} \,. \tag{12}$$

As we can see in Fig. 2, both fidelity and susceptibility approach the divergence when increasing the number of spins in the chain (N). One advantage of the latter is that it is independent of the discretization (δ) used. We can assure that the maximum of this function will be equivalent to the minimal value of fidelity, with the advantage that now we are working with an analytical function.

2.1 Finite size scaling

We recall that second order quantum phase transitions occur in the thermodynamic limit, $N \to \infty$, where the system presents no gap between the g.s. and the first excited state (see Fig. **??**). In order to understand how our finite system approaches this limit, we performed simulations for different Ns using the fidelity to characterize the transition of the system.

We first notice, as we have seen in Fig. 2, that the fidelity susceptibility, defined in (12), scales with the size of the system. Thus, the maximum of the function approaches the divergence as a function of N, until it reaches the asymptotic value in the thermodynamic limit. This finite-size scaling has been studied before [11, 12] and can be justified knowing that χ_F is directly related to the energy spectrum and, thus, to the system fluctuations using perturbation theory. If the system J- dependence is written explicitly as :

$$H = H_0 + JH_I \,.$$

It can be found using perturbation theory that:

$$\chi_F = \sum_{n \neq 0} \frac{|\langle \psi_n(J)|H_i|\psi_0(J)\rangle|^2}{(E_n(J) - E_0(J))^2} \,. \tag{13}$$

Here, $|\psi_n(J)\rangle$ and $E_n(J)$ are the spectrum, $|\psi_0(J)\rangle$ and $E_0(J)$ the ground state and its energy. Standard argumentation yields:

$$\chi_F \sim N^{\gamma/\nu} \,,$$

with γ and ν universal critical exponents, whose theoretical value is already known. This expression suggest that we can perform a scaling analysis in order to verify this expressions. We fit the scaling formula:

$$|J - J_C(N)| \sim N^{-1/\nu} \,, \tag{14}$$

where $J_C(N)$ is the critical value of the driving parameter for a given length N and $\nu = 1$ is known to be a universal exponent.

However, we found that using the fidelity (or its susceptibility) for extracting both J_c and ν is not so easy for two main reasons. The first one is that, given the shape of the divergent functions presented in Fig. 2, it is not easy to fit them. The second one is related to the orthogonality catastrophe. For sufficiently large N the Hilbert space is so big that any two states are (almost) orthogonal. Therefore, δ needs to be reduced as N increases in order to preserve the similarity of adjacent states.

To solve these problems, Zhou and coworkers [7, 12, 13] computed the scaling analysis with the fidelity per site $\lambda(J, J')$ instead, defined in the thermodynamic limit as:

$$\ln \lambda(J, J') = \lim_{N\to\infty} \frac{1}{N} \ln F(J, J') \; . \tag{15}$$

They use the logarithm in order to smooth the divergence of this function. Although it is defined in the thermodynamic limit, it can be extended to the finite case as:

$$\ln \lambda(J, J') = \frac{1}{N} \ln F(J, J') \; . \tag{16}$$

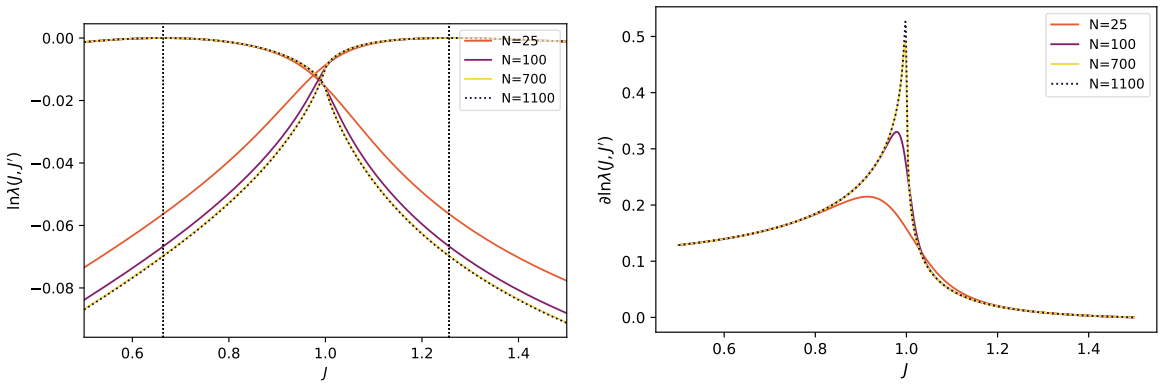

Fig. 3: (a) We compute the fidelity between every J and J'= 1.256. We repeat the curves for J'=0.664. Both J' are marked with a dotted vertical line. (b) The derivative of the $ln\lambda(J, J')$ for N= 25, 100, 700, 1100 and $J' = 1.4998$.

We plot the fidelity per site. In practical terms, to find the critical point of this function we also plot its derivative for different N (Fig. 3). The maximum of this derivative function (see Fig. 3b)) is identified with the intersection of lines in Fig. 3a). It is a pinch point of the λ function [7, 12]. This singular behaviour can only be associated to a transition of the system, i.e. it is the critical point we are looking for. Also, we see how the derivative approaches the divergence for large N. The critical value $J_C(N)$ and the scaling obtained by measuring the maximum of these derivative curves for different N is shown in Fig. 4a) and b) respectively.

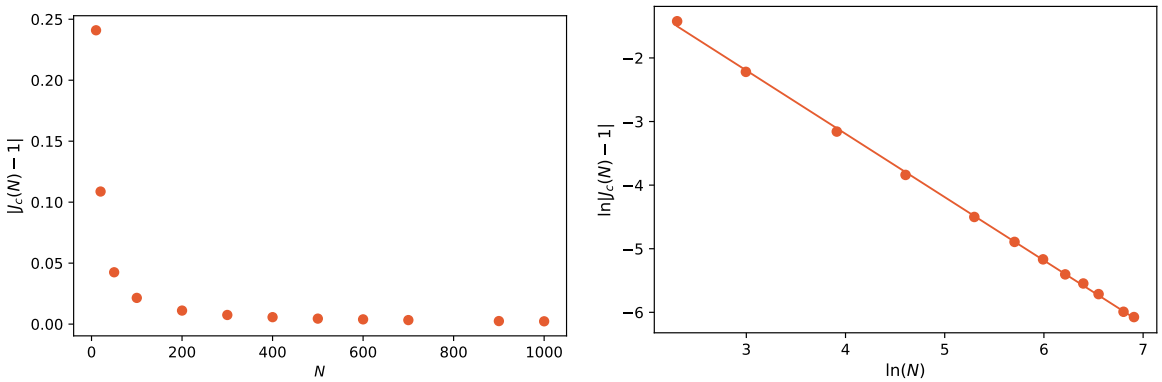

Fig. 4: (a) $J_C(N)$ predicted using the fidelity per site for different N. (b) Linear fit of the scaling of J_C with the length of the spin chain.

We find a critical exponent $\nu = 1.004$ which is remarkably close to the theoretical expected value $\nu = 1$. In conclusion, we have verified the scaling law our model presented, Eq. (14).

3 Support vector machine (SVM)

A support vector machine is a supervised learning algorithm, created to classify data sets in different categories. This type of algorithms learn to predict outcomes from "well" labeled data, i.e. they need prior training. We now present the mathematical foundations on which the SVMs are built, in order to fully understand its functioning. We start presenting the easiest case, linearly separable points i.e. classes we can separate through a straight line.

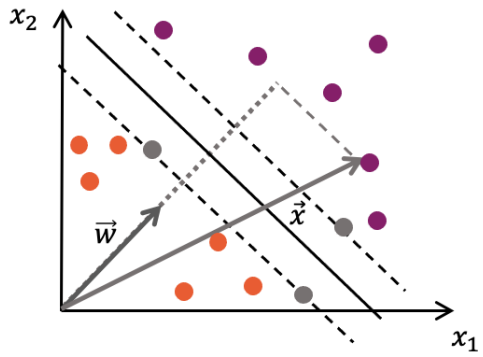

Fig. 5: Functioning of a support vector machine. The hyperplane (solid line), and support vectors (gray points) are shown.

The operation of the algorithm is based on the formulation of the optimal hyperplane that separates the set of points in classes, Cf. Fig. 5.

The construction of the hyperplane is based on the characterization of two parameters, the norm vector to the plane ($\vec{w}$), called weight vector, and the intercept term b, which defines the optimal hyperplane from all the planes perpendicular to $\vec{w}$.

With these two parameters, the algorithm is capable of classifying the points. To do so, it defines a decision function whose value is different for points in different classes. This decision function is defined as:

$$y_i = f(\vec{x_i}) = sign(\vec{\omega} \cdot \vec{x_i} + b)\,. \tag{17}$$

It is equal to one when the data point $\vec{x}$ lies on the positive side of the hyperplane and, when the point lies on the negative side of the optimal hyperplane, it is equal to -1. In fact the algorithm finds both $\vec{\omega}$ and b, such that, given a vector $\vec{x_i}$ depending on the sign of $f(\vec{x_i})$ it is assigned to a class or another.

In order to find these parameters, the algorithm needs a training set of already-classified points, so that it will use them to transform the issue into a minimization problem as we will see now. Our purpose is to understand the functioning of the algorithm. To do so, we reproduce here the steps followed by the SVM to separate the data set. Let´s simulate the simplest scenario. Suppose we have a training set with data we want to classify in two linearly separable groups. Firstly, we introduce a crucial concept in the definition of SVMs, the support vectors. Support vectors are defined as the only points in the training set that contribute to delimit the hyperplane. They define the geometric margins (dotted lines in Fig.5 at both sides of the plane). They are found by imposing:

$$\vec{\omega} \cdot \vec{x}_- + b = -1\,, \tag{18}$$

$$\vec{\omega} \cdot \vec{x}_+ + b = 1\,, \tag{19}$$

here, $\vec{x}_-$ and $\vec{x}_+$ are support vectors corresponding to left and right margins respectively. These constraints enable us to fix the width of the space delimited by the geometric margins. This width (d) can be computed projecting the distance between the support vectors $\vec{x_+} - \vec{x_-}$ onto a unitary vector normal to the hyperplane. This calculation is embodied in Fig.6.

$$d = (\vec{x_+} - \vec{x_-}) \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|} \,. \tag{20}$$

Manipulating the definitions given for support vectors in Eq.(18) and Eq.(19), multiplying at both sides for $\vec{\omega}$ we find:

$$\vec{x}_- = -(1+b)\frac{\vec{\omega}}{\|\omega\|^2} \tag{21}$$

$$\vec{x}_+ = (1-b)\frac{\vec{\omega}}{\|\omega\|^2} \,. \tag{22}$$

Finally, we subtract both vectors to calculate the distance between them:

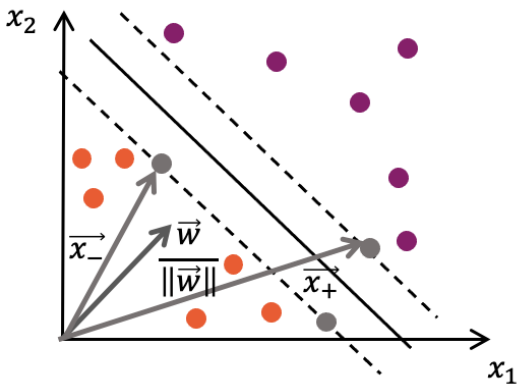

Fig. 6: Characterizing the optimal plane.

$$\vec{x_+} - \vec{x_-} = 2\,\frac{\vec{\omega}}{\|\omega\|^2} \tag{23}$$

and so

$$d = \frac{2}{\|\vec{\omega}\|} \,. \tag{24}$$

As we can see, the distance depends only on the module of the vector that defines the hyperplane.

Since we are looking for the optimal separation, we impose this distance between both margins to be maximum (see Fig. 6). From Eq.(24) is deduced that the maximization of the width is equivalent to the minimization of the normal vector ($\vec{\omega}$). Thus, we have transformed the initial problem into an equivalent minimization question. The minimization we want to perform presents a main restriction, the classification of points in the data set made by the decision function must match the classification given for these points. We must recall that the set used to solve the problem is already labelled (training data set). Before searching for the solution to the problem, we rewrite, for convenience, the decision function as:

$$y_i(\vec{\omega} \cdot \vec{x_i} + b) \geq 1 \,, \tag{25}$$

where y_i is the classification of each point, so $y_i = -1$ for points in the left side of the plane and $y_i = 1$ for points in the right side. It is easy to prove the equivalence between both definitions ((17)) and ((25)). Notice that this function cancels out for points in the separating hyperplane and equal to one for all the support vectors. Looking at the definition of the function we want to minimize and its restriction we conclude that this optimization problem admits a solution which

can be achieved using Lagrange multipliers [14]. We define the multipliers (α_i) and construct the Lagrangian:

$$\mathcal{L} = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_{i=1} \alpha_i[y_i(\vec{\omega}\cdot\vec{x_i}+b)-1]\,, \tag{26}$$

where we sum uniquely over the support vectors, or equivalently, we impose $\alpha_i \neq 0$ if and only if x_i is a support vector. The method consists on applying:

$$\nabla\mathcal{L} = 0 \tag{27}$$

and solving the system of equations obtained. For our particular problem, we obtain the solution in the form of two equations which fully characterize the separating hyperplane:

$$\vec{w} = \sum_{i=1} \alpha_i y_i x_i \tag{28}$$

$$b = y_i - \vec{w}^T\vec{x_i} \;\; for\; all\; \vec{x_i}\; with\; \alpha_i \neq 0 \tag{29}$$

It is interesting to redefine the decision function in order to find some of its properties. Using (28) and (29) we can express, for a given point $\vec{x}$ with label y:

$$f(\vec{x}) = y(\sum_{i=1} \alpha_i y_i \vec{x_i}\vec{x} + b) \geq 1\,. \tag{30}$$

Consequently, our algorithm is going to classify any given point as a function of the distance to the support vectors, i.e. the distance to the margins. Following this procedure, we can guess that the training set is going to be a crucial element in order to achieve an accurate classification of the set, since it is the set from where our algorithm learns.

So far, we have discussed linear separable data. Let's raise the difficulty a bit and suppose now that the data set is mainly separable but presents some outliers that break separability. In this case, we use the so-called soft margin technique [15] and drop the constrain that each point must lie on the correct side of the margin (25). Instead, we penalize points incorrectly classified using the hinge loss function:

$$max(0, 1 - y_i(\vec{\omega}\cdot\vec{x_i}+b)). \tag{31}$$

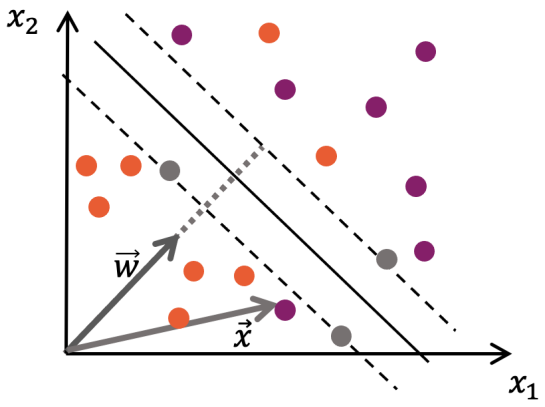

Fig. 7: Example of set solved with soft margin techniques.

It is zero if the constraint (25) is satisfied and otherwise proportional to the distance from the margin. The new decision function is:

$$y_i(\vec{\omega}\cdot\vec{x_i}+b) \geq 1 - \zeta_i\,. \tag{32}$$

Here, ζ_i allows x_i to lie outside the established margins. The optimization problem is now trading off how wide is the margin versus how many points are misclassified with such margin. It can be written as:

$$\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(\vec{\omega}\cdot\vec{x_i}+b)) + \lambda||\vec{\omega}||^2\,, \tag{33}$$

where λ represents the trade-off mentioned above. We can convert again this problem into a maximization problem, solvable using Lagrange multipliers, a bit more complicated this time.

$$\mathcal{L} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) \,. \tag{34}$$

Subjected to the constraints:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{35}$$

and

$$0 \leq c_i \leq \frac{1}{2n\lambda} \,, \tag{36}$$

where i sums to all the support vectors. Note that the dual problem, as this maximization is called, does not depend on parameters ζ_i . The constant $\frac{1}{2n\lambda}$ it the maximum possible size of the Lagrange multipliers for the support vectors. The solution to this problem is of the form:

$$\vec{\omega} = \sum \alpha_i y_i \vec{x}_i \tag{37}$$

and

$$b = y_k - \vec{\omega} \cdot \vec{x}_k \,, \tag{38}$$

where $\vec{x}_k$ is located on the margin´s boundary.
Soft margins are commonly used since few problems are completely linearly separable.

3.1 Kernel trick

Now, we can go a bit further and ask ourselves what would happen if we give our algorithm a non-linearly separable data set to work with, as in Fig 8. In these cases, we cannot compute a separating hyperplane to delimit different groups of data and thus, we use the *kernel trick*.

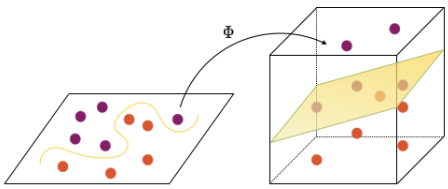

Fig. 8: Kernel Trick.

The Kernel trick is nothing but noticing that both the Lagrangian (26) and (30) depend only on the inner product:

$$K_{ij} \equiv \vec{x}_i \cdot \vec{x}_j \,. \tag{39}$$

. Here, we have already introduce the notation K an call it Kernel. This Kernel is nothing but a function that corresponds to a dot product in some expanded feature space.

Thus, the Kernel trick consists on mapping the points into a higher dimensional space, called feature space, where they may be linearly separable, as in Fig. 8. Being concrete, we define the map as:

$$\vec{x_i} \longrightarrow \Phi(\vec{x_i}) \,, \tag{40}$$

such that the Kernel (39) consists of the inner product in the feature space:

$$K(\vec{x_i}\vec{x_j}) = \Phi(\vec{x_i})\Phi(\vec{x_j}) \,. \tag{41}$$

Notice that to calculate the decision function between two points in the set, we only need to know how to compute the dot product in the feature space. As a consequence, it is not necessary to specify the mapping used. Finally, the decision function is also written in terms of the Kernel:

$$y\ (\ \sum_{i=1} \alpha_i y_i K(\vec{x_i}, \vec{x}) + b) \geq 0\,. \tag{42}$$

The crux of the matter is that classification is now given by the Kernel matrix. It is nothing but a function that takes two input data, and returns a real number characterizing their similarity. Obviously, in general, the map that measures this similarity is not known. Typically, what is done is mapping to a very high dimension space to ease the separation. However, if we know about the data we may find a mapping capable of measuring that similarity. This is what we study in the next section

3.2 Quantum Kernel(s)

In our case, we want to classify quantum states. We have already presented a quantity that measures the 'distance' or similarity between quantum states, which is the fidelity. Besides, we have argued before it is suitable for characterizing phase transitions. Therefore, it seems natural to choose:

$$K^{(F)}(\psi_0(J), \psi_0(J')) = |\left\langle \psi_0(J) \middle| \psi_0(J') \right\rangle| \tag{43}$$

as our quantum Kernel.
Also, from what we learnt in previous sections, we will use a second Kernel in terms of the fidelity per site. This one is a measure of similarity but with the advantages of avoiding the orthogonality catastrophe. We construct this second Kernel as:

$$K^{(\lambda)}(\psi_0(J), \psi_0(J')) = |\left\langle \psi_0(J) \middle| \psi_0(J') \right\rangle|^{1/N}\,. \tag{44}$$

In what follows, we discuss the performance of this two different kernels in the task of locating the phase transition in the quantum Ising model.

4 Results

4.1 Fidelity based Kernel ($K^{(F)}$)

It is time to present the results of comparing the prediction of the critical value made by the SVM with the results obtained with the fidelity analysis (Fig.4). In Fig. 9 we show this comparison with results obtained from the SVM trained in two intervals of J, one for each phase, far from the transition. The intervals are non-symmetric in order to challenge our algorithm abilities in the classifying task.

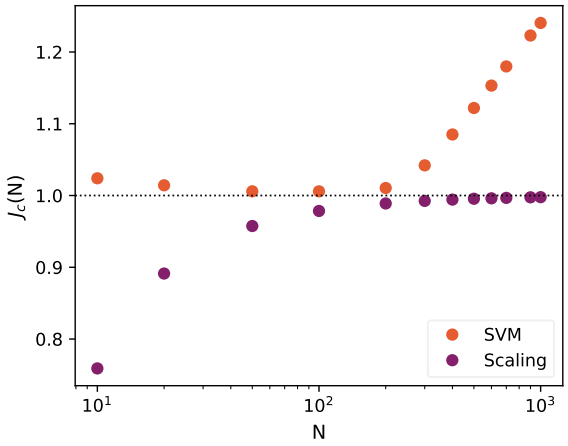

Fig. 9: Comparison of estimated J_c for both fidelity analysis and SVM results. The SVM has been trained in the intervals [0.6, 0.7] U [1.6, 1.7].

Looking at the results, it is obvious that SVM predictions do not scale with the number of spins in the chain (N). It seems advantageous for small chains, where the predictions approach better the theoretical phase boundary $J_C = 1$ than the benchmark results obtained with the finite size scaling. Nevertheless, for large spin chains, the SVM predictions seem to diverge from the expected J_C and so our system loses all its validity. In order to go deep into this phenomena we present three different results from the SVM trained with three different sets of Js. The first is made up of random Js all along the J-interval. The second one is the interval set used above. The third one is again an interval set, but this time made up with Js closer to the transition $J \in [0.7, 0.8]\ U\ [1.2, 1.3]$. As in Section 2, we work with:

$$J \in\ [0.25 : 1.75]\,,$$

but a larger separation between adjacent Js, since that high precision is not required anymore.

$$\delta_J = 0.0015\ .$$

All three training sets contain approximately a 10% of the Js in the J-interval.

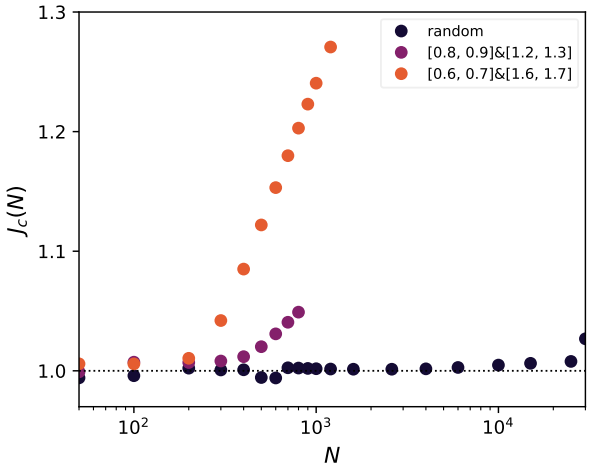

Fig. 10: Divergence of the predicted J_C with three different trainings, done with random Js, $J \in [0.7 : 0.8]\ U\ [1.2 : 1.3]$ and $J \in [0.6 : 0.7]\ U\ [1.6 : 1.7]$.

Looking at Fig. 10, both three trainings conclude in a divergence of the predicted transition boundary when they work with long spin chains. Nevertheless, we must highlight that the functioning of the algorithm is much better when it is trained with scattered values of J. In this case, the appearance of the divergence is delayed from $N \backsim 300$ to $N \backsim 30000$. The explanation of this phenomena is simple, the algorithm works much better when it has knowledge of all the space of classification. This way, it is capable of choosing better support vectors and constructing a more accurate hyperplane to distinguish the phases of the system studied. To fully analyze the functioning of the SVM and prove these arguments, we plot the distance of the ground states to the hyperplane:

$$d(\vec{x}) = \sum_{i=1} \alpha_i y_i \vec{x_i} \vec{x} + b\,, \tag{45}$$

as shown in Section 3 based on (17). This plot is made for interval and random training, and for different sizes of the chain (see Fig. 11 and 14). It characterizes the transition boundary.

In figure 11, in the interval training, it is marked in gray the intervals used for training the SVM. As we can see, for small N the distance function presents a coherent shape. It smoothly approaches null distance while J tends to J_c. It seems that the algorithm does not need a lot of support vectors to separate both phases. It is easy for the SVM to extrapolate the information in the training intervals to the rest of the set. These points contain enough information to enable an accurate classifying process. Meanwhile, for high N, the interval does not contain enough information to draw the boundary correctly. As a conclusion, we see a loss of accuracy reflected in the appearance of a flat zone at distance zero. The algorithm is not capable of characterizing the transition any more.

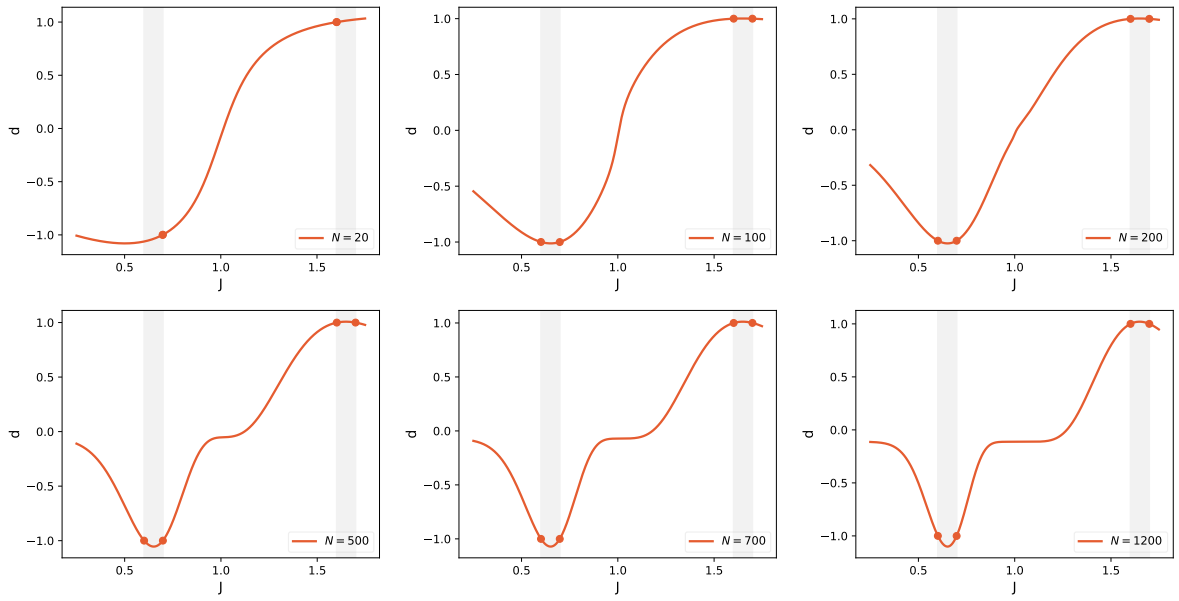

Fig. 11: Function distance to the hyperplane for N=20, 100, 200, 500, 700, 1200. The dots represent the support vectors chosen by the algorithm to decide the boundary. The training has been made in $J \in [0.6 : 0.7]\ U\ [1.6 : 1.7]$.

If we look now at the same figures done with a random training, the situation is a bit different. Since the algorithm now visits the whole J set, there are support vectors closer to the transition. Nevertheless, it has something in common with Fig.11; when N increases enough, the SVM fails. It takes all the training points as support vectors, and not only the nearest ones, loosing its ability to distinguish the better ones, and presents fluctuations in the distance function which prevent it from giving an accurate prediction.

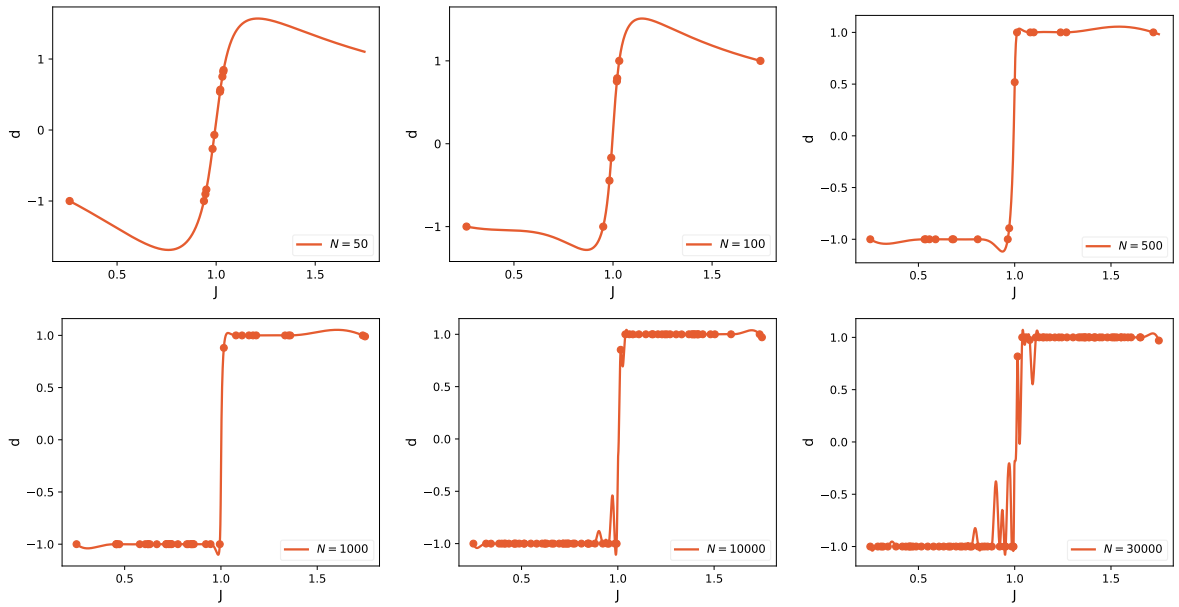

Fig. 12: Function distance to the hyperplane for N=50, 100, 500, 1000, 10000, 30000. Again the dots represent the support vectors chosen by the algorithm in each case.

In order to understand this rupture we recall Section 2.1 where we explain how fidelity is automatically linked to the appearance of the orthogonality catastrophe at a sufficient large Hilbert space. We recall that we work with states of 2^N dimensions and therefore, the similarity between adjacent states disappears fast with the increase of N. We propose a solution to this problem based on Zhou´s work [7, 12], as we have also explained in section 2.1, implementing a new quantum kernel based on the fidelity per site, see Sect. 3.2.

4.2 Fidelity-per-site based Kernel ($K^{(\lambda)}$)

We now present two different trainings, respectively made with intervals and random Js, and compare its predictions, observing their behaviour for large N.

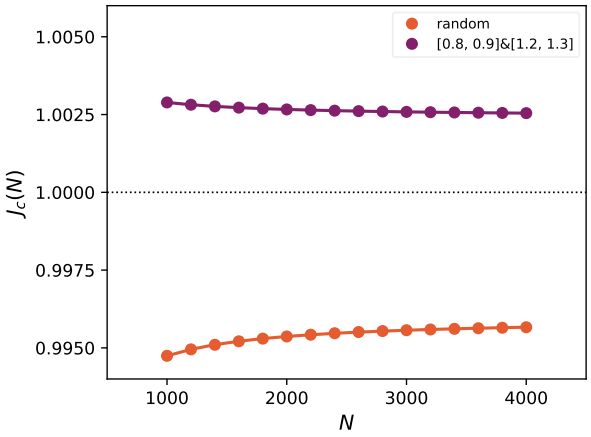

Fig. 13: SVM´s predicted J_C with the kernel based on the fidelity per site ($K^{(\lambda)}$) proposed and two different trainings, random Js and $J \in [0.8 : 0.9]\ U\ [1.2 : 1.3]$.

As expected, new predictions do not suffer the rupture seen before. Following what we have demonstrated in Section 2.1, the fidelity per site avoids the orthogonality catastrophe and so, presents this huge advantage when operating with large spin chains. Also, it can be seen that the interval training works as good as the random training. Consequently, our SVM estimates the transition boundary being trained with states located far from the transition. Finally, notice that critical values scale with N in same way that the λ function did, following the scaling law 14. For both trainings we find as critical exponents:

$$\nu_{rand} = 0.94 \tag{46}$$

and

$$\nu_{int} = 1.08 \tag{47}$$

We present again the distance function for different Ns in order to demonstrate the avoiding of rupture seen in Fig. 13

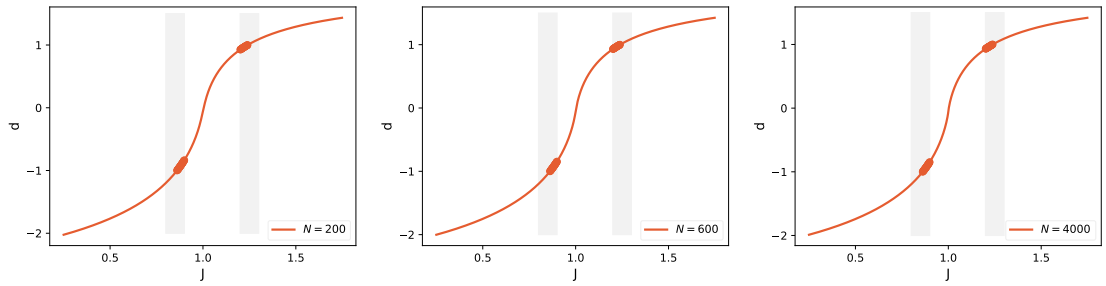

Fig. 14: Function distance to the hyperplane for N=200, 600 and 4000 respectively. SVM trained in the interval $J \in [0.8 : 0.9]\ U\ [1.2 : 1.3]$.

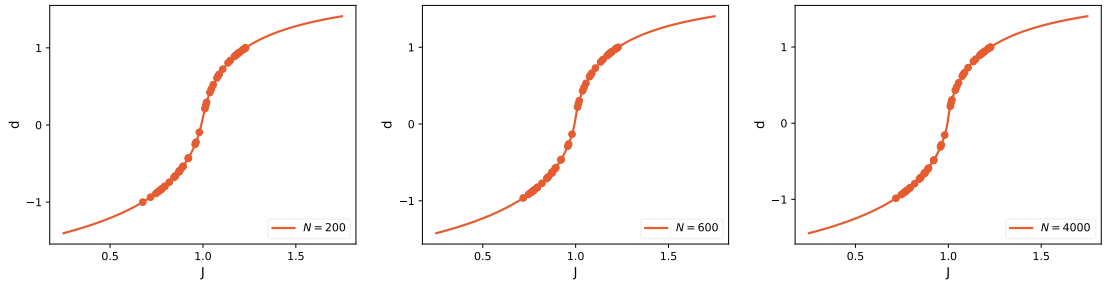

Fig. 15: Distance to the hyperplane for N= 200, 600 and 4000. SVM trained with a random set of Js.

The situation is quite different now. The distance functions are smooth and approach zero when J is J_c. We can also see that the function maintains its shape when we increase the length of the spin chain. We also see that it adapts its shape for different trainings, to assure that the support vectors are at distance ± 1 or less to the hyperplane. This is a direct consequence of its definition and the appearance of soft margins, seen in Section 3.

We have found the right kernel to deal with our classification problem. Nevertheless, apart from the quantum argument based on avoiding the orthogonality catastrophe mentioned above, we can also give a machine learning measure of whether a kernel is good.

4.3 Kernel Alignment

Based on [16], the kernel alignment is defined in terms of the Frobenius distance between matrices:

$$A := \frac{\mathrm{Tr}[k_1\ k_2]}{\sqrt{\mathrm{Tr}[k_1^2]\mathrm{Tr}[k_2^2]}}\,. \tag{48}$$

$A \in [0, 1]$ is a measure of the similarity between two kernels.

It is convenient to center the kernel:

$$[K_c]_{ij} = K_{ij} - \frac{1}{m}\sum_{i=1}^{m} K_{ij} - \frac{1}{m}\sum_{j=1}^{m} K_{ij} + \frac{1}{m^2}\sum_{i,j=1}^{m} K_{ij}\,, \tag{49}$$

where m is the dimension of the kernel matrix and K_c is a positive semi-definite matrix fulfilling the condition:

$$\frac{1}{m^2}\sum_{i,j=1}^{m} [K_c]_{ij} = 0\,. \tag{50}$$

This centering is nothing but the replacement of the map $\Phi(\vec{x_i})$ by $\Phi(\vec{x_i}) - E_x[\Phi]$ where $E_x[\Phi]$ is the expected value of Φ through the set considered to construct the Kernel matrix.

Following [16] the quality in the performance of our kernel is given by the alignment with the, so called, target kernel ($K^{(Y)}$):

$$K_{ij}^{(Y)} = y_i y_j \,, \tag{51}$$

where y_i, y_j are the labels of the training points x_i, x_j.

Thus, we can center both the target and training kernel using Eq. (49) and compute the alignment as a measure of the quality of the kernels chosen. In order to do so, firstly we rewrite the alignment expression as:

$$A(K_c, K_c^{(Y)}) = \frac{\sum_{i,j=1}^{m} [K_c]_{ij}[K_c^{(Y)}]_{ij}}{\sqrt{\sum_{i,j=1}^{m}[K_c]_{ij}^2 \sum_{i,j=1}^{m}[K_c^{(Y)}]_{ij}^2}} \,. \tag{52}$$

Obviously, the larger the alignment the better the Kernel. If A is small, then the classification is poor. For the fidelity based kernel, A first grows with the lattice size, N. But then, if $N >> 1$ A decreases, which agrees with the arguments related to the orthogonality catastrophe presented in Sections 2 and 4.1. This is plotted in Fig. 16. Looking now at the alignment for the fidelity-per-site based kernel ($K^{(\lambda)}$) things are quite different. The alignment is a growing function which approaches 1 for large Ns (see Fig. 16). This ratifies the arguments presented in Section 2 based on [7, 12]. Again, for small N the alignment is smaller since the boundary is determined more accurately for a system approaching the thermodynamic limit $N \to \infty$ (see Section 2 to find the scaling arguments supporting this reasoning).

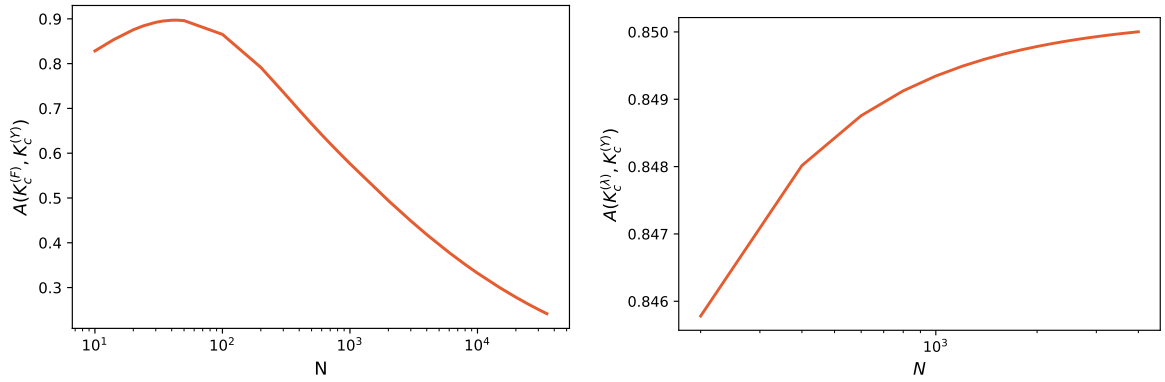

Fig. 16: (a) Alignment of ($K_c^{(F)}$) and the target centered kernel ($K_c^{(Y}$). (b) Alignment of ($K_c^{(\lambda)}$) and the target centered kernel ($K_c^{(Y}$).

A verification of all of this can be seen in Fig. 17 where $K^{(F)}$ is plotted for two sizes N. For $N = 40$, both phases are clearly distinguished. States in the same phase have a fidelity close to 1 and those in different phases, a fidelity close to 0 as expected. Nevertheless, for $N = 500$, even states in the same phase present a fidelity close to 0.

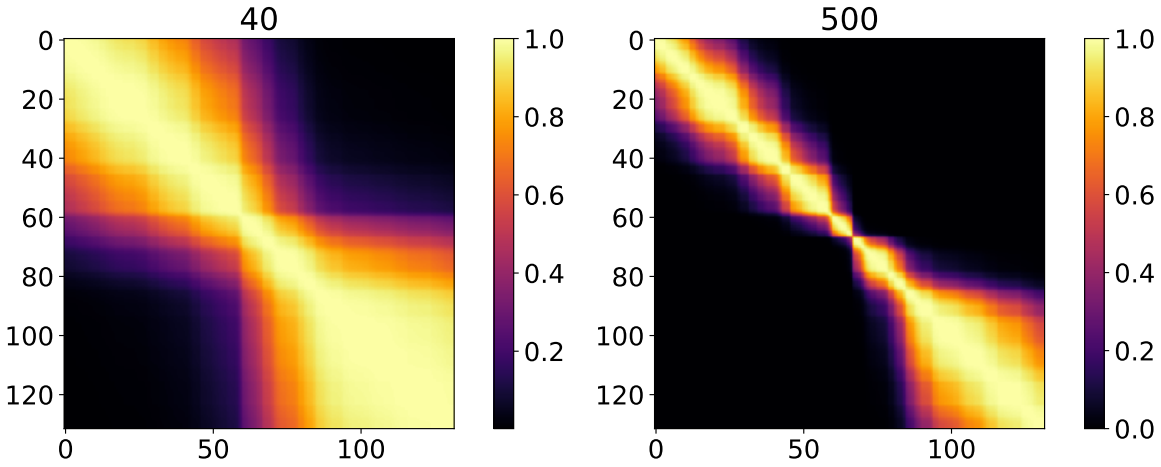

Fig. 17: Random training kernels for N=40 and 500 respectively and $(K^{(F)})$.

On the other hand, if we plot $K^{(\lambda)}$ we can clearly distinguish both phases also for large N. The kernel stays close to 1 for states in the same phase and close to 0 for those in different sides of the transition boundary.

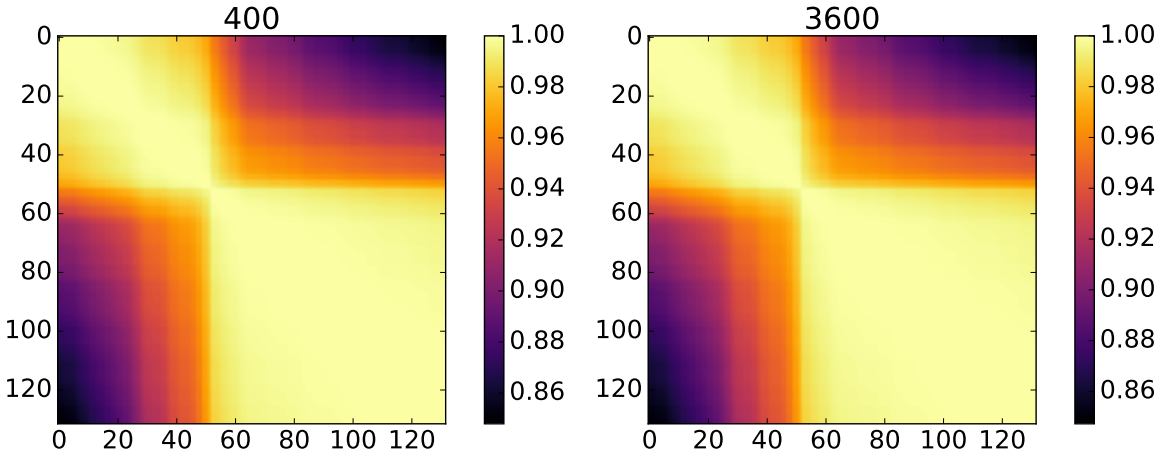

Fig. 18: Random training kernels for N=400 and 3600 respectively and $(K^{(\lambda)})$.

5 Conclusions

Through the development of this Undergraduate Dissertation we have presented an example of quantum computing supremacy in the context of quantum phase transitions. We have studied the one dimensional Ising model on its quantum version and developed two different kernels to locate the phase transition of this model using a machine learning algorithm known as Support Vector Machine.

In the first half of this Dissertation, we have presented the two pillars of this project, quantum physics and artificial intelligence. They seem to have nothing to do with each other and yet both are playing an indispensable role in the development of quantum computing. They are presented using particular examples: quantum phase transitions in the Ising model and Support Vector Machines.

After presenting these two concepts, we have searched for an example of quantum computing supremacy in the context of quantum machine learning. To do so, we have developed two different kernels to locate the phase transition of the Ising model using a SVM. Both kernels were based on an important quantum quantity known as fidelity. To fully understand the results, we have first studied the orthogonality catastrophe associated to the fidelity in high dimensional Hilbert spaces. This phenomenon is reflected in the results obtained with the first kernel presented (see Section 4.1).

Introducing the fidelity per site, we have been able to construct a new quantum Kernel capable of accurately characterizing the transition boundary, being trained exclusively with states located far away from the transition. This could be crucial when extending this technique to other non-analytically solvable models since it allows one to avoid dealing with the difficulties critical systems usually present near the transition. The validity of the method has been ratified, not only with the quantum arguments presented in Section 2 where we see how this quantity avoided the orthogonality catastrophe, but also through computer science assertions based on kernel similarities, presented in Section 4.3.

The fusion of these two branches, quantum physics and machine learning, has enabled us to fully understand the application of artificial intelligence in quantum physics and explore the immense world of quantum computing and its limitations.

6 References

[1] John Preskill. Quantum computing in the nisq era and beyond. Quantum, 2:79, Aug 2018.

[2] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. Contemporary Physics, 56(2):172–185, October 2014.

[3] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning, 2020.

[4] Cinzia Giannetti, Biagio Lucini, and Davide Vadacchino. Machine learning as a universal tool for quantitative investigations of phase transitions. Nuclear Physics B, 944:114639, July 2019.

[5] Subir Sachdev. Quantum phase transitions. Cambridge University Press, 2011.

[6] Matthias Vojta. Quantum phase transitions. Reports on Progress in Physics, 66(12):2069–2110, Nov 2003.

[7] Huan-Qiang Zhou and John Paul Barjaktarevič. Fidelity and quantum phase transitions. Journal of Physics A: Mathematical and Theoretical, 41(41):412001, September 2008.

[8] Paolo Zanardi and Nikola Paunković. Ground state overlap and quantum phase transitions. Physical Review E, 74(3), September 2006.

[9] Jacek Dziarmaga. Dynamics of a quantum phase transition: Exact solution of the quantum ising model. Physical Review Letters, 95(24), December 2005.

[10] Glen Bigan Mbeng, Angelo Russomanno, and Giuseppe E. Santoro. The quantum ising chain for beginners, 2020.

[11] SHI-JIAN GU. Fidelity approach to quantum phase transitions. International Journal of Modern Physics B, 24(23):4371–4458, September 2010.

[12] Huan-Qiang Zhou, Jian-Hui Zhao, and Bo Li. Fidelity approach to quantum phase transitions: finite-size scaling for the quantum ising model in a transverse field. Journal of Physics A: Mathematical and Theoretical, 41(49):492002, October 2008.

[13] Huan-Qiang Zhou. Renormalization group flows and quantum phase transitions: fidelity versus entanglement. arXiv preprint arXiv:0704.2945, 2007.

[14] Patrick Winston. Learning support vector machines. MIT OpenCourseWare, 2014.

[15] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. Support vector machines and machine learning on documents. In Introduction to Information Retrieval, pages 293–320. Cambridge University Press.

[16] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. pages 239–246, 08 2010.

A Code

All the development of this Undergraduate Dissertation was based on proprietary code developed by the author that can be found here.