A quantitative study of the 2D Ising model with machine learning techniques

Davide Vadacchino[1]

(with B. Lucini [2] and C. Giannetti [3])

[1] INFN - Sezione di Pisa

[2] Mathematics Department - Swansea University - UK

[3] College of Engineering - Swansea University - UK

DESY Zeuthen - February 10th, 2020

When is ML useful?

ML algorithms can be successful in cases where the task to be performed is hard or impossible to translate into a specific algorithm. For example:

- Given pictures of mushrooms, say which one is edible, which is poisonous.
- Given pictures of handwritten symbols, identify wether they correspond to letters of an alphabet.
- Discover correlations between a particular genetic signature and a particular illness.
- Extrapolate future desiderata from past purchases.

Amara's law

We tend to overestimate the effect of a technology in the short run and to underestimate its effect in the long run

Successes. . .

In many cases, these algorithms were successful:

- Sorting of mail in worldwide postal services
- Song proposal on Spotify
- AiGo won narrowly against Ke Jie, Go world champion. (He said it was an "horrible experience")

..and tragic of funny failures..

In other, funny or tragic cases:

- Amazon purchase predictor algorithm suggested phone covers featuring foot fungus images.
- Microsoft teenage chatbot "Tay.ai" on Twitter turned into "Nazi loving troll" after just one day online
- "How scientists fooled Google's AI into thinking a cat was guacamole" (InceptionV3 Google image recognition AI)

- Computational Learning Theory is a subfield of Artificial Intelligence studies. Many algorithms available: (deep) neural networks, Support Vector machines, ...
- Many ready-to-use libraries in a variety of programming languages: scikit-learn, tensorFlow, Theano, ... [Chang, Chih-Chung and Lin, Chih-Jen, 2011]
- In Physics: Several studies of ML applied to the study of phase transition are already present in the litterature. [Melko, Rogers, Carrasquilla and many others]
- In Health Sciences: Used to discover genotypes involved in drug resistance.

The archetypical problem

Identifying to which of a set of categories a new observation belongs, and to what extent.

This is a problems of Classification. ML algorithms can be classified according to:

- Type of task:
  - Supervised learning: the machine learns from labelled data(SVMs, for example), or from feedback on its action (reinforcement learning).
  - Unsupervised learning: no label is given, the machine must discover pattern in input data. (autoencoders. . .)
- Desired output:
  - Regression and Classification (SVM)
  - Clustering
  - Dimensionality Reduction
  - . . .

Our question...

What informations can we obtain on a model of statistical physics from a collection of its raw configurations at given temperatures?

- We choose to study the 2D Ising model because it is exactly solved and thus an ideal testbed for new approaches.
- We want to use one of the simplest and most transparent example of supervised learning algorithm: a Support Vector Machine. [V. N. Vapnik, A. Y. Chervonenkis '63]
- For comparison, we perform the standard multihistogram analysis on the same configurations. [Ferrenberg, Swenden '88]

Let us introduce Support Vector Machines...

Statement of the problem

We are Given a set of N *training* samples

$$(\vec{x}_1, y_1), (\vec{x}_2, y_2), \cdots, (\vec{x}_N, y_N)$$

where $\vec{x}_i \in \mathrm{R}^p$ are the input data and $y_i = \pm 1$ the class labels.

Example:

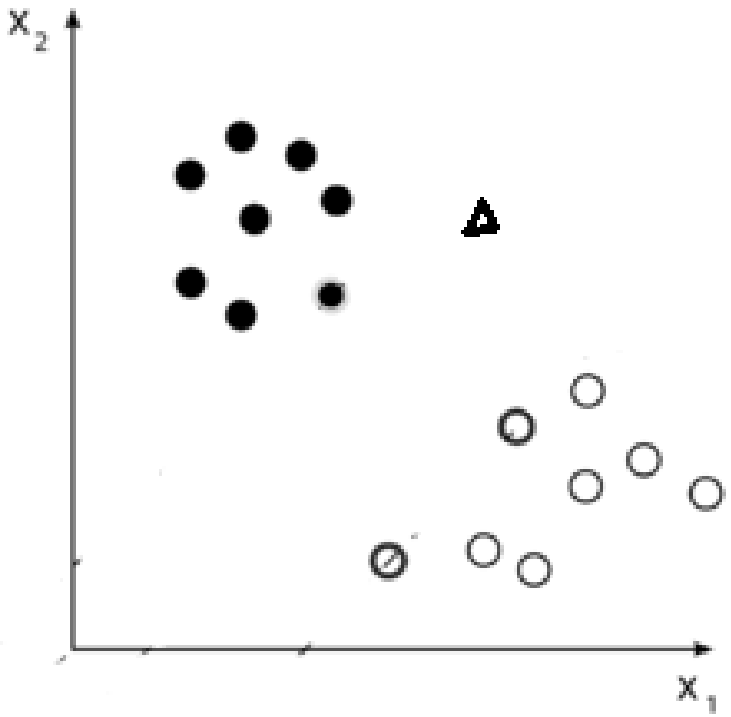
- Input data: amount of red in the pixels of a picture arranged in typographical ordering: $\vec{x} = (0.3, 0.9, 0.7, \cdots)$.
- Class: is there a dog($y = +1$) or a cat($y = -1$) in the picture?

General problem

Given the input data $\vec{x}^*$ of a new point, the test point, we want to find out its class label y^*.

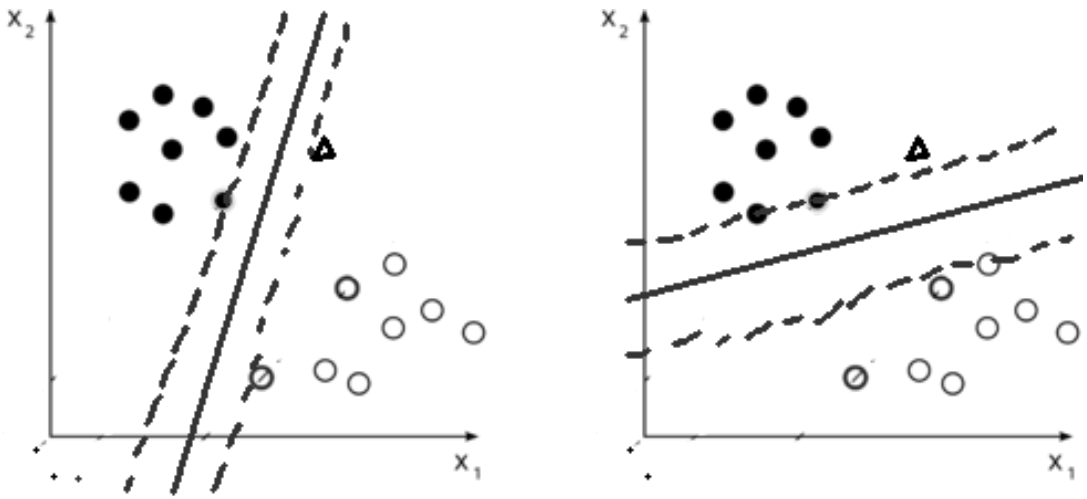
The case $p = 2$

The training and test points can be represented on the plane. What is the correct label for the test point (triangle) ?



The idea behind SVM

Draw the margin: the region delimited by the dashed lines in the figures. For each choice of the margin, a separating plane can be constructed (solid line).



Problem solved?

Once the margin and the separating hyperplane have been choosen, assign the triangle to the same class as the points lying on the same side of the separating hyperplane. (circle on the right, bullets on the left.). Which hyperplane to choose? The one with the largest margin!

Generally, an hyperplane is defined by

$$\vec{\omega} \cdot \vec{x} - b = 0$$

where $\vec{\omega}$ is the normal to the plane in $\mathbb{R}^p$ and $b \in \mathbb{R}$ is an offset.

$\vec{\omega}$ and b

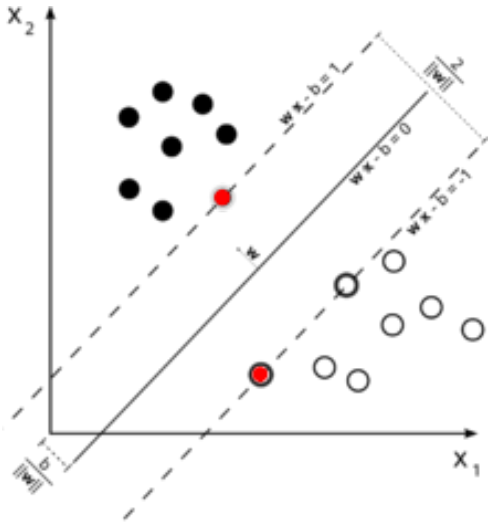
- We can scale $\vec{\omega}$ and b such that on the margin

$$\vec{\omega} \cdot \vec{x} - b = \pm 1$$

then for points on either side of the margin

$$(\vec{\omega} \cdot \vec{x} - b)y \geq 1$$

- The margin has length $2/||\vec{\omega}||$.
- The distance of the separating hyperplane from the origin is $b/||\vec{\omega}||$.



Support Vectors

The points lying on the margin boundaries (red in the picture), that directly determine the separating hyperplane, are called support vectors.

Solution of the problem

Once $\vec{\omega}$ and b are found for the maximum margin hyperplane, the distance of a new data point $\vec{x}$ from the latter is given by the decision function:

$$d(\vec{x}) = \vec{\omega} \cdot \vec{x} - b,$$

whose sign determines the classification

$$f(\vec{x}) = \text{sign}\, d\,(\vec{x})$$

Mathematically speaking. . .

This is a minimization problem for a quadratic function subject to a linear constraint: it always has a global solution if the data is linearly separable.

$$\min_{\omega, b} \frac{1}{2}||\vec{\omega}||^2 \quad / \quad y_i\,(\vec{\omega} \cdot \vec{x}_i - b) \geq 1,\ \forall i$$

where i runs over the training set.

When the data are not linearly separable...

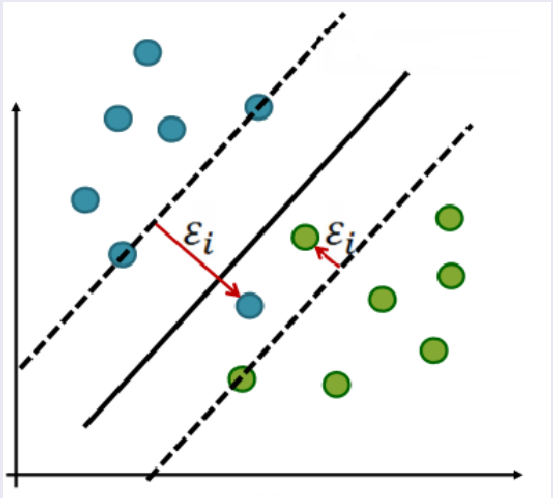
...we might still find a solution if we let some of the points into the margin,

$$y_i\left(\vec{\omega}\cdot\vec{x}_i - b\right) - 1 \leq 0$$

we then obtain a soft margin classifier. We introduce the slack variables ζ_i

$$\zeta_i \geq 1 - y_i\left(\vec{\omega}\cdot\vec{x}_i - b\right), \quad \forall i$$

when $\zeta_i = 0$, $\forall i$, we have the hard margin classifier again.



For a soft margin classifier

The minimization then problem becomes

$$\min_{\omega,b}\left(\frac{1}{2}||\vec{\omega}||^2 + C\frac{1}{N}\sum\zeta_i\right) \quad / \quad y_i\left(\vec{\omega}\cdot\vec{x}_i - b\right) \geq 1 - \zeta_i, \text{ and } \zeta_i \geq 0, \forall i$$

The slack variables introduce a penalty for points that fall into the margin. The parameter C weights the penalty of letting points into the margin.

Primal problem

By introducing the Lagrande multipliers α_i and γ_i,

$$L = \frac{1}{2}||\vec{\omega}||^2 + \frac{C}{N}\sum_{i=1}^{N}\zeta_i + \sum_{i=1}^{N}\alpha_i\left(1 - y_i\left(\vec{\omega}\cdot\vec{x}_i - b\right)\right) - \sum_{i=1}^{N}\gamma_i\zeta_i \ ,$$

where $\gamma_i, \alpha_i \geq 0$ because of the inequality constraints[a].
Setting the gradients of L to zero:

$$\nabla_\omega L = \vec{\omega} - \sum_i \alpha_i y_i \vec{x}_i = 0$$

$$\nabla_b L = -\sum_i \alpha_i y_i = 0$$

$$\nabla_\gamma L = \frac{C}{N} - \alpha_i - \gamma_i = 0$$

and substituting back in $L\ldots$

[a] KKT conditions

Dual problem

. . . we obtain the *quadratic program*: find the α_i's such that

$$L = -\frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \vec{x}_i \cdot \vec{x}_j + \sum_i \alpha_i, \quad \alpha_i, \gamma_i \geq 0$$

is *minimized*, with $0 \leq N\alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0$.

Note that:

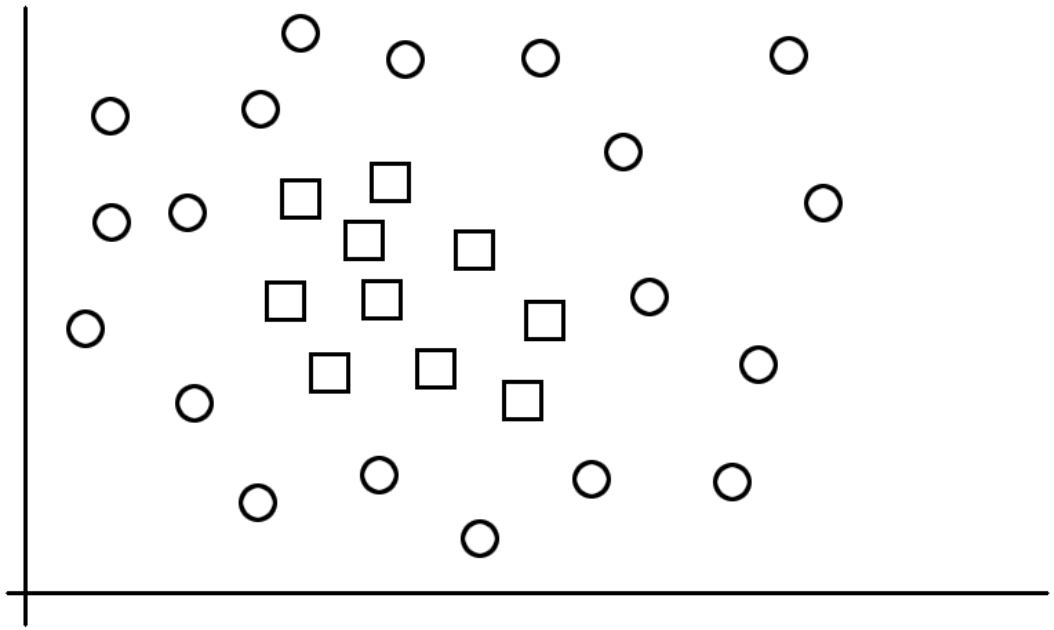
- The minimization problem only depends on scalar products between training data points.
- The vector $\vec{\omega}$ can be expressed from the training data,

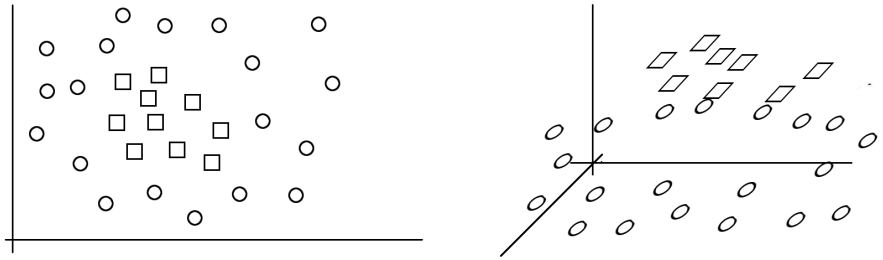
$$\vec{\omega} = \sum_i y_i \alpha_i \vec{x}_i \ .$$

- The problem consists in finding the α_i as described above. Once this is done, we can classify a new sample $\vec{x}$ by computing,

$$d(\vec{x}) = \sum_i y_i \alpha_i \vec{x}_i \cdot \vec{x} - b \ .$$

The dual formulation is very useful for an additional reason. What if the training data are as shown below ?





A non-separable problem in input space might still have a solution in a new space through the feature map

$$\Phi : \vec{x} \longrightarrow \Phi(\vec{x})$$

In the case above $\vec{x} \in \mathbb{R}^2$, $\Phi(\vec{x}) \in \mathbb{R}^3$. In the new space, we might recover linear separability.
The new minimization problem is

$$L = -\frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) + \sum_i \alpha_i$$

with $0 \leq N\alpha_i \leq C$ and $\sum \alpha_i y_i = 0$. The vector $\vec{\omega} = \sum_i y_i \alpha_i \ \phi(\vec{x}_i)$ is now a vector in $\mathbb{R}^3$ and the decision function is

$$d(\vec{x}) = \sum_{i=1}^{N} y_i \alpha_i \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) - b$$

Note that:

- The minimization problem and the decision function only depend on scalar products in feature space.
- In principle, Φ could have a very large number of components, in which case the problem could become computationally very demanding.

Say there exist a function $\mathcal{K}$, called kernel, such that

$$\mathcal{K}\left(\vec{x}, \vec{x}'\right) = \Phi(\vec{x}) \cdot \Phi(\vec{x}')$$

then, the minimization problem becomes

$$L = -\frac{1}{2}\alpha^T y^T \mathcal{K}\left(\vec{x}_i, \vec{x}_j\right) y\alpha + \alpha^T e, \qquad \alpha^T y = 0,\ 0 \leq N\alpha_i \leq C$$

Where the notation has been made more compact, and e is a vector with 1 in every component.
The decision function becomes

$$d(\vec{x}) = \sum_{i=1}^{N} y_i \alpha_i \mathcal{K}(\vec{x}_i, \vec{x}) - b$$

- To be accceptable, a Kernel must be symmetric and positive semi-definite. (Mercer's condition).
- The kernel is of crucial importance: it is the informational bottleneck of the system.

Some examples of Kernels:

- Linear, $\mathcal{K} = \vec{x} \cdot \vec{x}'$

$$\Phi(\vec{x}) = (x_1, x_2, x_3, \cdots)$$

- Polynomial, $\mathcal{K} = (c_o + \vec{x}_i \cdot \vec{x}_j)^d$. In the case $d = 2$,

$$\Phi(\vec{x}) = (x_1^2, \cdots, x_N^2, x_1 x_2, \cdots, x_{N-1} x_N, c_0 x_1, \cdots, c_0 x_N)$$

- Gaussian Kernel. In this case Φ has an infinite number of components.

$$\mathcal{K}(\vec{x}, \vec{x}') = \exp - \frac{||\vec{x} - \vec{x}'||^2}{2\sigma^2}$$

Note that

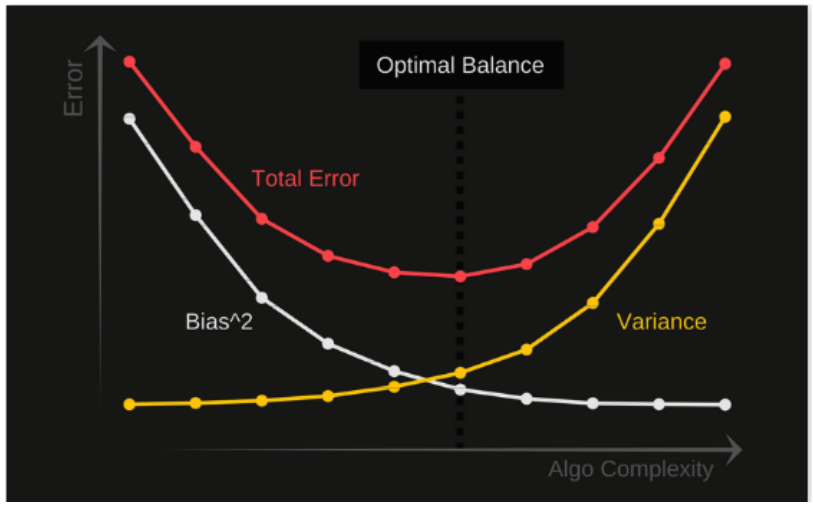
- In each case, d, c_0 and σ are hyperparameters to be fixed to maximize the quality of the classifier or using some prior knowledge on the system.
- Note that complicated kernels can be obtained from simple ones as long as Mercer's condition is respected:

Bias vs. Variance tradeoff

In choosing a Kernel we might incur in two opposite problems:

- Underfitting: simple Kernel, large classification error. There are not enough *free parameters* to fit. The bias is *large*.
- Overfitting: complex[a] Kernel, small classification error. There are so many parameters that *they fit to the noise*.

[a] As in *complicated*



How to measure the quality of our machine?

Cross Validation

Partition the original training data into a training set and a validation set. Count the ratio R of errors of prediction. Repeat with a different partition.

As a result, we obtain the Score:

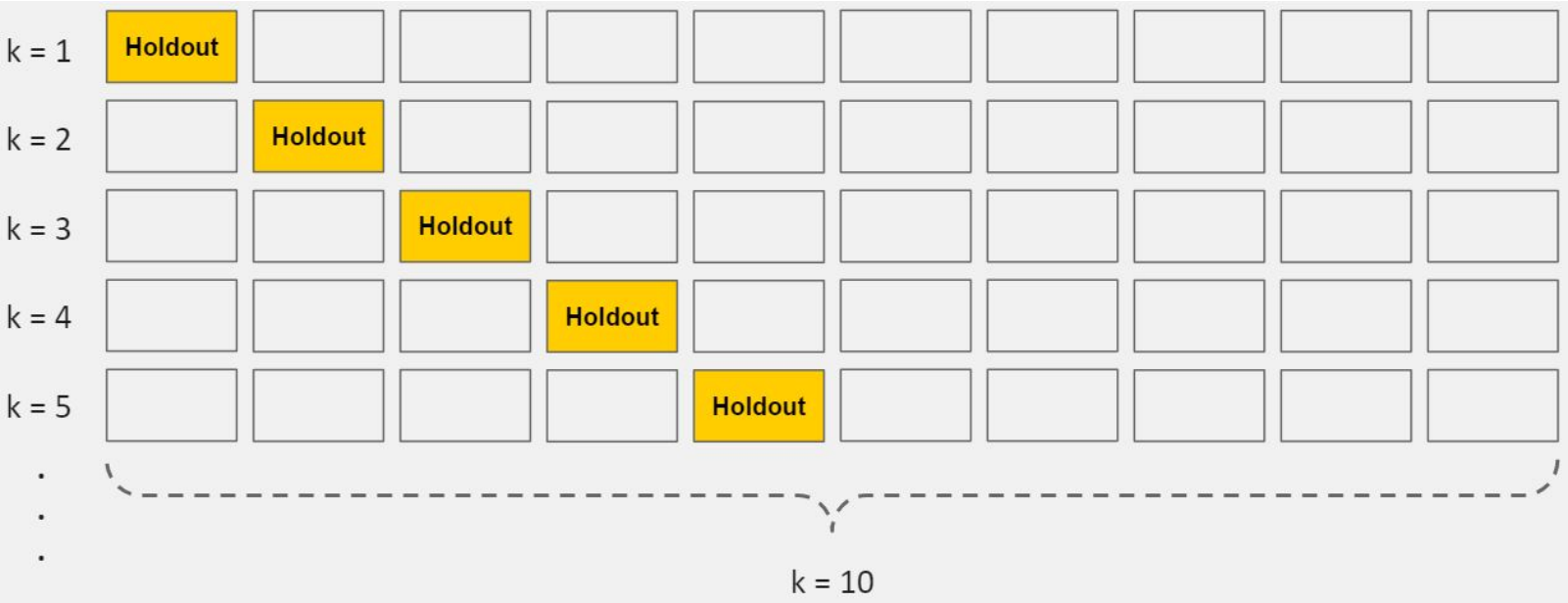
$$S = 1 - R$$

which is a way to evaluate how well the results of this statistical analysis will generalize to unknown samples.

- $S \sim 1$: good, the machine will correctly predict the class of new samples
- $S \sim 0.5$: bad, the machine will fail on most new samples.
- Intuitively, if one removes a point which is not a support vector, nothing should change. Thus,

$$R \leq \frac{\langle n_{SV} \rangle}{m} ,$$

  where m is the number of data points in the training set. Thus $\langle n_{SV} \rangle / m$ gives a rough comparative metric of the quality of classification.
- Other similar (and useful) inequalities can be shown to hold in Statistical Learning Theory.



Many different implementations:

- Leave-One-Out: use one data point as the validation set, the rest as training set.
- k-fold: partition the original data into k equal size subsamples, use one of these in turn as validation set.
- Stratified k-fold: same as before but respecting the distribution over the class labels.

SVM can be seen as the result of *empirical risk minimization*. Say we have a collection of data points X, with labels Y, related by the probability distribution $P(X, Y)$. Then, given a model f, we can compute its expected risk

$$R[f] = \int dP(X, Y)\mathbb{V}(Y, f(X)) \tag{1}$$

where $\mathbb{V}$ is the loss function that we assume convex. Examples are

- Indicator function (NP-Hard problem!): Statistics and Decision Theory.
- Quadratic: Least squares.
- Hinge loss: SVM with soft margin,

$$\mathbb{V}(Y, \operatorname{sign} d(X)) = \max(0, 1 - Y \operatorname{sign} d(X))$$

In this case $\mathcal{F}$ can be seen as the class of Kernels we choose.

The best model is found as

$$f^\star = \arg\min_{f \in \mathcal{F}} R[f]$$

- Overfitting if $\mathcal{F}$ is too large
- Underfitting if $\mathcal{F}$ is too small

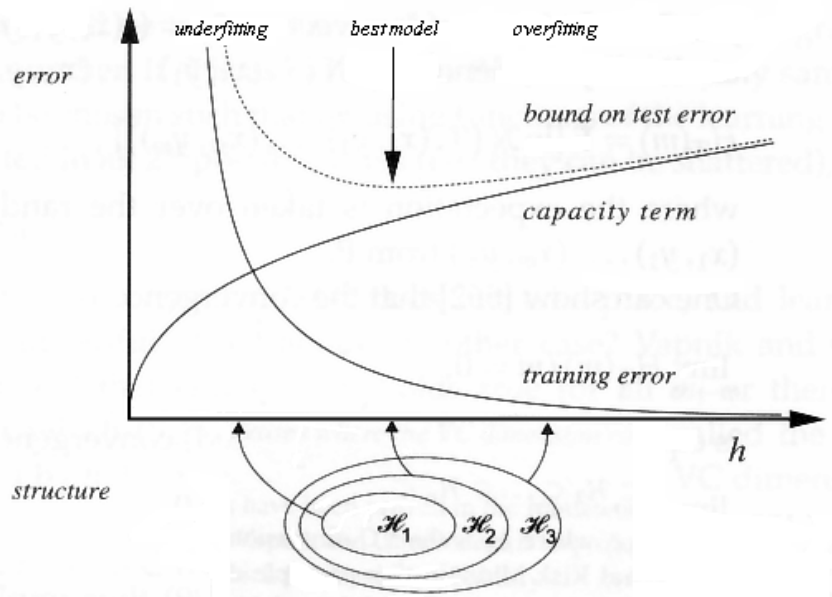
We define the empirical risk as

$$R_{\text{emp}}[f] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{V}(Y_i,\, f(X_i)) \ ,$$

And it can be shown that the LOO cross-validation estimate of R is a unbiased estimator of the Risk computed on $n-1$ samples,

$$\langle R_{\text{LOO}}[f_n] \rangle = \langle R[f_{n-1}] \rangle \ .$$

Scholkopf, Smola



Structural Risk Minimization

A possible strategy to choose the model that neither overfits nor underfits is to progressively restrict the complexity of the class of models $\mathcal{F}$ in which we look for the optimum. Vapnik, Chervonekis

The (ferromagnetic) Ising model is defined by the Hamiltonian,

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \qquad J > 0$$

where $\langle i,j \rangle$ denotes the sum over next neighbours and $\sigma_i = \pm$.
For this study, a square lattice and $D = 2$, then the model is exactly solved and :

- At $T_c = 2/\ln\left(1+\sqrt{2}\right)$, there is a second order phase transition separating an ordered from a disordered phase.
- The order parameter associated to the transition is

$$m = \frac{1}{L^2} \sum_{i=1}^{L^2} \sigma_i$$

- The critical exponents are $\nu = 1$ and $\gamma = 7/4$. Using the hyperscaling relations, all the other exponents can be computed.

- Several values of the linear size L were explored: $L = 128$, 240, 360, 440, 512, 760, 1024.
- A rough scan in the temperature was performed, and then refined after evaluating the magnetization and its susceptibility.
- $N = 200$ decorrelated configurations were collected for later analysis.
- The Wolff algorithm was used to avoid critical slowing down.
- The magnetic susceptibility χ^2 could be estimated at any intermediate T using the multihistogram method.
- The finite size behaviour of the pseudo-critical temperature $T_c(L)$ and the critical magnetic susceptibility $\chi^2_c(L)$

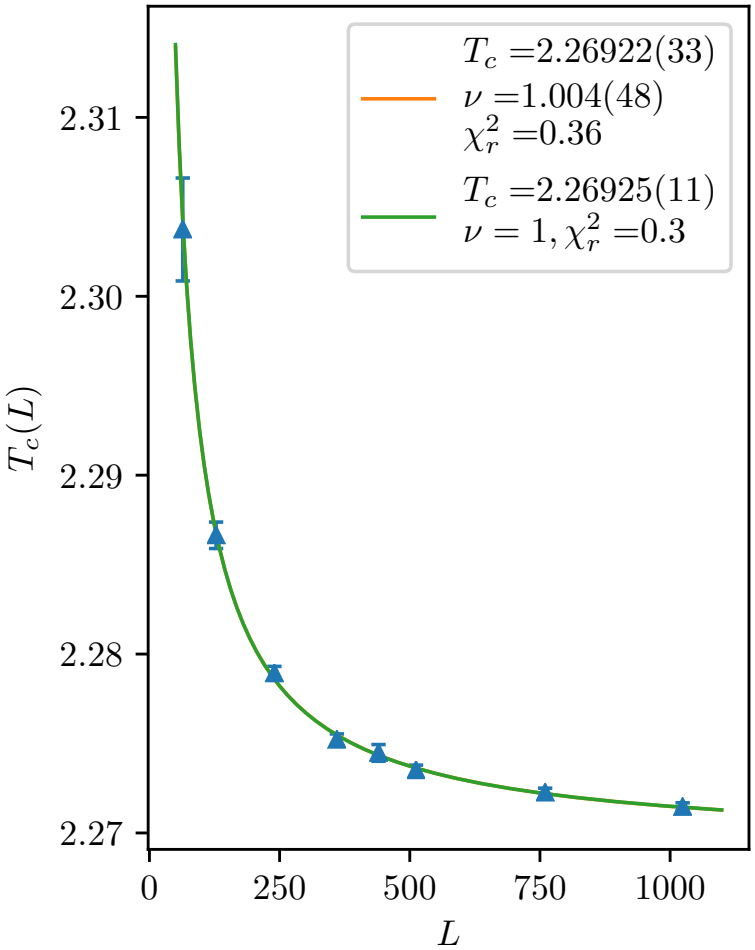
$$T_c(L) - T_c(\infty) \propto L^{1/\nu}, \quad \chi^2_c(L) \propto L^{\gamma/\nu}$$

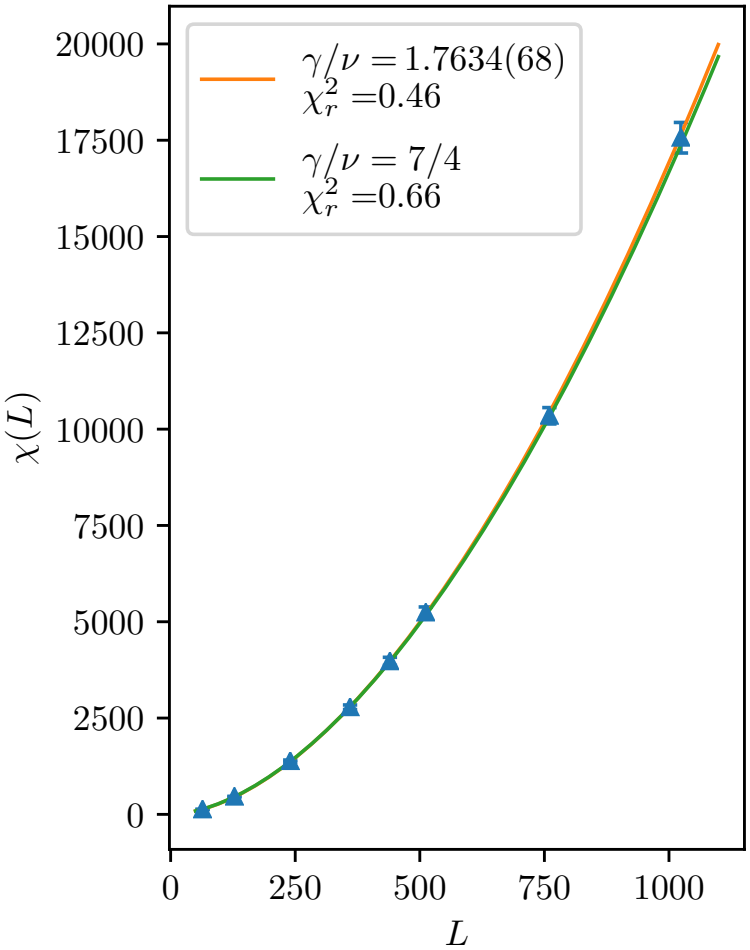
allows us to estimate T_c, ν and γ/ν. The rest of the indices can be obtained with the hyperscaling relations.

L	T_{min}	T_{max}	n_{steps}
64	2.280	2.330	20
128	2.275	2.294	20
240	2.273	2.285	24
360	2.270	2.280	20
440	2.270	2.280	20
512	2.2665	2.2770	22
760	2.27000	2.27400	20
1024	2.27000	2.27300	30

L	T_c	χ_{max}
64	2.3037(29)	$1.284(37) \cdot 10^2$
128	2.28664(74)	$4.590(97) \cdot 10^2$
360	2.27528(28)	$2.781(65) \cdot 10^2$
440	2.27448(47)	$3.97(10) \cdot 10^3$
512	2.27351(29)	$5.24(14) \cdot 10^3$
240	2.27892(39)	$1.383(28) \cdot 10^3$
760	2.27226(25)	$1.035(21) \cdot 10^4$
1024	2.27145(23)	$1.757(40) \cdot 10^4$

Table: On the left, scanning windows of temperatures for extracting the pseudocritical temperature $T_c(L)$ at each value of L; n_{steps} indicates the number of simulated values of T, all equally spaced between the two extremes T_{min} and T_{max}. On the right, values of the pseudocritical temperature T_c and the corresponding maximum of the magnetic susceptibility, as obtained from the multi-histogram method.


T_c =2.26922(33)
ν =1.004(48)
χ^2_r =0.36
T_c =2.26925(11)
$\nu = 1, \chi^2_r$ =0.3
$T_c(L)$
L
2.31
2.30
2.29
2.28
2.27
0
250
500
750
1000


$\gamma/\nu = 1.7634(68)$
χ^2_r =0.46
$\gamma/\nu = 7/4$
χ^2_r =0.66
$\chi(L)$
L
20000
17500
15000
12500
10000
7500
5000
2500
0
0
250
500
750
1000

By fitting the following scaling laws,

$$T_c(L) - T_c(\infty) \propto L^{-1/\nu}, \quad \chi_c^2(L) \propto L^{\gamma/\nu}$$

to the data, one obtains:

T_c	ν	χ_r^2	γ/ν	χ_r^2
2.26922(33)	1.004(48)	0.36	1.7634(68)	0.46
2.26925(11)	1 (exact)	0.3	7/4 (exact)	0.66

where $\chi_r^2 = \chi^2/d.o.f$, and:

- In the first row, the fits are performed with T_c, ν and γ as fittings parameters.
- In the second raw, the fits are performed with ν and γ fixed at their known value, and T_c used as a fitting parameter.

Our program

We want to use the same configurations produced for the "standard" study and analyze the data with a SVM with a minimal set of assumptions.

Our claim:

Using a SVM, we can obtain estimates of the critical temperature, the corresponding exponents and the global symmetry of the system.

Our only assumption

There is a second order phase transition somewhere in the probed temperature range.

- We will need a Training set: we take the 200 configurations at T_o and 200 at $T_d > T_o$. These training temperatures can be considered as additional hyperparameters.
- We classify 200 configurations at each intermediate temperature T using homogeneous polynomial kernels of degrees $n = 1, 2, 3 \ldots$. The degree of the polynomial kernel can be considered as an hyperparameter.
- The choice of using homogeneous polynomial kernels is less restrictive than it seems.

We studied

For each pair T_o, T_d, for each L and for each n:

- The decision function $d(\vec{x})$:
  1. Its dependence on T_o and T_d.
  2. Its value calculated on configurations at temperature T, for $T_o \leq T \leq T_d$.
- The score S that measures the quality of the classifier.
- The ratio of number of support vectors n_{SV} to the number of training points m.

For convenience, we studied

$$\tilde{d}(\vec{x}) = \frac{1}{2} d(\vec{x}) - b \ . \tag{2}$$

The training temperatures

- The training temperatures we initially chosen as far from each other as possible, i.e.

$$T_o = 0.5, \qquad T_d = 5.0$$

- In the following, unless mentionned, we always verified that the final results were independent of the choice of T_o and T_d.

The estimate of $\langle \tilde{d} \rangle$

- At each intermediate temperature T, $\langle \tilde{d} \rangle$ was obtained as an average the 200 configurations.

$$\langle \tilde{d} \rangle = \frac{1}{200} \sum_{j=1}^{200} \tilde{d}(\vec{x}_j) \ .$$

  where j labels the configuration.

- Its error was computed by resampling.

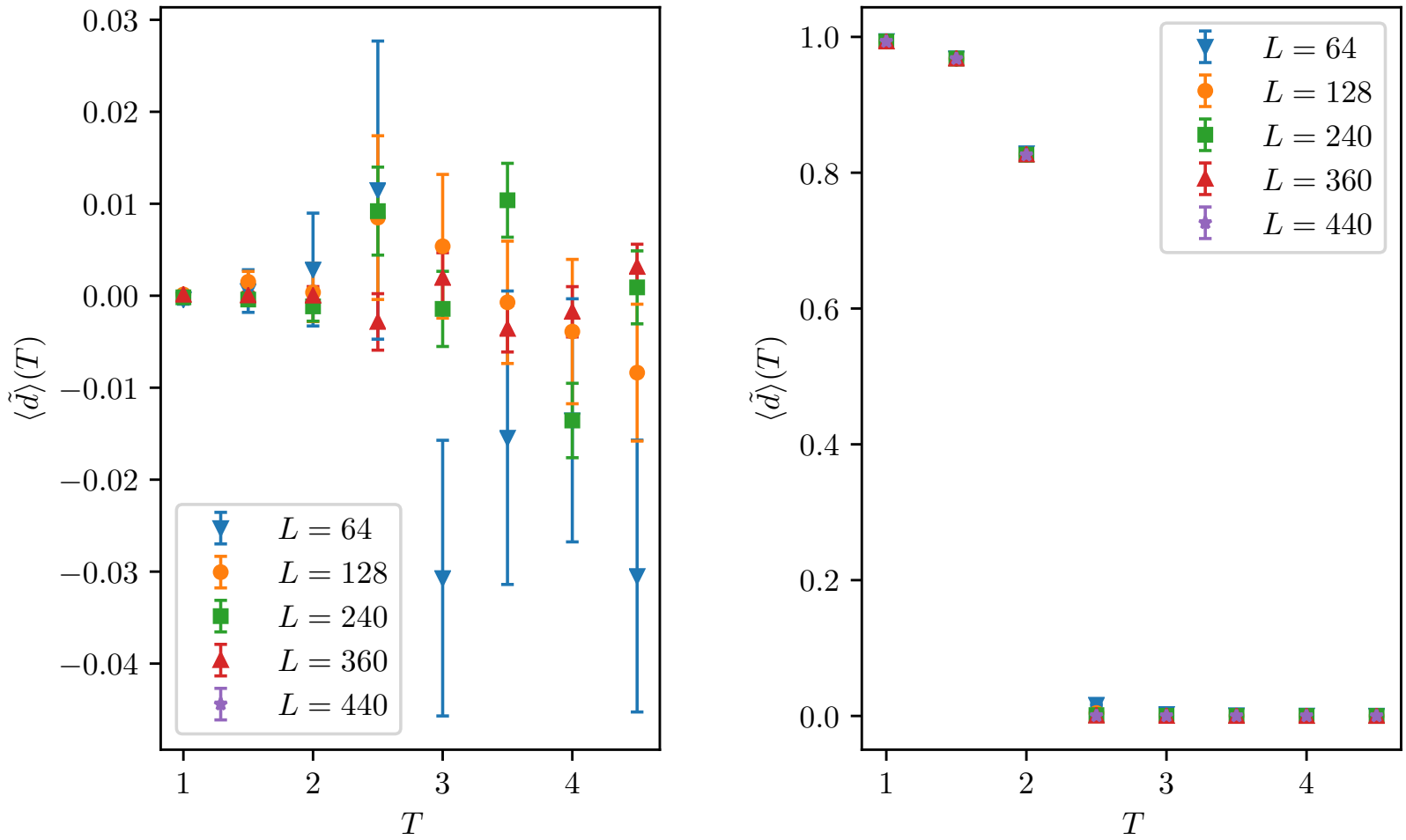


Figure: Odd power kernels on the left hand side, even power kernels on the right hand side. Larger L's behave similarly and are not included to avoid overcrowding the plots.

Very different results for even and odd power kernels

Even powers

- All the even power kernels behave as we expect

$$T \to T_o, \tilde{d} \to 1, \text{ and } T \to T_d, \tilde{d} \to 0 \ .$$

- The neighbourhood of $T \sim 2.5$ will be called critical region.

Odd powers

- As a function of T, no clear trend.
- The values of $\langle \tilde{d} \rangle$ seem to be always clustered around 0, the machine seems not to be able if the analyzed T is closer to T_o than to T_d.

The results are largely independent of the choice of T_o and T_d provided these are far from the critical region, in the case of even power kernels. Questions:

- What is the physical meaning of $\tilde{d}$ in the even power case?
- How does $\tilde{d}$ depend on T_o and T_d?
- Why the odd/even power kernels behave so differently?

Any even power kernel can be obtained from the quadratic one, any odd from the linear one. For now, We restrict our attention to the quadratic kernel

$$\mathcal{K}(\vec{x},\,\vec{x}') = \left(\frac{\vec{x}\cdot\vec{x}'}{L^2}\right)^2 ,$$

where L^2 has been introduced so that for identical configurations $\mathcal{K}(\vec{x},\,\vec{x}) = 1$.

The Quadratic kernel

$$\tilde{d}(\vec{x}) = \frac{1}{2}\sum_{i=1}^{n_{SV}} y_i\alpha_i\ \mathcal{K}\left(\vec{x}_i,\vec{x}\right) = \frac{1}{2L^4}\sum_{i=1}^{n_{SV}} y_i\alpha_i\ \left(\sum_{\vec{a}} x_i(\vec{a})\,x(\vec{a})\right)^2 ,$$

now define

$$\overline{C}(\vec{a},\,\vec{b}) = \frac{1}{L^4}\sum_{i=1}^{n_{SV}} y_i\alpha_i\,x_i(\vec{a})\,x_i(\vec{b}) \ .$$

then

$$\tilde{d}(\vec{x}) = \frac{1}{2}\sum_{\vec{a},\vec{b}} \overline{C}(\vec{a},\,\vec{b})\,x(\vec{a})\,x(\vec{b}) \ ,$$

The quantity $\overline{C}(\vec{a},\vec{b})$ can be interpreted as an *effective coupling* between two spins at positions $\vec{a}$ and $\vec{b}$ [Melko, Ponte]

If we pick T_o and T_d in various ways, and represent $\langle \overline{C}(\vec{0}, \vec{b}) \rangle \ldots$

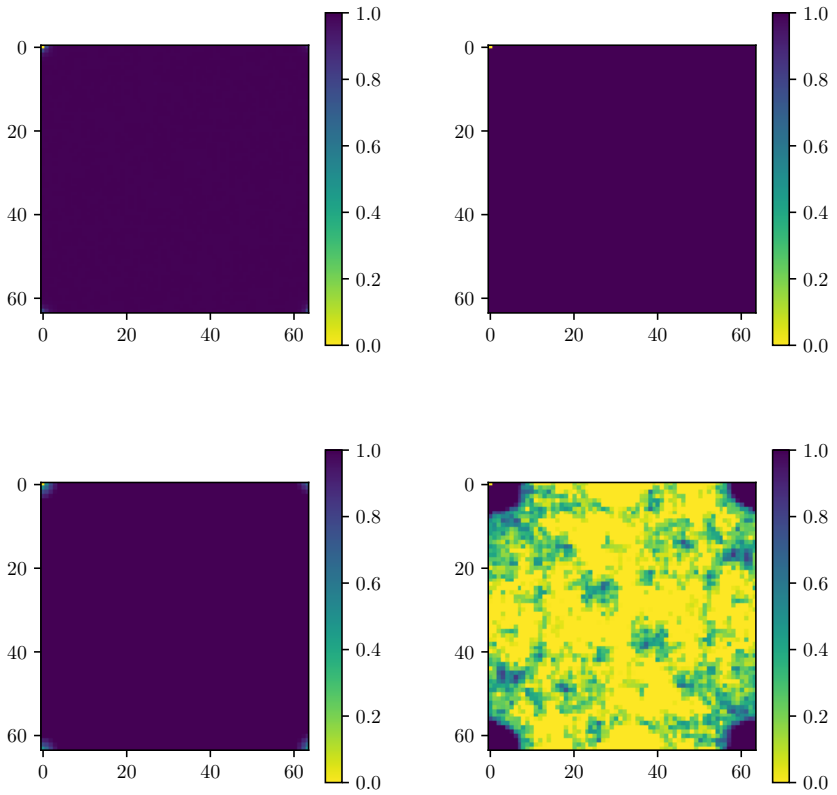


Figure: In typographical ordering: $(T_o,\ T_d) = (0.5,\ 5.0)$, $(T_o,\ T_d) = (0.5,\ 2.0)$, $(T_o,\ T_d) = (2.0,\ 2.5)$, $(T_o,\ T_d) = (3.0,\ 5.0)$.

- When T_o and T_d are both above the critical region, $\tilde{d}$ is a short ranged version of m^2
- If at least one of the training temperatures is below the critical region, then

$$\tilde{d} \propto m^2$$

  where m is the magnetization.
- These results are mostly independent on the choice of C.

This will eventually allow us to obtain the critical temperature and the critical exponents.

What happens with other Kernels, why should we discard them? Remember now two useful informations to estimate the generalization ability of our machine:

- The score $S = 1 - R$ where R is the classification error on a known test set
- The ratio $\langle n_{\mathrm{SV}} \rangle / m$, where m is the number or data points in the training set, n_{SV} the number of support vectors.
- The inequality

$$R < \frac{\langle n_{\mathrm{SV}} \rangle}{m}$$

With this, we did the following:

- Picked all possible pairs $(T_o,\ T_d)$ and train a SVM.
- Performed a stratified 10-fold cross validation procedure in which n_{SV} and R were estimated.
- Repeated at every value L of the linear size of the system.

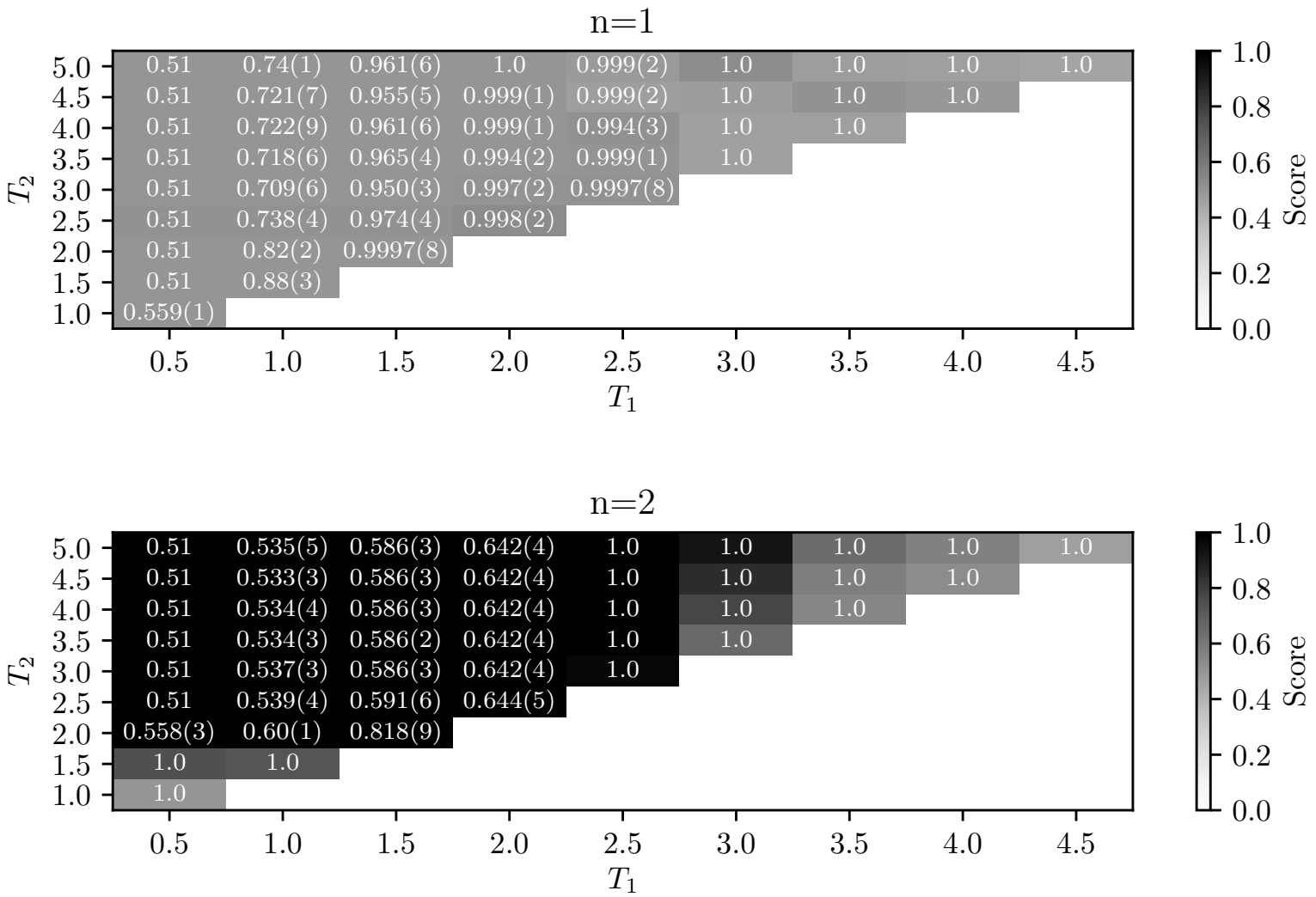


Figure: Score (grayscale) and value of $\langle n_{SV} \rangle / n$ for each $(T_o,\ T_d)$ at $L = 128$, $n = 1$ above, $n = 2$ below.

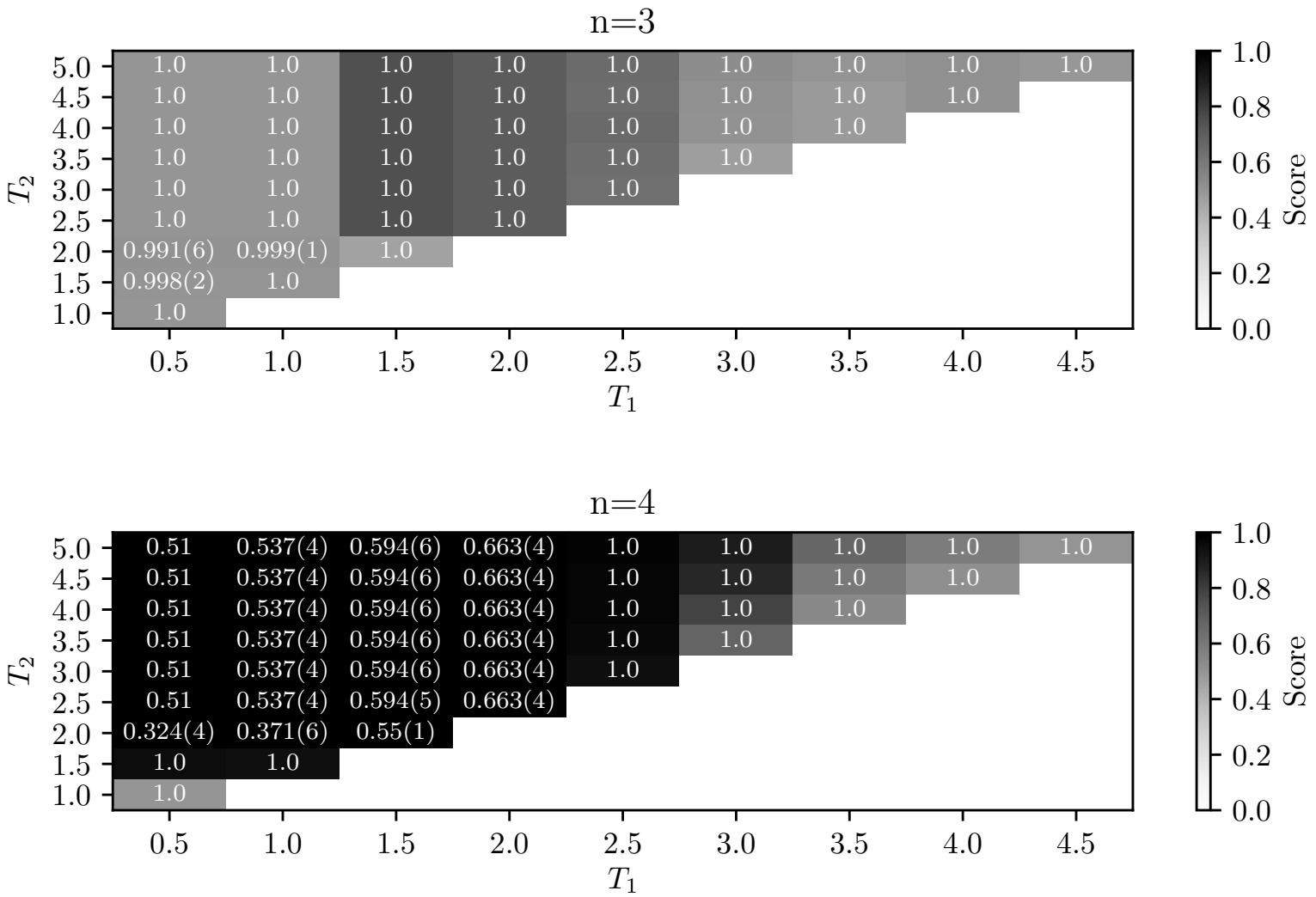


Figure: Score (grayscale) and value of $\langle n_{SV} \rangle / n$ for each $(T_o,\ T_d)$ at $L = 128$, $n = 3$ above, $n = 4$ below.

- A lot of gray on the $n = 1$ case: the score is poor (remember that a score of 0.5 is the poorest possible)
- For $n = 2$, good score with $T_o < 2.5$ and $T_d > 2.5$.
- The ratio $\langle n_{SV} \rangle / n$ is generally smaller for $n = 2$ and especially for $T_o < 2.5$ and $T_d > 2.5$.
- Even power kernels behave like $n = 2$, odd ones like $n = 1$.

What do we learn?

- We might as well ignore all powers other than $n = 1$ and $n = 2$.
- The machine performs better if T_o and T_d are chosen on either side of the critical region.

Question

Why does the $n = 2$ Kernel perform so much better than the one with $n = 1$?

Symmetry!

Assume that we know that the input data must be symmetric with respect to some transformation.
Examples:

- Translation invariance if a particular object can be anywhere in a picture.
- Rotation-Reflection invariance for a circular object
- Internal invariance with respect to some transformation.

Then various approaches are possible

- *Filter* the data so that the input are symmetric.
- *Restrict* the class of Kernels to choose from so that they reflect the symmetry. [1]

Symmetry in input data

Assume that the input data is symmetric with respect to the action of a symmetry group G of elements g,

$$\vec{x} \to \vec{x}' = g\vec{x} = (gx_1, gx_2, \cdots, gx_N) \ .$$

We can ask the decision function to be invariant,

$$d(g\vec{x}) = d(\vec{x}) \quad \forall g \in G, \vec{x} \ ,$$

And, as a consequence, the kernel must be Totally Invariant (TI),

$$\mathcal{K}(\vec{x}, g\vec{x}') = \mathcal{K}(g\vec{x}, \vec{x}') = \mathcal{K}(\vec{x}, \vec{x}') \quad \forall g \in G, \vec{x}, \vec{x}'$$

[1] Remember Structural Risk Minimization?

To implement our prior knowledge about the symmetry, we thus require:

- Total invariance of the Kernel.
- Completeness of the set of invariant features: every pair of orbits of the group should be distinguishable by using the features in the set.

It can be shown that a TI kernel that also satisfies the second criteria can be obtained by projecting on the group

$$\mathcal{K}_G(\vec{x}, \vec{x}') = \frac{1}{|G|^2} \sum_{g,g' \in G} \mathcal{K}(g\vec{x}, g'\vec{x}')$$

Group projection

For a generic function $f(\vec{x})$, we average over the action of the group

$$\tilde{f}(\vec{x}) = \frac{1}{|G|} \sum_{g \in G} f(g\vec{x})$$

where $|G|$ is the order of the group. This operation is a projection in the mathematical sense.

There are two extreme cases:

- If $\mathcal{K}$ is already invariant, then $\mathcal{K}_G = \mathcal{K}$.
- If the projection is null, only noise will be fitted.

- In the case of the Ising model, the global symmetry is $\mathbb{Z}_2$

$$g\vec{x} = -\vec{x}$$

- Odd homogeneous polynomial will be projected to 0, even homogeneous polynomial will be projected to themselves.
- In statistical learning theory, one can show that the expected error of classification is reduced by the projection above if the dataset shares the symmetry.

$$R[\,\mathcal{P}_G[f]\,] \leq R[f]$$

where $\mathcal{P}_G$ is the projection on the known symmetry group of the data.

- The behaviour of the score S and of the ratio $\langle n_{SV} \rangle / m$ is a strong indication that the symmetry of the model, which was a priori unknow to us, is $\mathbb{Z}_2$.

Disclaimer

The rigorous mathematical implication is:

- Symmetry implies higher score

and not:

- Higher score implies symmetry

This information can however be used comparatively.

Our strategy

Since $\tilde{d} \propto m^2$, the variance (susceptibility) σ_d of $\tilde{d}$ must reach a maximum at the pseudocritical temperture $T_c(L)$. Thus:

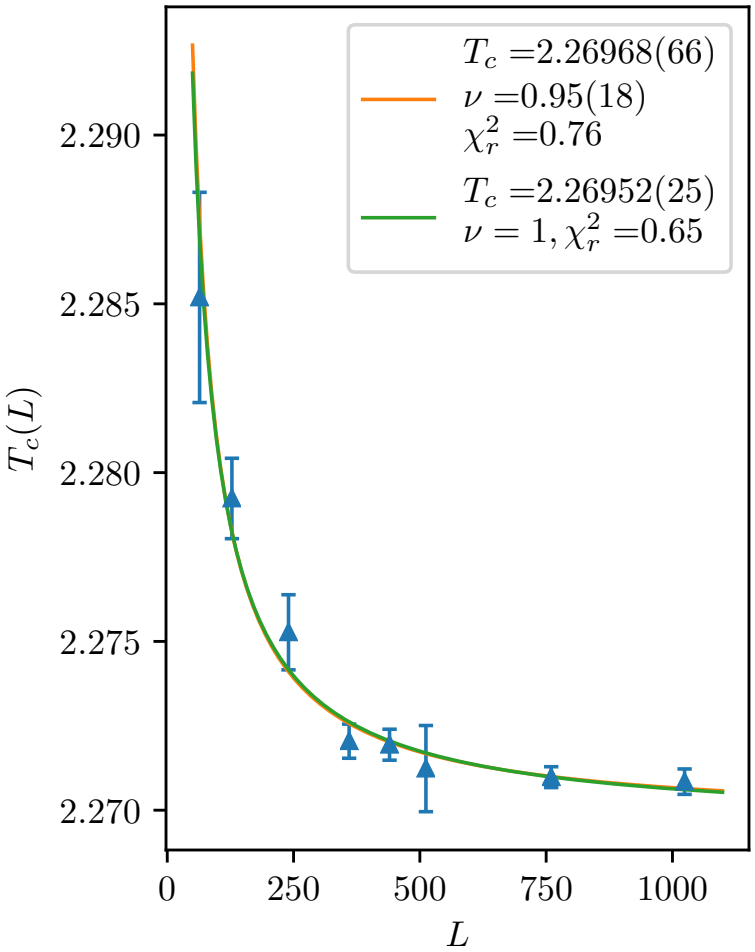
- At each volume L, compute $\langle \tilde{d} \rangle$ and its error/susceptibility σ_d over the range of T's and extract the location and value of the maximum of σ_d.
- Fit the scaling behaviour of $T_c(L)$ and $\sigma_{\mathrm{d,max}}(L)$ with the appropriate power law.

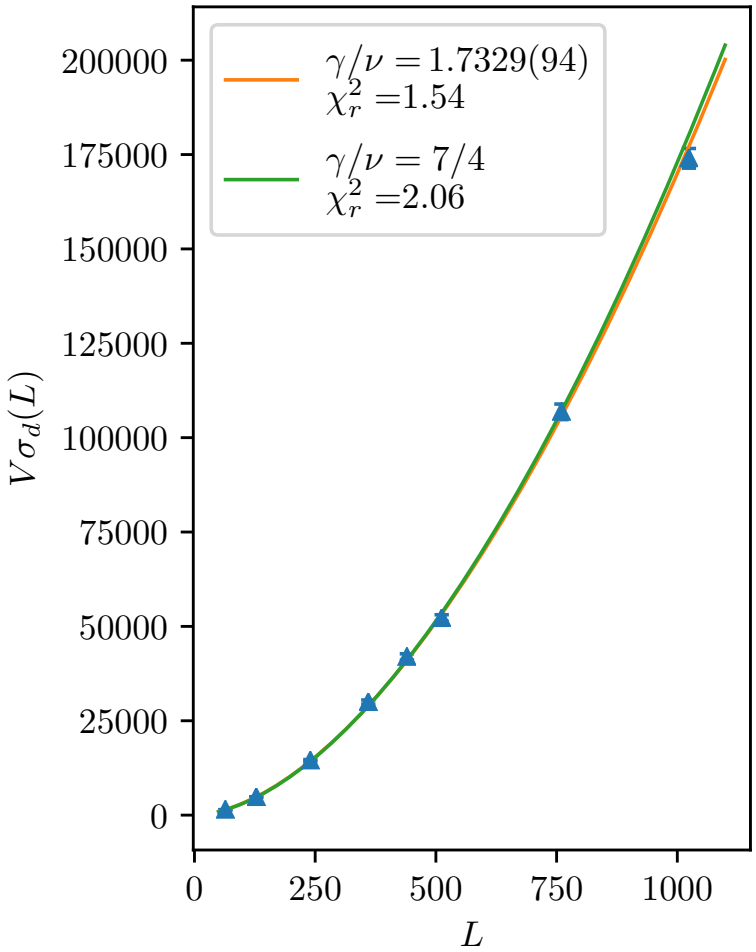
$$T_c(L) - T_c(\infty) \propto L^{1/\nu}, \quad V\sigma_{\mathrm{d,max}}(L) \propto L^{\frac{\gamma}{\nu}}$$

L	$T_c(L)$	$V\sigma_d$
64	2.2852(31)	$1.426(31) \cdot 10^3$
128	2.2792(12)	$4.782(85) \cdot 10^3$
240	2.2753(11)	$1.448(24) \cdot 10^4$
360	2.27204(51)	$2.995(55) \cdot 10^4$
440	2.27194(46)	$4.193(82) \cdot 10^4$
512	2.2712(13)	$5.221(87) \cdot 10^4$
760	2.27098(31)	$1.068(21) \cdot 10^5$
1024	2.27085(38)	$1.740(26) \cdot 10^5$

Table: Position ($T_c(L)$) and volume-multiplied value ($V\sigma_d$) of the maximum of the decision function error at each investigated lattice size L.

Computation of T_c and ν





T_c	ν	χ^2_r	γ/ν	χ^2_r
2.26968(66)	0.95(18)	0.79	1.733(10)	1.54
2.26954(25)	1 (exact)	0.65	7/4 (exact)	2.06

Table: Results obtained with the SVM.

T_c	ν	χ^2_r	γ/ν	χ^2_r
2.26922(33)	1.004(48)	0.36	1.7634(68)	0.46
2.26925(11)	1 (exact)	0.3	7/4 (exact)	0.66

Table: Results obtained with the standard analysis.

Conclusions

- T_c, ν and γ could be estimated by interpreting the decision function.
- The accuracy of these estimates is slightly worse than that obtained with the multihistogram method.
- A strong suggestion to the global symmetry of the model was given by comparing the behaviour of the Score and of the number of support vectors over the possible polynomial kernels.

Future directions & Improvements

- Study the bias introduced by considering only 200 configurations.
- Study what kind of order parameters can be identified with the above algorithm. Especially interesting are the transitions for which no local order parameter can be identified.
- First order transitions?
- Try other global symmetries (Potts Model, WIP).

Thank you for your attention

Given a particular real valued non-negative convex *loss function* $\mathbb{V}(\hat{y}, y)$, the *risk* associated to the model function f can be expressed as,

$$R[f] = \mathbb{E}[\mathbb{V}(f(x), y] = \int dP(x, y)\mathbb{V}(f(x), y) \ , \tag{3}$$

where $P(x, y)$ is the *underlying probability distribution* of x and y. The solution to the learning problem reads then

$$f^{\star} = \arg\min_{f \in \mathcal{F}} R(f) \ , \tag{4}$$

where $\mathcal{F}$ is a class of model functions among which we expect to find $f^{\star}$

We say that the underlying probability distribution $P(x, y)$ is symmetric with respect to some compact symmetry group G of elements g, if

$$P(g\,x, y) = P(x, y),\ \forall g \in G\,. \tag{5}$$

We define the projection of a function f on the group G as

$$\mathcal{P}_G(f) = \frac{1}{|G|}\sum_{g\in G} f(gx) \tag{6}$$

where $|G|$ is the order of the group, and $g \in G$.
The operator $\mathcal{P}_G$, called Reynolds operators in the mathematics litterature, is a projector. Thus, it partitions $\mathcal{F}$ in a unique way,

$$\mathcal{H} = \mathcal{H}_G \oplus \mathcal{H}_\perp \tag{7}$$

where $\mathcal{H}_G$ is invariant. It follows that $\mathcal{H}_\perp \sim Ker(P_G)$. For any function in $\mathcal{H}$,

$$f(x) = f_I^G(x) + f_\perp(x) = P_G(f) + f_\perp(x) \tag{8}$$

with $P_G(f_\perp(x)) = 0$.

First, decompose X in the orbits of the group G,

$$\int_{\mathcal{X}} dP(x, y) = \sum_{g \in G} \int_{\mathcal{X}/G} dP(gx, y) \tag{9}$$

where $\mathcal{X}/G$ is the X space of orbits of G. Thus

$$R(f) = \sum_{g \in G} \int_{\mathcal{X}/G} dP(gx, y) \mathbb{V}(f(gx), y) \ . \tag{10}$$

Now if $dP(gx, y) = dP(x, y)$, by letting the sum over the elements of G filter to the right, we obtain

$$R(f) = \int_{\mathcal{X}/G} dP(x, y) \sum_{g \in G} \mathbb{V}(f(gx), y) = \frac{1}{|G|} \int_{\mathcal{X}} dP(x, y) \sum_{g \in G} \mathbb{V}(f(gx), y) \ . \tag{11}$$

Since L is convex,

$$\frac{1}{|G|} \sum_{g \in G} \mathbb{V}(f(gx), y) \geq \mathbb{V}\left(\frac{1}{|G|} \sum_{g \in G} f(gx), y \right) = \mathbb{V}\left(P_G[f], y\right) \ , \tag{12}$$

Thus, being $\mathbb{V}$ non-negative, the integrals of the quantities in the inequality lie in the same order, and we obtain,

$$R(f^G(x)) \leq R(f) \ . \tag{13}$$

Thus, starting from $\mathcal{F}$, the expected risk is lower if we pick f from the invariant subspace $\mathcal{F}_G$