

Université de Montréal

On the VC-dimension of Tensor Networks

par

Behnoush Khavari

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Computer Science

5 january 2022

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

On the VC-dimension of Tensor Networks

présenté par

Behnoush Khavari

a été évalué par un jury composé des personnes suivantes :

Prof. Simon Lacoste-Julien

(président-rapporteur)

Prof. Guillaume Rabusseau

(directeur de recherche)

Prof. Gauthier Gidel

(membre du jury)

Résumé

Les méthodes de réseau de tenseurs (TN) ont été un ingrédient essentiel des progrès de la physique de la matière condensée et ont récemment suscité l'intérêt de la communauté de l'apprentissage automatique pour leur capacité à représenter de manière compacte des objets de très grande dimension. Les méthodes TN peuvent par exemple être utilisées pour apprendre efficacement des modèles linéaires dans des espaces de caractéristiques exponentiellement grands [1]. Dans ce manuscrit, nous dérivons des limites supérieures et inférieures sur la VC-dimension et la pseudo-dimension d'une grande classe de Modèles TN pour la classification, la régression et la complétion. Nos bornes supérieures sont valables pour les modèles linéaires paramétrés par structures TN arbitraires, et nous dérivons des limites inférieures pour les modèles de décomposition tensorielle courants (CP, Tensor Train, Tensor Ring et Tucker) montrant l'étroitesse de notre borne supérieure générale. Ces résultats sont utilisés pour dériver une borne de généralisation qui peut être appliquée à la classification avec des matrices de faible rang ainsi qu'à des classificateurs linéaires basés sur l'un des modèles de décomposition tensorielle couramment utilisés. En corollaire de nos résultats, nous obtenons une borne sur la VC-dimension du classificateur basé sur le *matrix product state* introduit dans [1] en fonction de la dimension de liaison (i.e. rang de train tensoriel), qui répond à un problème ouvert répertorié par Cirac, Garre-Rubio et Pérez-García [2].

Mots clés: réseau de tenseur, décomposition de tenseur, VC-dimension, apprentissage supervisé

Abstract

Tensor network (TN) methods have been a key ingredient of advances in condensed matter physics and have recently sparked interest in the machine learning community for their ability to compactly represent very high-dimensional objects. TN methods can for example be used to efficiently learn linear models in exponentially large feature spaces [1]. In this manuscript, we derive upper and lower bounds on the VC-dimension and pseudo-dimension of a large class of TN models for classification, regression and completion. Our upper bounds hold for linear models parameterized by arbitrary TN structures, and we derive lower bounds for common tensor decomposition models (CP, Tensor Train, Tensor Ring and Tucker) showing the tightness of our general upper bound. These results are used to derive a generalization bound which can be applied to classification with low-rank matrices as well as linear classifiers based on any of the commonly used tensor decomposition models. As a corollary of our results, we obtain a bound on the VC-dimension of the matrix product state classifier introduced in [1] as a function of the so-called bond dimension (i.e. tensor train rank), which answers an open problem listed by Cirac, Garre-Rubio and Pérez-García [2].

Key words: Tensor network, Tensor decomposition, VC-dimension, Supervised learning

Contents

Résumé	5
Abstract	7
List of Tables	13
List of Figures	15
Liste des sigles et des abréviations	17
Remerciements	19
Introduction	21
Notations	27
Chapter 1. Tensors and Tensor Networks	29
1.1. Introduction	29
1.2. Notation	29
1.3. Tensors and Tensor Networks	30
1.3.1. Fundamental operations on tensors	30
1.4. Tensor network decompositions and tensor rank	32
1.4.1. Candecomp/Parafac (CP)	33
1.4.2. Tucker	34
1.4.3. Tensor Train (TT)	37
1.4.4. Other decompositions: Hierarchical Tucker and Tensor Ring	40
1.5. Classification with Tensor Train Weight	41
1.6. Equivalence of TNs with Convolutional Arithmetic Circuits	42
1.6.1. CP Model as a Shallow CAC/CNN	43
1.6.2. Hierarchical Tucker Decomposition as a Deep CAC	44
Chapter 2. Generalization Bound and Complexity Measures	47

2.1.	Introduction	47
2.2.	Classical Generalization Bounds for Classification	47
2.2.1.	Finite Class of Hypotheses	48
2.2.2.	Infinite Class of Hypotheses	50
2.3.	Generalization Bounds for Regression	54
2.3.1.	Finite Class of Hypotheses	54
2.3.2.	Infinite Class of Hypotheses	54
Chapter 3. Generalization Bound and VC-dimension of Tensor Networks .		57
3.1.	Introduction	57
3.1.1.	Tensor Network structures	57
3.2.	Tensor Network Learning Models	59
3.2.1.	Examples	60
3.3.	Bounds on the VC/Pseudo-dimension and the Generalization Gap	61
3.3.1.	Special cases	64
3.3.2.	Experiments	65
3.4.	Lower Bounds	66
3.4.1.	Proof of Theorem 11	67
3.4.2.	Rank-One Tensors	68
3.4.3.	Tensor Train and Tensor Ring	68
3.4.4.	Tucker	72
3.4.5.	CP	72
Chapter 4. Conclusion and Future Directions		77
Bibliography		79
Appendix A. Useful Formulas		85
A.1.	Essential Inequalities	85
A.1.1.	Markov's inequality	85
A.1.2.	Chebyshev's inequality	86
A.1.3.	Hoeffding's inequality	86
A.1.4.	Popoviciu's inequality [3]	89
Appendix B. Proofs for Chapter 2		91

B.1.	Proof of Lemma 2.2.1	91
B.2.	Proof of Corollary 2.2.2.....	93
B.3.	Proof of Lemma 2.2.4	94
Appendix C. VC-dimension of Half-Spaces		97
Appendix D. Lower Bounds on the Number of Sign Patterns.....		99
D.1.	Main Results.....	99
D.1.1.	Lower-bound on the Number of Sign Patterns of Low-rank Matrices [4]...	99
D.1.2.	Lower-bound on the Number of Sign Patterns of Tensor Trains	100
D.2.	Proofs for Section D.1.....	102
D.2.1.	Proof of General Position for the Ranks $r = 2$	102
D.2.2.	Proof of General Position based on Moment Curve for Tensor Train	103
D.2.3.	Dichotomy Counting [5].....	104

List of Tables

1.1	Some properties of CP, Tucker and TT compared against each other	41
3.1	Summary of our results for common TN structures. Both lower and upper bounds hold for the VC/pseudo-dimension of $\mathcal{H}_G^{\text{classif}}$, $\mathcal{H}_G^{\text{completion}}$ and $\mathcal{H}_G^{\text{regression}}$ for the corresponding TN structure G (see Equations (3.2.1-3.2.3)). The upper bounds follow from applying our general upper bound (Theorem 8) to each TN structure. The lower bounds are proved for each TN structure specifically. Each lower bound is followed by the condition under which it holds in parenthesis (small font). Note that the two bounds for TT and TR hold for both TN structures.....	66

List of Figures

1.1	Tensor network representation of common operations on matrices and vectors...	30
1.2	TN representation of the outer product of three vectors.....	31
1.3	Mode- n product of a third order tensor with three matrices	32
1.4	Inner product of tensors \mathcal{T} and \mathcal{S}	32
1.5	CP decomposition of a 4-th order tensor	34
1.6	Tucker decomposition of a 4-th order tensor	34
1.7	Figure from [6]. TT-SVD algorithm applied to a 4-th order tensor	39
1.8	Illustration of some common tensor networks.....	40
1.9	Figure from [1]. Each pixel value of a grayscale image is mapped to a normalized two-component vector.....	42
1.10	Decomposition of the weight tensor as a tensor train	42
1.11	Figure from [7]. CAC corresponding to the CP decomposition of the weight tensor of a linear model.....	44
1.12	Illustration of the Equation (1.6.5), i.e., the first recursion step for the construction of the Hierarchical Tucker tensor. Note that r_0 shows the dimension of the hyper-edge along all its modes.	45
1.13	Illustration of the Hierarchical Tucker representation of a tensor of order eight, defined by Equation (1.6.6).....	45
1.14	Figure from [7]. Deep CAC corresponding to the HT decomposition of the weight tensor.....	46
2.1	From [8]. (a) Illustration of shattering of three points in 2-dim by lines. (b) Two categories of four points in general position in two dimensions. The two sign patterns illustrated here are the ones that are not linearly separable. On the left, the four points lie on a convex hull. On the right one point lies inside the convex hull of the other three points.	53

2.2	Pseudo-shattering of two points in one dimension, with thresholds t_1 and t_2 witnessing the pseudo-shattering.	56
3.1	Disentangling the graph structure of a tensor network from its parameters.	58
3.2	Graph structures of TN representation of common decomposition models for 4th order and 9th order tensors. For CP, the black dot represents a hyperedge corresponding to a joint contraction over 4 indices. For the ease of representation, the edge weights, i.e., the dimensions of the core tensors are not shown.	59
3.3	Dashed lines represent the theoretical bound, full lines represent the generalization gap (averaged over 20 runs for both experiments), and shaded areas show the standard deviation. (left) Generalization error for two models with ranks $r = 2$ and $r = 4$ as a function of training size. (right) Generalization error for two sample sizes $n = 2000$ and $n = 4000$ as a function of the rank of the learned hypothesis.	66
3.4	Visualization of the proof of the lower bound on the VC-dimension of a tensor train tensor	71
D.1	Figure from [9]	100
D.2	(1) A 4-th order tensor train \mathcal{G} with the core highlighted in red being free to take any arbitrary values. (2) Mode- n Matricization of the same tensor \mathcal{G} w.r.t. the mode corresponding to the highlighted core.	101
D.3	Part (1) The broken TT of Figure D.2. Part (2) illustrates the matricization of the broken TT.	101

Liste des sigles et des abréviations

CAC	Convolutional Arithmetic Circuit
CNN	Convolutional Neural Network
CP	Candecomp/Parafac
ERM	Empirical Risk Minimization
FC	Fully Connected
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HOSVD	Higher Order SVD
HT	Hierarchical Tucker
ML	Machine Learning
MPS	Matrix Product State

ReLU	Rectified Linear Unit
SVD	Singular Value Decomposition
TN	Tensor Network
TR	Tensor Ring
TT	Tensor Train

Remerciements

It is my pleasure to thank the many people who made this thesis possible. First and foremost, I would like to express my deep gratitude to my supervisor, Professor Guillaume Rabusseau, for providing me with this masters opportunity at the university of Montreal and MILA. I feel so privileged for having found this opportunity and I will be ever grateful for. I really enjoyed every course that I took during this period at UdeM/MILA and I would like to thank my professors for those great classes. But again, all of that became possible thanks to my supervisor. Thank you Guillaume! It was an absolute pleasure learning about tensor networks in your excellent class on *tensor factorization techniques*. Also, I really enjoyed working under your supervision on this topic and I greatly appreciate your continuous support, precious advice and encouragements throughout my course of study and research work.

I would like to thank all the group members, past and present, especially Dr. Jacob Miller for the very helpful discussions on tensor networks. Also, I really enjoyed our great group meetings that helped me a lot to learn about many interesting aspects of tensor networks as well as other subjects in machine learning. A big 'Thank you!' to all of you Guillaume, Jacob, Ali, Andy, Beheshteh, Kaiwen, Farzaneh, Marawan, Maude, Meraj, Michael, Omar and Tianyu!

I cannot thank my family enough for their endless emotional support that always gave me courage and confidence to keep going!

Finally, I owe many thanks to my lovely housemates, Karen&André, who made my residence in Montreal such a wonderful experience. Merci infiniment!

Introduction

We know vectors and matrices as 1-dimensional and 2-dimensional arrays respectively. Tensors are the generalization of vectors and matrices to higher-order arrays. Therefore, vectors are first-order tensors and matrices are second-order tensors. One important place in machine learning (ML) where vectors and matrices are extensively used is in dealing with data with grid-like structure. Structured data can have different number of dimensions. A classical example of 1-dimensional data is the data with temporal dependence or time series, e.g., audio data, which is naturally represented as a vector whose entries are the frequencies recorded at each time step. Grey-scale images on the other hand, are well-known examples of 2-dimensional objects that are represented by matrices, where the entries of the matrix record the intensity of the color at the corresponding pixel of the image.

If we now consider RGB images, we need three such matrices to record the intensities over the three red, green and blue channels and in this case, a convenient way to keep all this information is to use a structured data type like a third-order tensor. By continuing this discussion to more and more complex data, like videos, which add time as the fourth dimension to the story, we observe how higher and higher-order tensors can be seen as natural candidates to represent specific types of data. This being said, which data type to take to represent the data points is a choice that among other factors depends on the learning model as well. While a neural network with fully-connected neurons takes the vectorization of images as input, in order to implement a convolutional neural network model on image data the matrix structure of images has to be kept.

Now, as we consider tensors of higher order, they become more expensive to deal with. That is because the number of entries grows exponentially with the order of the tensor; a vector of dimension d has d entries, a $d \times d$ matrix has d^2 entries, and so on, a p -th order tensor with dimension d along all its modes has d^p parameters. This fact makes it costly to work with high-order tensors. Tensor decomposition techniques get around this problem by decomposing a high-order tensor into lower-order components that compared to the initial tensor, have considerably less parameters. This idea is the generalization of the well-known low-rank decomposition of matrices in linear algebra, to the realm of multi-linear algebra. There exist many types of tensor decompositions, such as CP decomposition [10], Tucker

decomposition [11] and tensor train (TT) decomposition [12], to name a few. As the order of tensors increases, the ordinary notation of multi-linear algebra results in very lengthy expressions to represent operations on tensors even for the simplest tensor decompositions, let alone if for any reason, some more complicated decomposition is required. This negatively affects the tractability of tensor operations, and hence the introduction of tensor networks. Tensor networks (TNs) have emerged in the quantum physics community as a mean to compactly represent wave functions of large quantum systems [13, 14, 15] which are tensors of potentially very high orders. Their introduction in physics can be traced back to the work of Penrose [16] and Feynman [17].

As a generalization of specific tensor decompositions, TN methods rely on factorizing a high-order tensor into small factors and have recently gained interest from the machine learning community for their ability to efficiently represent and perform operations on very high-dimensional data and high-order tensors.

Yet, the practicality of tensors is not restricted to data representation. A prevailing application of tensors is in parameterizing large machine learning models more efficiently. One of the first steps in this direction was done in [18], where tensor decomposition techniques are used to compress fully-connected layers in neural networks by first tensorizing the corresponding dense weight matrices. An important concern in working with modern deep networks is the huge number of their parameters which to some extent is due to the fully-connected (FC) layers. This number can reach orders of magnitude of millions due to both the input representation and output layers having large dimensions. The authors of [18] propose to reshape the FC layer into a high-order tensor before applying a low-rank tensor approximation to it and report a compression factor around 7 on the VGG network. This reshaping of low-order arrays into high-order tensors has proved better performance in case of matrix completion [19] as well and has become a common practice in completion tasks [19, 20, 21].

Apart from their successful application in compressing large neural networks [18, 22, 23, 24, 25], high-order tensors have been used in designing novel approaches to supervised [1, 26, 27] and unsupervised [28, 29, 30] learning. Most of these methods leverage the fact that TN can be used to efficiently parameterize high-dimensional linear maps, which is appealing from two perspectives: it makes it possible to learn models in exponentially large feature spaces *and* it acts as a regularizer, controlling the capacity of the class of hypotheses considered for learning. As another application, [31] takes advantage of tensor decomposition techniques to estimate the parameters of common latent variable models, such as Gaussian mixture model (GMM) and Hidden Markov model (HMM), in the framework of the method of moments. Their key observation is the natural representation of the n -th order moment by a n -th order tensor and using tensor decomposition methods to solve the corresponding equations.

Besides these applications, on the *theory* side, tensor networks served to the development of new insights on the expressiveness of deep neural networks [7, 32, 33, 34]. Especially, in [7] the statement *deeper models are exponentially more expressive than shallower models*, or *depth efficiency* in short, was examined theoretically for convolutional arithmetic circuits (CAC).

Regarding the studies on the theoretical foundations of tensor networks, while the expressive power of TN models has been studied recently [35, 36], the focus has mainly been on the representation capacity of TN models, but not on their ability to *generalize* in the context of supervised learning tasks. In this work, we study the generalization ability of TN models by deriving lower and upper bounds on the VC-dimension and pseudo-dimension of TN models commonly used for classification, completion and regression, from which bounds on the generalization gap of TN models can be derived. Using the general framework of tensor networks, we derive an upper bound for models parameterized by *arbitrary* TN structures, which applies to all commonly used tensor decomposition models [37] such as CP [10], Tucker [38] and tensor train (TT) [12], as well as more sophisticated structures including hierarchical Tucker [39, 40], tensor ring (TR) [41] and projected entangled state pairs (PEPS) [42].

The goal of supervised learning is to learn a function f mapping inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$ from a sample of input-output examples $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ drawn from an unknown distribution D where each $y_i \simeq f(x_i)$. Given a space of hypothesis $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, one natural objective is to find the hypothesis $h \in \mathcal{H}$ minimizing the *risk* $R(h) = \mathbb{E}_{(x,y) \sim D} \ell(h(x), y)$ where $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ is a loss function measuring the quality of the predictions made by h . However, since the distribution D is unknown, machine learning algorithms often rely on the *empirical risk minimization* principle which consists in finding the hypothesis $h \in \mathcal{H}$ that minimizes the *empirical risk* $\hat{R}_S(h) = \frac{1}{n} \sum_{i=1}^n \ell(h(x_i), y_i)$. It is easy to see that the empirical risk is an unbiased estimator of the risk (though, notice that this is only true if we have not used the sample S to learn a minimizer function $h_{\min} \in \mathcal{H}$, which is by default assumed) and one of the focus of learning theory is to provide guarantees on the quality of this estimator. Such guarantees include *generalization bounds*, which are probabilistic bounds on the *generalization gap* $R(h) - \hat{R}_S(h)$. The generalization gap naturally depends on the size of the sample S , but also on the richness (or *capacity*, complexity) of the hypothesis class \mathcal{H} . There exist several ways to measure the complexity of \mathcal{H} including VC-dimension, Rademacher complexity, covering numbers and packing numbers [43, 8, 44, 45].

In this work, we focus on two combinatorial measures of complexity, VC-dimension for *classification*, and pseudo-dimension for *regression* and *completion*. For the classification task, we consider the class of linear models $h : \mathcal{X} \mapsto \text{sign}(\langle \mathcal{X}, \mathcal{W} \rangle)$ taking p -th order tensors $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_p}$ as input and whose weight tensor \mathcal{W} is compactly represented using some

tensor network. Our analysis proceeds mainly in two steps. First, we formally define the notion of TN learning model by disentangling the underlying graph structure of a TN from its parameters (the core tensors, or factors, involved in the decomposition). This allows us to define, in a conceptually simple way, the hypothesis class \mathcal{H}_G corresponding to the family of linear models whose weights are represented using an arbitrary TN structure G . We then proceed to deriving upper bounds on the VC-dimension and generalization error of the class \mathcal{H}_G . For the regression and completion tasks a quite similar approach results in the same bound on the pseudo-dimension. These bounds follow from a classical result from Warren [46] which was previously used to obtain generalization bounds for neural networks [47], matrix completion [4], tensor completion [48] as well as probability classes based on quantum circuits [49]. The bounds we derive naturally relate the capacity of \mathcal{H}_G to the underlying graph structure G through the number of nodes and effective number of parameters of the TN. To assess the tightness of our general upper bound, we derive lower bounds for particular TN structures (rank-one, CP, Tucker, TT and TR). These lower bounds show that, for completion, regression and classification, our general upper bound is tight up to a log factor for rank-one, TT and TR tensors, and is tight up to a constant for matrices. This implies that our upper bound for tensor networks in general is tight; but better upper bounds for specific tensor network structures could be derived in the future. Lastly, as a corollary of our results, we obtain a bound on the VC-dimension of the tensor train classifier introduced in [1], which answers one of the open problems listed by Cirac, Garre-Rubio and Pérez-García in [2].

Related work Machine learning models using low-rank parametrization of the weights have been investigated (mainly from a practical perspective) for various decomposition models, including low-rank matrices [50, 51, 52], CP [53, 54, 55], Tucker [56, 57, 58, 59], tensor train [19, 60, 26, 1, 27, 61, 62, 63, 64] and PEPS [65]. From a more theoretical perspective, generalization bounds for matrix and tensor completion have been derived in [4, 48] (based on the Tucker format for the tensor case). A bound on the VC-dimension of low-rank matrix classifiers was derived in [52] and a bound on the pseudo-dimension of regression functions whose weights have low Tucker rank was given in [59] (for both these cases, we show that our results improve over these previous bounds, see Section 3.3.1). To the best of our knowledge the VC-dimension of tensor train classifiers has not been studied in the past, but the statistical consistency of the convex relaxation of the tensor completion problem was studied in [66, 67] for the Tucker decomposition and in [68] for the tensor train decomposition. In [69] the authors study the complexity of learning with tree tensor networks using the notion of metric entropy and covering numbers. They provide generalization bounds which are qualitatively similar to ours, but their results only hold for TN structures whose underlying graph is a tree (thus excluding models such as

CP, tensor ring and PEPS) and they do not provide lower bounds. Lastly, in [49], the expressive power of a class of quantum circuits is studied using the pseudo-dimension. The pseudo-dimension is bounded in terms of the dimension of the qudits, the depth of the quantum circuit and the number of unitaries in the circuit. The setup of the problem as well as the techniques used to upper-bound the pseudo-dimension are very similar to our study. In their work, they consider two different setups. The first one has a fixed circuit structure, resulting in their upper-bound in Theorem 3.3, which is similar to the general upper-bound that we provide in Theorem 8. In the second setup, they consider variable circuit structure with fixed depth and fixed total number of unitaries, but otherwise arbitrary architecture. In this case the upper-bound takes a depth dependence as stated in Theorem 3.7. For this part of their work, we did not do a similar study on TNs. Regarding the tightness of their bounds, they do not provide lower-bounds on the pseudo-dimension.

Summary of contributions We introduce a *unifying framework for TN-based learning models*, which generalizes a wide range of models based on tensor factorization for completion, classification and regression. This framework allows us to consider the class \mathcal{H}_G of low-rank TN models for a given *arbitrary TN structure* G (Section 3.1.1). We provide general *upper bounds on the pseudo-dimension and VC-dimension* of the hypothesis class \mathcal{H}_G for *arbitrary TN structure* G for regression, classification and completion. Our results naturally relate the capacity of \mathcal{H}_G to the number of parameters of the underlying TN structure G (Section 3.3). From these results, we derive a *generalization bound for TN-based classifiers parameterized by arbitrary TN structures* (Theorem 10). We compare our results to previous bounds for specific decomposition models and show that our general upper bound is always of the same order and sometimes even improves on previous bounds (Section 3.3.1). We derive several lower bounds showing that our general upper bound is tight up to a log factor for particular TN structures (Section 3.4). A summary of the lower bounds derived in this work, as well as upper bounds implied by our general result for particular TN structures, can be found in Table 3.1.

This thesis is based on a paper published at NeurIPS 2021 by the author and their supervisor [70]. The content of the paper is mainly included in Chapter 3 of this manuscript.

The outline of the thesis is as follows. We start Chapter 1 by introducing tensors as the generalization of vectors and matrices to arrays of higher order and we continue by defining the main tensor operations. Then, we present tensor networks as a convenient graphical notation for dealing with tensor decompositions of high-order tensors. Following that, we introduce different notions of rank for high-order tensors and study several common tensor networks in more details. Also, we review some supervised learning models in the literature which are based on those tensor network structures [1, 7]. Then, we see the equivalence of some tensor network models with specific sum-product neural networks and briefly review

how this equivalence can be used to show the depth efficiency in neural networks [7].

Chapter 2 goes over some basic concepts in supervised learning theory; we study the generalization bound for binary classification models with both finite and infinite hypothesis classes. Closely related to that, we define the VC-dimension of binary-valued function classes and upper-bound the generalization gap in terms of the VC-dimension. We end this chapter by explaining how a similar notion of complexity, called the pseudo-dimension, is defined for the class of real-valued functions and serves to quantify the complexity of hypothesis classes for regression and completion tasks.

Chapter 3 mainly contains the content of our paper [70]. We include our two main theorems on the upper and lower bounds on the VC/pseudo-dimensions of tensor network models. To make it easier to follow the details of our analysis, here we have expanded some parts of the paper. Furthermore, while in the paper the calculation of our lower bounds are put in the appendices, in the thesis we have included all those calculations in the main body of the fourth chapter. We have also added some examples of those proofs for tensor networks of relatively low order. Chapter 4 gives a short summary of our results along with commenting on several possible future directions.

Finally, we devote Appendices A to D to lengthier calculations or proofs of theorems and lemmas.

Notations

$[k]$	set of integers from 1 to k
$[h, k]$	set of integers from h to k
$ S , a $	cardinality of the set S , absolute value of the scalar a
δ_{ij}	Kronecker symbol: equals 1 if $i = j$ and 0 otherwise
$a, \mathbf{v}, \mathbf{M}, \mathcal{T}$	scalar, vector, matrix, tensor
I_n	$n \times n$ identity matrix
$\text{Tr}(\mathbf{M})$	trace of the matrix \mathbf{M}
$\langle \mathcal{T}, \mathcal{S} \rangle$	inner product between vectors, matrices or tensors
$\ \cdot\ _F$	Frobenius norm
$V^{\otimes k}$	k -th order tensor (k -fold tensor) product of the vector space V
$\mathbf{M}_{i,:}, \mathbf{M}_{:,j}, \mathcal{T}_{k,:,:}$	i -th row of \mathbf{M} , j -th column of \mathbf{M} , k -th mode-1 slice of \mathcal{T}
$\mathbf{T}_{(k)}$ or $\mathcal{T}_{(k)}$	mode- k matricization of the tensor \mathcal{T}
$\mathbf{v} \circ \mathbf{u}$	outer product between vectors, matrices or tensors
$\mathbf{v} \otimes \mathbf{u}$	Kronecker product (for vectors, matrices and higher-order tensors)
$\mathcal{T} \times_k \mathbf{M}$	mode- k matrix product
$\mathbb{P}[\cdot]$	probability of an event
$\mathbb{E}[\cdot]$	expectation of a random variable
$\text{sign}(\cdot)$	Sign function
$\mathcal{Y}^{\mathcal{X}}$	the space of functions $f : \mathcal{X} \mapsto \mathcal{Y}$

Chapter 1

Tensors and Tensor Networks

1.1. Introduction

This whole chapter is a brief review of some basic concepts in the tensor network literature which are tightly related to the subject of our study. The goal of this chapter is to give a comprehensive view of tensors and tensor operations as well as some relevant tensor decompositions in the tensor network representation. We review two well-known algorithms used to build some specific tensor decompositions. We also give some examples of tensor network models used in supervised learning tasks like *classification* and show how some of them are equivalent to specific neural network models.

1.2. Notation

In this section, we present basic notions of tensor algebra and tensor networks. We start by introducing some notations. For any integer k we use $[k]$ to denote the set of integers from 1 to k . For a set S , the notation $|S|$ represents the cardinality of the set. We use lower case bold letters for vectors (e.g. $\mathbf{v} \in \mathbb{R}^{d_1}$), upper case bold letters for matrices (e.g. $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$) and bold calligraphic letters for higher order tensors (e.g. $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$). The inner product of two k -th order tensors $\mathcal{S}, \mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_k}$ is defined by $\langle \mathcal{T}, \mathcal{S} \rangle = \sum_{i_1=1}^{d_1} \dots \sum_{i_k=1}^{d_k} \mathcal{T}_{i_1 \dots i_k} \mathcal{S}_{i_1 \dots i_k}$. The outer product of two vectors $\mathbf{u} \in \mathbb{R}^{d_1}$ and $\mathbf{v} \in \mathbb{R}^{d_2}$ is denoted by $\mathbf{u} \circ \mathbf{v} \in \mathbb{R}^{d_1 \times d_2}$ with elements $(\mathbf{u} \circ \mathbf{v})_{i,j} = \mathbf{u}_i \mathbf{v}_j$. The outer product generalizes to an arbitrary number of vectors. We use the notation $(\mathbb{R}^d)^{\otimes p}$ to denote the space of p -th order hypercubic tensors of size $\underbrace{d \times d \times \dots \times d}_{p \text{ times}}$. We denote by $\mathcal{Y}^{\mathcal{X}}$ the space of functions $f : \mathcal{X} \mapsto \mathcal{Y}$. $\text{sign}(\cdot)$ stands for the sign function. Finally, given a graph $G = (V, E)$ and a vertex $v \in V$, we denote by $E_v = \{e \in E \mid v \in e\}$ the set of edges incident to the vertex v .

$$\text{---} \textcircled{\mathbf{A}} \text{---} \textcircled{\mathbf{B}} \text{---} = \mathbf{AB} \quad \textcircled{\mathbf{A}} \text{---} \textcircled{\mathbf{A}} = \text{Tr}(\mathbf{A}) \quad \textcircled{\mathbf{x}} \text{---} \textcircled{\mathbf{M}} \text{---} \textcircled{\mathbf{y}} = \mathbf{x}^\top \mathbf{M} \mathbf{y}$$

Figure 1.1. Tensor network representation of common operations on matrices and vectors.

1.3. Tensors and Tensor Networks

A *tensor* $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_p}$ can simply be seen as a multidimensional array of scalars ($\mathcal{T}_{i_1, \dots, i_p} : i_n \in [d_n], n \in [p]$). Tensors can be seen as the generalization of vectors and matrices to arrays of higher order. As the order increases the representation of array becomes more difficult. Tensor networks provide a simple way of representing and dealing with these high-order objects and substantially simplify the analysis of tensor operations in several ways.

Complex operations on tensors can be intuitively represented using the graphical notation of tensor network (TN) diagrams [14, 13]. In tensor networks, a p -th order tensor is illustrated as a node with p edges (or *legs*) in a graph $\begin{array}{c} d_1 \\ \textcircled{\mathbf{T}} \\ d_2 \dots \end{array}$. That is, a vector \mathbf{v} of dimension d is

simply shown as a one-node graph with one edge as $\begin{array}{c} d \\ \textcircled{\mathbf{v}} \end{array}$ and a $m \times n$ matrix is represented as $\begin{array}{c} m \\ \textcircled{\mathbf{M}} \\ n \end{array}$. An edge between two nodes of a TN represents a contraction over the corresponding modes of the two tensors. Consider the following simple TN with two nodes: $\begin{array}{c} m \\ \textcircled{\mathbf{A}} \\ n \end{array} \text{---} \begin{array}{c} \textcircled{\mathbf{x}} \end{array}$. The first node represents a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and the second one a vector $\mathbf{x} \in \mathbb{R}^n$. Since this TN has one dangling leg (i.e. an edge which is not connected to any other node), it represents a first order tensor, i.e. a vector. The edge between the second leg of \mathbf{A} and the leg of \mathbf{x} corresponds to a contraction between the second mode of \mathbf{A} and the first mode of \mathbf{x} . Hence, the resulting TN represents the classical matrix-product between a tensor and a vector, which can be seen by calculating the i -th component of this TN: $i \text{---} \textcircled{\mathbf{A}} \text{---} \textcircled{\mathbf{x}} = \sum_j \mathbf{A}_{ij} \mathbf{x}_j = (\mathbf{A} \mathbf{x})_i$. Examples of TN representations of some other common operations on matrices and vectors can be found in Figure 1.1.

In this thesis, we mainly deal with the factorization of tensors into lower-order tensors, including matrices and vectors. The combination of these constitutional components into the high-dimensional tensor is through the notion of tensor product. There are different types of such products and here we mention a couple of them that we will see later again.

1.3.1. Fundamental operations on tensors

Matricization and vectorization We first introduce modes of a tensor. Each dimension or *way* of a p -th order tensor is called a mode, i.e., a tensor of order p has p modes [11]. Corresponding to each mode, we can extract *fibers* of a tensor; mode- i fibers of a tensor for $i = 1, \dots, p$, are obtained by fixing all indices of the tensor but the i -th

one. E.g., for a third-order tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, we have $d_2 d_3$ mode-1 fibers, which are all vectors $\mathcal{T}_{:,i_2,i_3} \in \mathbb{R}^{d_1}$ for $i_2 \in [d_2]$ and $i_3 \in [d_3]$. Here the colon notation for the first index of \mathcal{T} means that we go over all possible values of the first index.

Matricization of a p -th order tensor is rearranging its entries into a matrix. There exist many ways to matricize or *flatten* a tensor, here we only consider the *mode- n* matricization that means reordering a tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_n \times \dots \times d_p}$ as a matrix $\mathbf{T}_{(n)} \in \mathbb{R}^{d_n \times d_1 \dots d_{n-1} d_{n+1} \dots d_p}$, or more compactly $\mathcal{T}_{(n)} \in \mathbb{R}^{d_n \times \prod_{i \neq n} d_i}$. Stating it in our new terminology, the mode- n matricization of \mathcal{T} has the mode- n fibers of \mathcal{T} as columns.

Tensor contraction Contraction is an operation between two tensors of arbitrary orders. In general, if two tensors have the same dimensions along some of their modes, one can contract these tensors along any of those modes. As an example, consider two tensors $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$ and $\mathcal{S} \in \mathbb{R}^{d_5 \times d_2 \times d_6 \times d_3}$. We can contract the two tensors into a new 4-th order tensor \mathcal{U} with the components $\mathcal{U}_{i,j,m,n} = \sum_{k=1}^{d_2} \sum_{l=1}^{d_3} \mathcal{T}_{i,k,l,j} \mathcal{S}_{m,k,n,l}$. Needless to say, the contraction does not need to be done along all modes that have the same dimensions in the two tensors; therefore, in this example, we can contract the two tensors along only one of their modes, which results in a tensor of order six, e.g., $\mathcal{V}_{i,l,j,m,n,h} = \sum_{k=1}^{d_2} \mathcal{T}_{i,k,l,j} \mathcal{S}_{m,k,n,h}$.

Outer product The outer product, as defined earlier, generalizes to an arbitrary number of vectors. The outer product of p vectors $\mathbf{v}_1 \in \mathbb{R}^{d_1}, \dots, \mathbf{v}_p \in \mathbb{R}^{d_p}$, denoted by $\mathbf{v}_1 \circ \dots \circ \mathbf{v}_p \in \mathbb{R}^{d_1 \times \dots \times d_p}$, is a p -th order tensor \mathcal{T} with elements $\mathcal{T}_{i_1, \dots, i_p} = (\mathbf{v}_1)_{i_1} \dots (\mathbf{v}_p)_{i_p}$, where $(\mathbf{v}_1)_{i_1}$ stands for the i_1 -th element of \mathbf{v}_1 . Finally, if a p -th order tensor \mathcal{T} is decomposable as $\mathcal{T} = \mathbf{v}_1 \circ \dots \circ \mathbf{v}_p$, we call it a rank-1 tensor (observe that not all tensors are rank-1 tensors). From our above definition, each element of a rank-1 tensor is the product of the corresponding vector elements. To see the representation of outer product in tensor network format, Figure 1.2 illustrates the third-order tensor $\mathcal{T} = \mathbf{u} \circ \mathbf{v} \circ \mathbf{w}$.

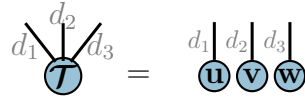


Figure 1.2. TN representation of the outer product of three vectors.

Mode- n product This product can be seen as a generalization of the matrix product to tensors of arbitrary orders. Mode- n product is defined between a p -th order tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_n \times \dots \times d_p}$ and a matrix $\mathbf{M} \in \mathbb{R}^{m \times d_n}$. The operation consists of contracting the n -th mode of the tensor with the second mode of the matrix which results in a new tensor with the dimensionality m instead of d_n at the n -th mode. More precisely: $(\mathcal{T} \times_n \mathbf{M})_{i_1, \dots, i_{n-1}, j, i_{n+1}, \dots, i_p} = \sum_{i_n=1}^{d_n} \mathcal{T}_{i_1, \dots, i_n, \dots, i_p} \mathbf{M}_{j, i_n} \in \mathbb{R}^{d_1, \dots, d_{n-1}, m, d_{n+1}, \dots, d_p}$. As a simple example, for a third-order tensor \mathcal{X} , we have $(\mathcal{X} \times_2 \mathbf{M})_{i_1 i_2 i_3} = \sum_j \mathcal{X}_{i_1 j i_3} \mathbf{M}_{i_2 j}$. Figure 1.3

illustrates the mode- n product of a third order tensor \mathcal{T} along its three modes with three matrices \mathbf{A} , \mathbf{B} and \mathbf{C} .

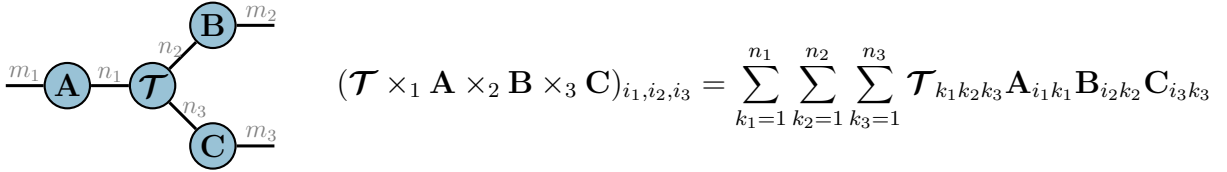


Figure 1.3. Mode- n product of a third order tensor with three matrices

Inner product The inner product is defined for two tensors of the same size as their contraction along all modes. The inner product of two k -th order tensors $\mathcal{S}, \mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_k}$ is defined by $\langle \mathcal{T}, \mathcal{S} \rangle = \sum_{i_1=1}^{d_1} \dots \sum_{i_k=1}^{d_k} \mathcal{T}_{i_1 \dots i_k} \mathcal{S}_{i_1 \dots i_k}$. In tensor network format, the inner product is represented by linking all corresponding edges of these tensors together. Figure 1.4 illustrates the inner product of two third-order tensors.

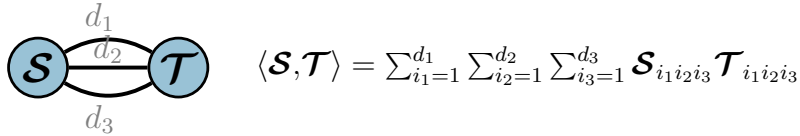


Figure 1.4. Inner product of tensors \mathcal{T} and \mathcal{S}

Frobenius norm The Frobenius norm of a tensor \mathcal{T} of order p is defined as the square root of the sum of its squared entries, i.e., $\|\mathcal{T}\|_F = \left(\sum_{i_1, \dots, i_p} \mathcal{T}_{i_1, \dots, i_p}^2 \right)^{\frac{1}{2}} = \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle}$.

Tensor rank The rank of a high-order tensor can be defined in several ways. It is a key concept in tensor studies and there exist several variants of it, each of which is associated with a special *tensor decomposition*. We will define some of these different types of rank as we proceed in this chapter. At this point, it suffices to introduce it as a generalization of the matrix rank. Recall that the rank of a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, denoted by r , is the smallest possible value for which there exist two matrices $\mathbf{A} \in \mathbb{R}^{m \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times n}$ whose matrix product gives \mathbf{M} , i.e., $\mathbf{M} = \mathbf{AB}$.

1.4. Tensor network decompositions and tensor rank

Manipulating high-order tensors is computationally expensive, because the number of tensor entries grows exponentially with the order. Tensor decompositions get around this problem by breaking down a big tensor into smaller tensor components of lower order and dimension which altogether have much less entries than the initial tensor. Among many possible tensor network decompositions of a given tensor, we only introduce some of the more common ones that we will study in the next sections and subsequent chapters. Also,

we will see that for each of these prevalent decompositions, an associated notion of rank is defined. The first decomposition to introduce here is the Candecomp/Parafac (CP) [10].

1.4.1. Candecomp/Parafac (CP)

For any tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_p}$ of order p , there exist an integer $r \geq 1$ and rp vectors $\{\{\mathbf{t}_i^{(k)} \in \mathbb{R}^{d_i}\}_{k=1}^r\}_{i=1}^p$, in terms of which \mathcal{T} is decomposed as

$$\mathcal{T} = \sum_{k=1}^r \lambda^{(k)} \mathbf{t}_1^{(k)} \circ \mathbf{t}_2^{(k)} \circ \dots \circ \mathbf{t}_p^{(k)} \quad (1.4.1)$$

It is easy to show that such a decomposition always exists; that is because there always exists a trivial decomposition, i.e., when $r = d_1 d_2 \dots d_p$ with the vectors $\mathbf{t}_i^{(k)}$ being all canonical bases of the real spaces \mathbb{R}^{d_i} and the scalars $\lambda^{(k)}$ being the entries of the tensor \mathcal{T} . Obviously, the interesting case is when $r < d_1 d_2 \dots d_p$ and the smaller r , the more efficient the CP decomposition. The *CP-rank*, or simply the rank, of a tensor \mathcal{T} is the smallest value r for which the CP representation (1.4.1) exists. To mention a major difference between tensors and matrices, note that while the singular value decomposition of a matrix and accordingly its rank can be calculated in polynomial time, it is a NP-hard problem to determine the rank of a tensor [71]. Nevertheless, there are some bounds for special cases, e.g., it is not difficult to show that for a third-order tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, the tensor rank is upper-bounded by

$$\text{rank}(\mathcal{T}) \leq \min\{d_2 d_3, d_1 d_3, d_1 d_2\} \quad (1.4.2)$$

Another difference between matrices and tensors in terms of the rank decomposition is the uniqueness. For a matrix of rank r , we can find infinitely many distinct decompositions; if a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ has rank r , it can be decomposed as $\mathbf{M} = \mathbf{A}\mathbf{B}$ with $\mathbf{A} \in \mathbb{R}^{m \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times n}$. However, the right-hand side would not change if we had instead $\mathbf{M} = \mathbf{A}\mathbf{W}^{-1}\mathbf{W}\mathbf{B}$ with $\mathbf{W} \in \mathbb{R}^{r \times r}$ being an invertible matrix. Now, by defining $\mathbf{S} = \mathbf{A}\mathbf{W}$ and $\mathbf{T} = \mathbf{W}^{-1}\mathbf{B}$, matrix \mathbf{M} can also be decomposed as $\mathbf{M} = \mathbf{S}\mathbf{T}$, which means that the rank decomposition of a matrix is not unique. This is not the case for the CP decomposition. The CP decomposition can be unique (up to trivial permutations and scaling) under relatively weak assumptions [11]. In [11] the sufficient and necessary conditions for the uniqueness of a given CP decomposition of a tensor of an arbitrary order and a given rank are explained. We do not go through the details of this subject, as it is out of the scope of our study.

To give the TN representation of the CP, we now introduce an equivalent representation of this decomposition. This representation works by arranging all r sets of vectors $\mathbf{t}_i^{(k)} \in \mathbb{R}^{d_i}$'s into matrices of size $d_i \times r$, which we denote by \mathbf{T}_k 's for $k \in [p]$. This alternative representation is based on mode- n product. In terms of these matrices, the elements of \mathcal{T} are written as

$$\mathcal{T}_{i_1, \dots, i_p} = \sum_{k_1, \dots, k_p} \delta_{k_1, \dots, k_p} (\mathbf{T}_1)_{i_1, k_1} (\mathbf{T}_2)_{i_2, k_2} \dots (\mathbf{T}_p)_{i_p, k_p}, \quad (1.4.3)$$

with δ_{k_1, \dots, k_p} being the Kronecker delta function in p dimensions, i.e., $\delta_{i_1, i_2, \dots, i_p} = 1$ if $i_1 = i_2 = \dots = i_p$, and 0 otherwise. Note that, in this representation, we have absorbed the scalar values $\lambda^{(k)}$ in (1.4.1) into the matrices \mathbf{T}_i 's. From the definition of mode- n product, this relation can be rewritten as

$$\mathcal{T} = \mathcal{I} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \cdots \times_p \mathbf{T}_p, \quad (1.4.4)$$

with \mathcal{I} being the super-identity, i.e., the p -th order tensor representation of the corresponding Kronecker delta. Having this formula, the TN representation of the CP decomposition of a 4-th order tensor is illustrated as below

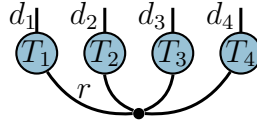


Figure 1.5. CP decomposition of a 4-th order tensor

The black dot in the diagram represents the super-identity tensor \mathcal{I} . Also, the alternative definition of the CP given in (1.4.4) shows that the CP decomposition can be seen as a special case of the Tucker representation, which we will see in the next subsection.

1.4.2. Tucker

As defined earlier, the *mode- n* product of a p -th order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_n \times \dots \times d_p}$ and a matrix $\mathbf{M} \in \mathbb{R}^{m \times d_n}$, denoted by $\mathcal{X} \times_n \mathbf{M}$, is of size $d_1 \times \dots \times d_{n-1} \times m \times d_{n+1} \times \dots \times d_p$. The Tucker decomposition of a p -th order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_p}$, is defined by

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{p-1} \mathbf{U}_{p-1} \times_p \mathbf{U}_p, \quad (1.4.5)$$

where $\mathcal{G} \in \mathbb{R}^{r_1 \times \dots \times r_p}$ and $\mathbf{U}_i \in \mathbb{R}^{d_i \times r_i}$. The tensor \mathcal{G} is called the core tensor and the matrices $\{\mathbf{U}_i\}_{i=1}^p$ are known as the factor matrices. The tuple $(r_i)_{i=1}^p$ containing the dimensions of the core tensor along all its modes is called the Tucker-rank. As an example, the TN representation of the Tucker decomposition of a 4-th order tensor \mathcal{X} is as shown in Figure 1.6.

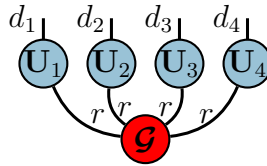


Figure 1.6. Tucker decomposition of a 4-th order tensor

By comparing this definition with Equation (1.4.4) we observe that the CP decomposition is a special case of the Tucker with the core tensor \mathcal{G} being the super-identity tensor. Furthermore, since for every tensor there always exists a CP decomposition, we conclude that

the Tucker decomposition always exists as well. More interestingly, it is not difficult to show that the factor matrices $\mathbf{U}_i \in \mathbb{R}^{d_i \times r_i}$ can always be set as unitary. This can be verified based on three facts; first, that any matrix \mathbf{M} has a SVD decomposition as $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, with \mathbf{U} and \mathbf{V} being unitary matrices, secondly, that as long as dimensionality-wise consistent, $\mathcal{T} \times_n \mathbf{U}_1 \times_n \mathbf{U}_2 = \mathcal{T} \times_n (\mathbf{U}_2 \mathbf{U}_1)$, and lastly, that the order of mode- n products for different modes does not matter, i.e., $\mathcal{T} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots = \mathcal{T} \times_2 \mathbf{U}_2 \times_1 \mathbf{U}_1 \dots$, and this generalizes to any number of mode- n products with any arbitrary ordering of them. Based on these three observations, we now show how the non-unitary parts of the factor matrices can be absorbed inside the core tensor \mathcal{G} and leads to a Tucker decomposition with unitary factors. Consider a tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ with a Tucker decomposition $\mathcal{T} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_p \mathbf{U}_p$. By replacing each factor matrix $\{\mathbf{U}_i\}_{i=1}^p$ with its SVD decomposition $\mathbf{U}_i = \mathbf{S}_i \mathbf{\Sigma}_i \mathbf{T}_i^\top$, we have $\mathcal{T} = \mathcal{G} \times_1 (\mathbf{S}_1 \mathbf{\Sigma}_1 \mathbf{T}_1^\top) \times_2 (\mathbf{S}_2 \mathbf{\Sigma}_2 \mathbf{T}_2^\top) \cdots \times_p (\mathbf{S}_p \mathbf{\Sigma}_p \mathbf{T}_p^\top)$. From the second observation made above, this is equal to

$$\mathcal{T} = \mathcal{G} \times_1 (\mathbf{\Sigma}_1 \mathbf{T}_1^\top) \times_1 \mathbf{S}_1 \times_2 (\mathbf{\Sigma}_2 \mathbf{T}_2^\top) \times_2 \mathbf{S}_2 \cdots \times_p (\mathbf{\Sigma}_p \mathbf{T}_p^\top) \times_p \mathbf{S}_p \quad (1.4.6)$$

Then, from the third fact mentioned above, the ordering of the mode- n products can be changed as follows

$$\mathcal{T} = \mathcal{G} \times_1 (\mathbf{\Sigma}_1 \mathbf{T}_1^\top) \times_2 (\mathbf{\Sigma}_2 \mathbf{T}_2^\top) \cdots \times_p (\mathbf{\Sigma}_p \mathbf{T}_p^\top) \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \cdots \times_p \mathbf{S}_p \quad (1.4.7)$$

Note that to go from Equation (1.4.6) to Equation (1.4.7), we only need to change the order of mode- n products with different modes and hence, it is possible to apply the third consideration above. Finally, we can define a new core tensor as $\mathcal{C} = \mathcal{G} \times_1 (\mathbf{\Sigma}_1 \mathbf{T}_1^\top) \times_2 (\mathbf{\Sigma}_2 \mathbf{T}_2^\top) \cdots \times_p (\mathbf{\Sigma}_p \mathbf{T}_p^\top)$ and rewrite Equation (1.4.6) as

$$\mathcal{T} = \mathcal{C} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \cdots \times_p \mathbf{S}_p \quad (1.4.8)$$

which is a Tucker decomposition for \mathcal{T} with unitary factor matrices $\mathbf{S}_1, \dots, \mathbf{S}_p$.

Henceforth, whenever we talk about the Tucker decomposition we assume that the factor matrices are orthogonal. Note that, the unitarity of matrices \mathbf{U}_i implies the following relation that can be seen as the inverse of Equation (1.4.5)

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{U}_1^\top \times_2 \mathbf{U}_2^\top \cdots \times_{p-1} \mathbf{U}_{p-1}^\top \times_p \mathbf{U}_p^\top, \quad (1.4.9)$$

which results from reordering the terms in the identity $\mathcal{G} = \mathcal{G} \times_1 \mathbf{U}_1^\top \mathbf{U}_1 \times_2 \mathbf{U}_2^\top \mathbf{U}_2 \cdots \times_{p-1} \mathbf{U}_{p-1}^\top \mathbf{U}_{p-1} \times_p \mathbf{U}_p^\top \mathbf{U}_p$.

We proceed by stating a theorem on the rank of a Tucker decomposition.

Theorem 1. *The Tucker-rank of a tensor \mathcal{T} is given by the ranks of its matricizations, i.e., $\text{rank}(\mathcal{T}_{(i)})$.*

PROOF. To prove this, we first show that the i -th component of the tucker-rank of \mathcal{T} , which is denoted by r_i with $i \in [p]$, is always lower-bounded by the matrix rank of the i -th

matricization of \mathcal{T} . We utilize a useful formula which gives the matricized version of (1.4.5):

$$\mathcal{T}_{(i)} = \mathbf{U}_i \mathcal{G}_{(i)} (\mathbf{U}_n \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_{i-1} \otimes \mathbf{U}_{i+1} \otimes \cdots \otimes \mathbf{U}_1)^\top \quad (1.4.10)$$

Since $\mathbf{U}_i \in \mathbb{R}^{d_i \times r_i}$, we have $\text{rank}(\mathcal{T}_{(i)}) \leq r_i$; that is equivalent to having r_i lower-bounded by $\text{rank}(\mathcal{T}_{(i)})$. Next, we show that this lower-bound is in fact reachable; that is, for any tensor \mathcal{T} , there always exists a Tucker representation of multi-rank $(r_i)_{i=1}^p$ with $r_i = \text{rank}(\mathcal{T}_{(i)})$. The proof of this part is constructive and results in an algorithm, called higher-order singular value decomposition (HOSVD) [72], which constructs the Tucker decomposition of a given tensor \mathcal{T} . The proof is based on a key observation; that for any p -th order tensor \mathcal{T} , with the SVD decompositions of the mode- i matricizations as $\mathcal{T}_{(i)} = \mathbf{U}_i \Sigma_i \mathbf{V}_i^\top, i \in [p]$, the following identity is valid

$$\mathcal{T} = \mathcal{T} \times_1 \mathbf{U}_1 \mathbf{U}_1^\top \times_2 \mathbf{U}_2 \mathbf{U}_2^\top \cdots \times_p \mathbf{U}_p \mathbf{U}_p^\top \quad (1.4.11)$$

To prove this, we first show $\mathcal{T} = \mathcal{T} \times_1 \mathbf{U}_1 \mathbf{U}_1^\top$. Notice that the i -mode matricization of a tensor is uniquely defined, therefore, if we prove this equality between a mode- i matricization of both sides, that implies the equality for the tensors as well. To show $\mathcal{T}_{(1)} = (\mathcal{T} \times_1 \mathbf{U}_1 \mathbf{U}_1^\top)_{(1)}$, we write

$$\begin{aligned} (\mathcal{T} \times_1 \mathbf{U}_1 \mathbf{U}_1^\top)_{(1)} &= \mathbf{U}_1 \mathbf{U}_1^\top \mathcal{T}_{(1)} \\ &= \mathbf{U}_1 \mathbf{U}_1^\top \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^\top = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^\top = \mathcal{T}_{(1)} \end{aligned} \quad (1.4.12)$$

This is also true for any other mode matricization, i.e., we have $\mathcal{T} = \mathcal{T} \times_i \mathbf{U}_i \mathbf{U}_i^\top$. This leads to Equation (1.4.11). Now, by reordering Equation (1.4.11), we get

$$\mathcal{T} = (\mathcal{T} \times_1 \mathbf{U}_1^\top \times_2 \mathbf{U}_2^\top \cdots \times_p \mathbf{U}_p^\top) \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_p \mathbf{U}_p \quad (1.4.13)$$

By defining the core tensor as $\mathcal{G} = \mathcal{T} \times_1 \mathbf{U}_1^\top \times_2 \mathbf{U}_2^\top \cdots \times_p \mathbf{U}_p^\top$ and factor matrices as \mathbf{U}_i for $i = 1, \dots, p$, the Tucker decomposition with the minimal rank tuple $(r_i)_{i=1}^p$ is constructed. This gives the HOSVD algorithm for the exact decomposition of a tensor in a Tucker representation. \square

Finally, without going into the details, we mention that the HOSVD algorithm also gives a quasi-optimal result for the low-rank Tucker decomposition of a tensor, i.e., with multi-rank $(r'_i)_{i=1}^p < (\text{rank}(\mathcal{T}_{(i)}))_{i=1}^p$. Although finding the best low-rank approximation is a NP-hard problem [71], the quasi-optimal approximation is obtained by arranging only the first r'_i left singular vectors of each $\mathcal{T}_{(i)}$ in a unitary matrix $\mathbf{U}'_i \in \mathbb{R}^{d_i \times r'_i}$ (truncated SVD) and defining the approximate core tensor and factor matrices as $\mathcal{G} = \mathcal{T} \times_1 \mathbf{U}'_1 \times_2 \mathbf{U}'_2 \cdots \times_p \mathbf{U}'_p$ and $\{\mathbf{U}'_i\}_{i=1}^p$ respectively. The quasi-optimality of this algorithm means that if \mathcal{T}^* is the best low-rank approximation of \mathcal{T} in terms of the Frobenius norm, the approximate tensor \mathcal{T}'

resulting from the HOSVD satisfies the following inequality [72]

$$\|\mathcal{T} - \mathcal{T}'\|_F \leq \sqrt{3}\|\mathcal{T} - \mathcal{T}^*\|_F. \quad (1.4.14)$$

1.4.3. Tensor Train (TT)

Another important TN is the tensor train decomposition [12], also known as matrix product state (MPS) [13, 15], which factorizes a n -th order tensor \mathcal{T} in the following form



This corresponds to

$$\mathcal{T}_{i_1, i_2, \dots, i_n} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{n-1}=1}^{r_{n-1}} (\mathcal{G}_1)_{i_1, \alpha_1} (\mathcal{G}_2)_{\alpha_1, i_2, \alpha_2} \cdots (\mathcal{G}_{n-1})_{\alpha_{n-2}, i_{n-1}, \alpha_{n-1}} (\mathcal{G}_n)_{\alpha_{n-1}, i_n} \quad (1.4.15)$$

where the tuple $(r_i)_{i=1}^{n-1}$ associated with the TT representation is called TT-rank. One special point about the tensor train representation is that the TT-rank of the *minimal* tensor train decomposition of a tensor can be determined in terms of some matricizations of the tensor[12]. However, these are not the mode- i matricizations. We define a new type of matricization for a p -th order tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ denoted by $\mathcal{T}_{[k]} \in \mathbb{R}^{\prod_{i=1}^k d_i \times \prod_{j=k+1}^p d_j}$ for $k \in \{1, \dots, p-1\}$ with the components written as

$$(\mathcal{T}_{[k]})_{i_1, \dots, i_k; i_{k+1}, \dots, i_p} = \mathcal{T}_{i_1, \dots, i_p} \quad (1.4.16)$$

That means $\mathcal{T}_{[k]} \in \mathbb{R}^{d_1 d_2 \cdots d_k \times d_{k+1} d_{k+2} \cdots d_p}$. First, we prove that the TT-rank of \mathcal{T} cannot be smaller than the rank of these matricizations. Consider the matricization defined in Equation (1.4.16) for a tensor \mathcal{T} in tensor train format. From Equations (1.4.15) and (1.4.16), we have

$$\begin{aligned} (\mathcal{T}_{[k]})_{i_1, \dots, i_k; i_{k+1}, \dots, i_p} &= \sum_{\alpha_1, \dots, \alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{p-1}} (\mathcal{G}_1)_{i_1, \alpha_1} \cdots (\mathcal{G}_k)_{\alpha_{k-1}, i_k, \alpha_k} (\mathcal{G}_{k+1})_{\alpha_k, i_{k+1}, \alpha_{k+1}} \cdots (\mathcal{G}_p)_{\alpha_{p-1}, i_p} \\ &= \sum_{\alpha_k} \left(\sum_{\alpha_1, \dots, \alpha_{k-1}} (\mathcal{G}_1)_{i_1, \alpha_1} \cdots (\mathcal{G}_k)_{\alpha_{k-1}, i_k, \alpha_k} \right) \left(\sum_{\alpha_{k+1}, \dots, \alpha_{p-1}} (\mathcal{G}_{k+1})_{\alpha_k, i_{k+1}, \alpha_{k+1}} \cdots (\mathcal{G}_p)_{\alpha_{p-1}, i_p} \right) \end{aligned} \quad (1.4.17)$$

By doing some reshaping, we can see the first factor inside the parenthesis as a matrix in $\mathbb{R}^{d_1 \cdots d_k \times \alpha_k}$ and the second term in the parenthesis as another matrix in $\mathbb{R}^{\alpha_k \times d_{k+1} \cdots d_p}$. Then, from this matrix product, we infer that the matricization $\mathcal{T}_{[k]}$ has rank at most equal to r_k . Therefore, so far we have proved that the k -th component of the TT-rank tuple is greater than or equal to the rank of the matricization $\mathcal{T}_{[k]}$. Now, we show that the equality is in fact possible; that is, for any tensor of arbitrary order p , the minimal TT representation exists, which is a tensor train where each component k of the TT-rank is given by the rank

of the corresponding matricization $\mathcal{T}_{[k]}$. The proof is constructive and results in an algorithm known as TT-SVD.

We consider the SVD of $\mathcal{T}_{[1]}$, i.e., $\mathcal{T}_{[1]} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^\top$. Note that $\mathbf{U}_1 \in \mathbb{R}^{d_1 \times r_1}$ and intuitively this matrix \mathbf{U}_1 can serve as the first core of the TT decomposition of \mathcal{T} . Let's call the other part of the SVD decomposition \mathbf{W}_1 , that is $\mathbf{W}_1 = \mathbf{\Sigma}_1 \mathbf{V}_1^\top \in \mathbb{R}^{r_1 \times d_2 d_3 \cdots d_p}$. Also, we introduce the p -th order tensor $\mathcal{W}_1 \in \mathbb{R}^{r_1 \times d_2 \times d_3 \times \cdots \times d_p}$ which is made by rearranging the components of the matrix \mathbf{W}_1 . To continue, a key step is to show that the rank of the $[k]$ -th matricization of the p -th order tensor \mathcal{W}_1 is less than or equal to the corresponding matricization rank of the initial tensor \mathcal{T} . From the SVD decomposition of $\mathcal{T}_{[1]}$ and the definition of \mathbf{W}_1 , we have

$$\mathbf{W}_1 = \mathbf{U}_1^\top \mathcal{T}_{[1]} \quad (1.4.18)$$

From Equation (1.4.17) for the matricization $\mathcal{T}_{[k]}$, we have

$$\mathcal{T}_{i_1, \dots, i_p} = \sum_{\alpha_k=1}^{r_k} \mathbf{M}_{i_1, \dots, i_k; \alpha_k} \mathbf{N}_{\alpha_k; i_{k+1}, \dots, i_p} \quad (1.4.19)$$

with $\mathbf{M}_{i_1, \dots, i_k; \alpha_k}$ and $\mathbf{N}_{\alpha_k; i_{k+1}, \dots, i_p}$ being respectively equal to $\sum_{\alpha_1, \dots, \alpha_{k-1}} (\mathcal{G}_1)_{i_1, \alpha_1} \cdots (\mathcal{G}_k)_{\alpha_{k-1}, i_k, \alpha_k}$ and $\sum_{\alpha_{k+1}, \dots, \alpha_{p-1}} (\mathcal{G}_{k+1})_{\alpha_k, i_{k+1}, \alpha_{k+1}} \cdots (\mathcal{G}_p)_{\alpha_{p-1}, i_p}$. Then, Equation (1.4.18) results in the following relation

$$(\mathcal{W}_1)_{\alpha_1, i_2, \dots, i_p} = (\mathbf{W}_1)_{\alpha_1, i_2 \cdots i_p} = \sum_{i_1} (\mathbf{U}_1)_{i_1, \alpha_1} (\mathcal{T}_{[1]})_{i_1, i_2, i_3, \dots, i_p} = \sum_{i_1} (\mathbf{U}_1)_{i_1, \alpha_1} \mathcal{T}_{i_1, i_2, i_3, \dots, i_p} \quad (1.4.20)$$

This can be rewritten as

$$\begin{aligned} (\mathcal{W}_1)_{\alpha_1, i_2, \dots, i_p} &= \sum_{i_1} (\mathbf{U}_1)_{i_1, \alpha_1} \mathcal{T}_{i_1, i_2, i_3, \dots, i_p} = \sum_{i_1} (\mathbf{U}_1)_{i_1, \alpha_1} \sum_{\alpha_k=1}^{r_k} \mathbf{M}_{i_1, \dots, i_k; \alpha_k} \mathbf{N}_{\alpha_k; i_{k+1}, \dots, i_p} \\ &= \sum_{\alpha_k=1}^{r_k} \left(\sum_{i_1} (\mathbf{U}_1)_{i_1, \alpha_1} \mathbf{M}_{i_1, \dots, i_k; \alpha_k} \right) \mathbf{N}_{\alpha_k; i_{k+1}, \dots, i_p} \end{aligned} \quad (1.4.21)$$

Now, since the left-hand side of this equation is equal to $(\mathcal{W}_{1[k]})_{\alpha_1, i_2, \dots, i_k; i_{k+1}, \dots, i_p}$, we conclude that the rank of any matricization $\mathcal{W}_{[k]}$ is at most equal to r_k .

Knowing this, we can reshape \mathbf{W}_1 as $\mathbf{W}_1^{\text{reshaped}} \in \mathbb{R}^{r_1 d_2 \times d_3 \cdots d_p}$ and again apply a SVD decomposition as $\mathbf{W}_1 = \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^\top$ with $\mathbf{U}_2 \in \mathbb{R}^{r_1 d_2 \times r_2}$ and $\mathbf{W}_2 = \mathbf{\Sigma}_2 \mathbf{V}_2^\top \in \mathbb{R}^{r_2 \times d_3 \cdots d_p}$. By reshaping \mathbf{U}_2 as $\mathcal{U}_2 \in \mathbb{R}^{r_1 \times d_2 \times r_2}$, the second core tensor of the TT decomposition is formed. By proceeding in this same way and iteratively applying SVD we find all core tensors and thus, the tensor train representation of \mathcal{T} with the TT-rank (r_1, \dots, r_{p-1}) is constructed; note that each r_k is the rank of the corresponding matricization $\mathcal{T}_{[k]}$. Finally, since the SVD decomposition always exists for any matrix, this constructive proof shows the existence of the *minimal* tensor train for any arbitrary tensor. Figure 1.7 illustrates the application of TT-SVD algorithm on a 4-th order tensor.

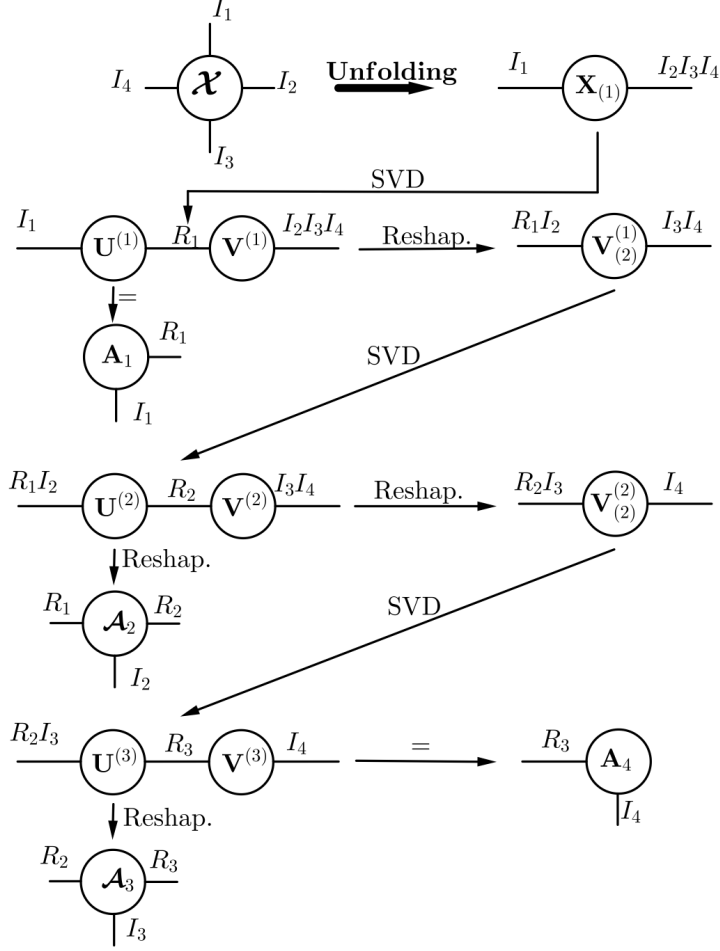


Figure 1.7. Figure from [6]. TT-SVD algorithm applied to a 4-th order tensor

It is worth mentioning that as in the Tucker case, the TT-SVD algorithm can be modified to find an approximate TT decomposition rather than the exact one. This is again realized by using truncated SVD inside the TT-SVD algorithm instead of the ordinary SVD. The resulting TT decomposition is quasi-optimal; that is, if \mathcal{T}^* is the best low-rank approximation of a given tensor \mathcal{T} in terms of the Frobenius norm, the approximate tensor \mathcal{T}' resulting from the (truncated) TT-SVD satisfies the following inequality [12]

$$\|\mathcal{T} - \mathcal{T}'\|_F \leq \sqrt{p-1} \|\mathcal{T} - \mathcal{T}^*\|_F. \quad (1.4.22)$$

As the final point regarding the TT decomposition, in [12], it is shown how different tensor operations such as summation, elementwise (Hadamard) product as well as inner product of TT tensors reduce to some operations on the core tensors of the TT tensors and result in tensors with bounded TT-rank in terms of the ranks of the initial tensors. Besides that, the inner product of TT tensors with uniform TT-rank r has linear (in terms of the tensor orders) time complexity $\mathcal{O}(ndr^4)$ which is a substantial improvement over the exponential time d^n which is ordinarily required for the inner product of two n -th order tensors with

uniform dimension d . Table 1.1 summarizes some significant differences between the tensor networks that we have seen so far.

1.4.4. Other decompositions: Hierarchical Tucker and Tensor Ring

As could be expected, there are other tensor networks that we have not considered in our short introduction to the subject. Figure 1.8 represents some of those tensor networks. Two of them which can be seen as the generalization of the tensor networks that we saw above, are tensor ring (TR) [41] (also known as periodic MPS) and PEPS decompositions. They have initially emerged in quantum physics and recently gained interest in the machine learning community (see e.g., [65, 20, 73, 74]). Each one of these TNs generalize tensor trains in their own way. Especially TR overcomes two limitations of TT; first, that the two bordering tensors of TT are constrained to have dimension 1 along one of their three modes, secondly, the bond dimension of TT increases from the borders towards the center. The TR decomposition expresses each component of a p -th order tensor \mathcal{T} as the trace of a product of slices of p core tensors $\mathcal{G}^{(1)} \in \mathbb{R}^{r_0 \times d_1 \times r_1}$, $\mathcal{G}^{(2)} \in \mathbb{R}^{r_1 \times d_2 \times r_2}$, \dots , $\mathcal{G}^{(p)} \in \mathbb{R}^{r_{p-1} \times d_p \times r_p}$ with $r_p = r_0$. As an example, for a 4-th order tensor we have $\mathcal{T}_{i_1, i_2, i_3, i_4} = \text{Tr}(\mathcal{G}_{:, i_1, :}^{(1)} \mathcal{G}_{:, i_2, :}^{(2)} \mathcal{G}_{:, i_3, :}^{(3)} \mathcal{G}_{:, i_4, :}^{(4)})$. The tensor train (TT) decomposition is then a particular case of the tensor ring where r_0 is equal to 1.

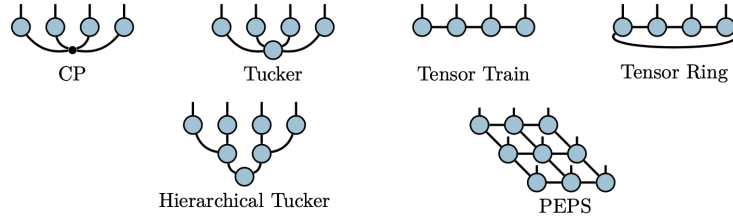


Figure 1.8. Illustration of some common tensor networks.

One more tensor network of special interest is the hierarchical Tucker (HT) decomposition initially introduced in [39, 40]. An important property of this TN is its high expressive power compared to some less expressive TNs like the CP. In the final section of this chapter we show an equivalence between the CP and the HT tensor networks on the one hand and some specific shallow and deep neural networks on the other hand. This is a part of a study of the expressive power of shallow and deep neural networks based on a tensor network analysis [32]. Before going through that, in the next section we exemplify a TN learning model which was introduced in [1], based on TT.

	CP	Tucker	Tensor Train
# Parameters	$r \sum_{i=1}^p d_i$	$\prod_{i=1}^p r_i + \sum_{i=1}^p r_i d_i$	$\sum_{i=2}^{p-1} r_i r_{i-1} d_i + d_1 r_1 + d_p r_{p-1}$
Computing the rank	NP-hard	Polynomial	Polynomial
Low-rank approximations	?	Quasi-optimal algorithm (HOSVD)	Quasi-optimal algorithm (TTSVD)

Table 1.1. Some properties of CP, Tucker and TT compared against each other

1.5. Classification with Tensor Train Weight

The goal of this section is to briefly show an example of a tensor-network-based learning model. In [26] the interaction between data features has been modeled by mapping input data into a tensor in a high dimensional space. As a way to avoid the curse of dimensionality, the model is regularized by using the TT representation of the exponentially high-dimensional weight tensors; the rank of the tensor is a hyperparameter of the model which controls the amount of regularization. In a separate work [1], the authors closely follow the same idea to perform image classification task on MNIST dataset and obtain a test error of 0.97%. The idea is to work with data projected into a high-dimensional representation space, as in kernel models. In order to deal with the exponentially high-dimensional weight tensor, they suggest a TT representation of this tensor. In this way, the number of model parameters scales linearly with the tensor dimension, rather than exponentially. The rest of this section is based on the tensor train classifier proposed in [1] which is one of the first works showing the potential of quantum-inspired tensor networks for supervised learning.

Let $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ be a sample drawn i.i.d. from an unknown distribution D , where each $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$.

In [1], Stoudenmire and Schwab propose to map each element of the input vector to a two-dimensional *local* space. This is called a *local feature map*. Figure 1.9 shows an example of this feature map for the case of image input data where each image pixel is mapped to a 2-dim vector. Then, the input tensor in the high-dimensional space is constructed by taking the outer product of these *local* feature maps; that means, if we consider $\mathbf{x} = (x_1, \dots, x_p)$, the suggested M -dimensional local feature map over every component x_i is of the general form $\phi(x_i) = (\phi_1(x_i) \ \phi_2(x_i) \ \dots \ \phi_M(x_i))^\top$. Then, the representation of the input data, is defined as the outer product of these individual local maps

$$\mathbf{x} = (x_1, x_2, \dots, x_p) \mapsto \Phi(\mathbf{x}) = \phi(x_1) \circ \phi(x_2) \cdots \circ \phi(x_p) \in (\mathbb{R}^M)^{\otimes p} \quad (1.5.1)$$

The classifier is given by the contraction of the data tensor and a p -th order weight tensor \mathcal{W} , as below

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i_1, i_2, \dots, i_p=1}^M \mathcal{W}_{i_1, i_2, \dots, i_p} \phi(x_1)_{i_1} \phi(x_2)_{i_2} \dots \phi(x_p)_{i_p} \right) = \text{sign}(\langle \mathcal{W}, \Phi(\mathbf{x}) \rangle) \quad (1.5.2)$$

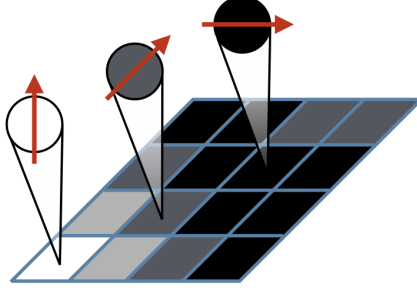


Figure 1.9. Figure from [1]. Each pixel value of a grayscale image is mapped to a normalized two-component vector.

The weight tensor \mathcal{W} needs to be in the space $(\mathbb{R}^M)^{\otimes p}$ which is typically of very high dimension; the authors of [1] overcome this problem by considering \mathcal{W} in the tensor train representation of low rank given in Eq. (1.4.15). That means

$$\mathcal{W}_{i_1, i_2, \dots, i_p} = \sum_{\alpha_i=1}^{r_i} \mathcal{T}_{1, i_1}^{\alpha_1} \mathcal{T}_{2, i_2}^{\alpha_1, \alpha_2} \dots \mathcal{T}_{p-1, i_{p-1}}^{\alpha_{p-2}, \alpha_{p-1}} \mathcal{T}_{p, i_p}^{\alpha_{p-1}} \quad (1.5.3)$$

or pictorially, as in Figure 1.10.

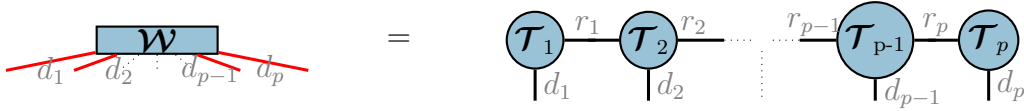


Figure 1.10. Decomposition of the weight tensor as a tensor train

1.6. Equivalence of TNs with Convolutional Arithmetic Circuits

In this section, we introduce two more examples of tensor-network-based learning models which have CP and HT as their weight tensor. Following the lines of [7], we show how they are respectively equivalent to shallow and deep convolutional arithmetic circuits (CAC) which are special types of sum-product neural networks.

As a *reminder*, For a tensor \mathcal{T} of order p and another tensor \mathcal{S} of order q , i.e., $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_p}$ and $\mathcal{S} \in \mathbb{R}^{d'_1 \times \dots \times d'_q}$, we have $\mathcal{T} \circ \mathcal{S} \in \mathbb{R}^{d_1 \times \dots \times d_p \times d'_1 \times \dots \times d'_q}$. This tensor product is defined by

$$(\mathcal{T} \circ \mathcal{S})_{i_1, \dots, i_p, j_1, \dots, j_q} = \mathcal{T}_{i_1, \dots, i_p} \mathcal{S}_{j_1, \dots, j_q} \quad (1.6.1)$$

Also, the CP decomposition of a p -th order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_p}$ given by $\mathcal{X} = \sum_{i=1}^r \mathbf{u}_i^1 \circ \mathbf{u}_i^2 \circ \dots \circ \mathbf{u}_i^p$, can be represented as a tensor network as in Figure 1.5, if we define p matrices $\{\mathbf{U}^i \in \mathbb{R}^{d_i \times r}\}_{i=1}^p$, where each \mathbf{U}^i is made up of r vectors \mathbf{u}_r^i as its columns.

1.6.1. CP Model as a Shallow CAC/CNN

While following [7], to unify our notation for the TN models, we use the same notation as in the previous section for the tensor train model. Assume the tensorial data $\Phi(\mathbf{x})$ of potentially large dimension in $(\mathbb{R}^M)^{\otimes p}$ space. A linear model to do classification or regression on this input space consists of tensor weights of the same dimension:

$$f(\mathbf{x}) = \sum_{i_1, \dots, i_p} \mathbf{W}_{i_1, \dots, i_p} \Phi(\mathbf{x})_{i_1, \dots, i_p} \quad (1.6.2)$$

$\{\mathbf{W}_{i_1, \dots, i_p}\}_{i_1, \dots, i_p=1}^M$ are M^p entries of the weight tensor \mathbf{W} that is to learn. A possible TN representation of \mathbf{W} is the CP decomposition in Equation (1.4.1). Replacing that in Equation (1.6.2), we get

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i_1, \dots, i_p} \left(\sum_{l=1}^r a_l \mathbf{a}_{l, i_1}^1 \mathbf{a}_{l, i_2}^2 \cdots \mathbf{a}_{l, i_p}^p \right) \Phi(\mathbf{x})_{i_1, \dots, i_p} \\ &= \sum_{i_1, \dots, i_p} \left(\sum_{l=1}^r a_l \mathbf{a}_{l, i_1}^1 \mathbf{a}_{l, i_2}^2 \cdots \mathbf{a}_{l, i_p}^p \right) \phi_{i_1}(\mathbf{x}_1) \cdots \phi_{i_p}(\mathbf{x}_p) \end{aligned} \quad (1.6.3)$$

with $\mathbf{a}_l^k \in \mathbb{R}^M$ for all $k = 1, \dots, p$. The last expression can be reorganized so that $f(\mathbf{x})$ takes the following form

$$f(\mathbf{x}) = \sum_{l=1}^r a_l \prod_{k=1}^p \sum_{i_k=1}^M \mathbf{a}_{l, i_k}^k \phi_{i_k}(\mathbf{x}_k) \quad (1.6.4)$$

From this, we can describe a shallow CNN that is equivalent to the above CP-classifier model. To elaborate on this analogy with the CNN, we think of the input data in $(\mathbb{R}^M)^{\otimes p}$ as an image. In that case, p can be considered as the spatial size of the image, like height \times width. Also, M is interpreted as the number of channels, which is 3 for RGB images. Each $\phi_{i_k}(\mathbf{x}_k)$ is one pixel in the i_k -th channel which is convolved by r different kernel functions (filters) $\{\mathbf{a}_{l, i_k}^k\}_{l=1}^r$. These kernels are scalars (or more precisely, each set $\{\mathbf{a}_{l, i_k}^k\}_{i_k=1}^M$ is a kernel of volume $M \times 1 \times 1$) and hence this is a 1×1 convolution. As in ordinary CNN, we have a summation over M input channels which is done by the innermost summation over the index i_k . The output of this summation is another representation of the data, again of spatial size p , but now in r different channels. The next operation is a *global product pooling* which multiplies all p elements of each of the r channels together and downscales the spatial size to 1, while keeping the number of channels r . This operation is shown by the product over the index k and the result of that is a vector of dimension r . Finally, the inner product of this vector with a vector made up of a_l 's results in an scalar which is the score of the input data \mathbf{x} . This is done in the output layer of the corresponding CNN through the summation over the index l . This convolutional arithmetic circuit is illustrated in Figure 1.11 from [7]. Note that we have only one layer of convolution and hence a shallow CAC.

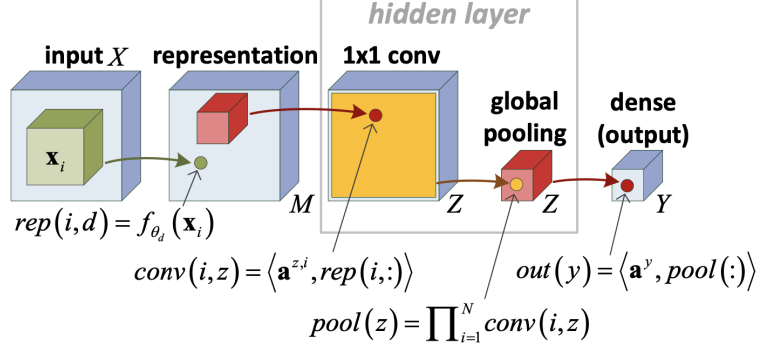


Figure 1.11. Figure from [7]. CAC corresponding to the CP decomposition of the weight tensor of a linear model.

1.6.2. Hierarchical Tucker Decomposition as a Deep CAC

In order to follow the analysis of HT-classifier and its corresponding convolutional arithmetic circuit (CAC) in [7], we start by reviewing the representation of the tensor product of matrices in tensor network diagrams. First, we remember from the CP decomposition that a summation of the form $\sum_{i=1}^r a_i \mathbf{b}_i \circ \mathbf{c}_i$ is equivalent to a matrix product \mathbf{BAC}^T , where \mathbf{B} and \mathbf{C} are matrices of dimension $d_B \times r$ and $d_C \times r$, with d_B and d_C being the dimensions of vectors $\{\mathbf{b}_i\}_{i=1}^r$ and $\{\mathbf{c}_i\}_{i=1}^r$ respectively. \mathbf{A} is a $r \times r$ diagonal matrix with the scalars $\{a_i\}_{i=1}^r$ as its entries. Therefore, the above summation is represented as the tensor network



. If we now consider a similar summation with matrices $\mathbf{B}_i, \mathbf{C}_i$ instead of the vectors $\mathbf{b}_i, \mathbf{c}_i$, i.e., $\sum_{i=1}^r a_i \mathbf{B}_i \circ \mathbf{C}_i$, then, by stacking the matrices \mathbf{B}_i into a tensor \mathcal{B} and \mathbf{C}_i 's



into \mathcal{C} , this summation is represented by the diagram

where \mathbf{A} is again a $r \times r$ diagonal matrix with the scalars $\{a_i\}_{i=1}^r$ as its entries. With this intuition about the TN representation of tensor decompositions including the outer product of tensors of arbitrary orders, here we just give a high-level overview of the classifier model that was detailed in [7].

It is in fact a special HT with a binary tree structure; moreover, the core tensors enjoy some types of symmetries that makes it possible to represent all of them by tensors of order at most two, i.e., with matrices. Following the notation of [7], The predictor function is defined recursively. The first step of the recursion includes the outer product of the vector parameters \mathbf{v}_j^α

$$\psi^{1,j,\gamma} = \sum_{\alpha=1}^{r_0} a_\alpha^{1,j,\gamma} \mathbf{v}_{2j-1}^\alpha \circ \mathbf{v}_{2j}^\alpha, \quad j \in [p/2], \quad \gamma \in [r_1], \quad (1.6.5)$$

where $\mathbf{v}_{2j}^\alpha, \mathbf{v}_{2j-1}^\alpha \in \mathbb{R}^d$, $\forall j \in [p/2], \forall \alpha \in [r_0]$.

TN representation of this step is shown in Figure 1.12. As described above, \mathbf{V}_i matrices in this figure for odd i values, $i = 2j - 1$, have vectors \mathbf{v}_{2j-1}^α as their columns, i.e., $\mathbf{V}_i \in \mathbb{R}^{d \times r_0}$.

Also, for each $j \in [\frac{p}{2}]$ and each $\gamma \in [r_1]$, we can put the entries $\{a_\alpha^{1,j,\gamma}\}_{\alpha=1}^{r_0}$ in a vector in \mathbb{R}^{r_0} . Then, for each j , by placing all r_1 vectors as columns of matrices $\mathbf{A}_j \in \mathbb{R}^{r_0 \times r_1}$, the following figure illustrates the outer product of j tensors $\psi^{1,j,\gamma}$ of Equation (1.6.5), for $j = 4$.

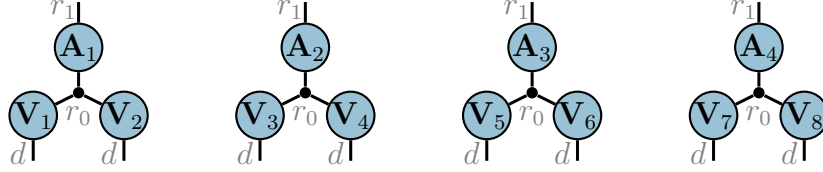


Figure 1.12. Illustration of the Equation (1.6.5), i.e., the first recursion step for the construction of the Hierarchical Tucker tensor. Note that r_0 shows the dimension of the hyper-edge along all its modes.

The next step of the recursion combines $\psi^{1,j,\gamma}$ s to calculate $\psi^{2,j,\gamma}$ s, and so on and so forth, for all $\psi^{l,j,\gamma}$ s in terms of $\psi^{l-1,j,\gamma}$ s. Finally, the last equation in the recursion gives the weight tensor \mathcal{W} , which is the same as $\psi^{L,j=1,\gamma=1}$ in terms of $\psi^{L-1,j,\gamma}$ s.

$$\begin{aligned}
 \psi^{2,j,\gamma} &= \sum_{\alpha=1}^{r_1} a_\alpha^{2,j,\gamma} \psi^{1,2j-1,\alpha} \circ \psi^{1,2j,\alpha}, \quad j \in [\frac{p}{4}], \quad \gamma \in [r_2] \\
 &\dots \\
 \psi^{l,j,\gamma} &= \sum_{\alpha=1}^{r_{l-1}} a_\alpha^{l,j,\gamma} \psi^{l-1,2j-1,\alpha} \circ \psi^{l-1,2j,\alpha}, \quad j \in [\frac{p}{2^l}], \quad \gamma \in [r_l] \\
 &\dots \\
 \mathcal{W} &= \sum_{\alpha=1}^{r_{L-1}} a_\alpha^L \psi^{L-1,1,\alpha} \circ \psi^{L-1,2,\alpha}
 \end{aligned} \tag{1.6.6}$$

In terms of TN format, the whole weight tensor defined by Equations (1.6.6) has the HT representation in Figure 1.13, for a 8-th order tensor

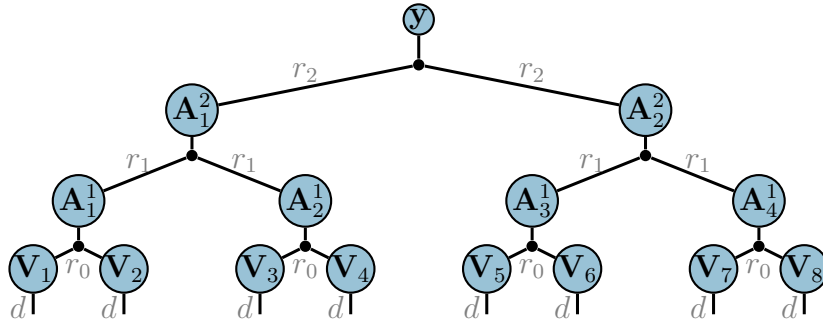


Figure 1.13. Illustration of the Hierarchical Tucker representation of a tensor of order eight, defined by Equation (1.6.6).

With the same observations as mentioned above, each \mathbf{A}_j^l is made by arranging all r_l vectors in $\mathbb{R}^{r_{l-1}}$, i.e., $\{\{a_\alpha^{l,j,\gamma}\}_{\alpha=1}^{r_{l-1}}\}_{\gamma=1}^{r_l}$, as columns of a matrix in $\mathbb{R}^{r_{l-1} \times r_l}$. Finally, the vector $\mathbf{y} \in \mathbb{R}^{r_{L-1}}$ contains $\{a_\alpha^L\}_{\alpha=1}^{r_{L-1}}$ as its entries.

Using a similar analysis as in Section 1.6.1, the HT-based classifier is equivalent to a deep CAC as in Figure 1.14 [7].

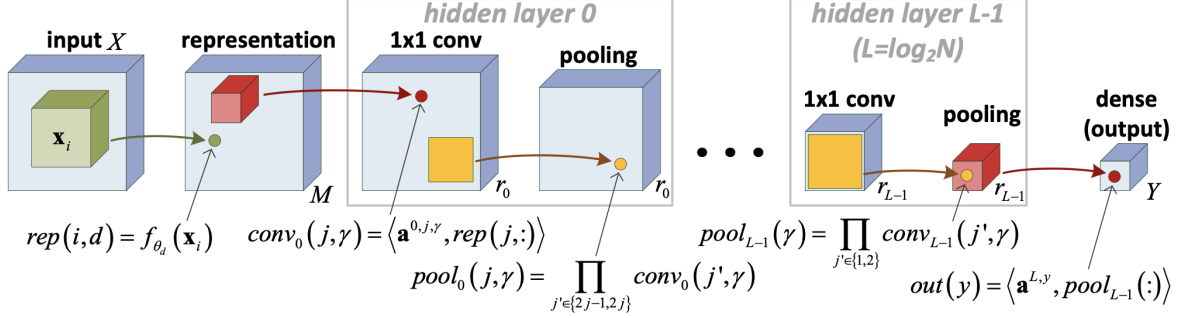


Figure 1.14. Figure from [7]. Deep CAC corresponding to the HT decomposition of the weight tensor.

Note that in each layer of this CAC, we again have 1×1 convolutions; however, the collapse of the spatial dimension of the image occurs gradually over the layers, rather than at once which was the case in the CP model. That means, instead of the global product pooling in the CP model, here at each level, local product pooling operations are applied over size-2 windows (because the HT tree is chosen to be binary in this case), and hence the number of layers of this deep CAC is $\log p$.

Finally, we mention one important result in [7] on the expressiveness of shallow and deep CACs based on this equivalence. That is, in order for a shallow network to express a function captured by a random deep CAC network, with probability 1 in the corresponding function space, the former needs exponentially more parameters compared to the latter. The proof is based on this observation that representing a HT network in CP format, with probability 1 requires exponentially more parameters in terms of the order of the tensor. This result comes from the fact that the CP rank of a tensor is always larger than or equal to the rank of any of its matricizations. Since with probability 1, there always exists [7] a special matricization of the HT network for which the rank is at least $r^{\frac{p}{2}}$, with r being the HT rank and p the order of the tensor, the corresponding CP rank will be higher than $r^{\frac{p}{2}}$; this results in exponentially more parameters for the CP than for the HT representation (again with probability 1). This is what the authors call *complete efficiency*, which means the set of functions for which this statement does not hold, has measure zero in the corresponding function space. Also interestingly, in [32] it is shown that this depth efficiency would no longer be *complete* for the case of neural networks with ReLU non-linearity.

Chapter 2

Generalization Bound and Complexity Measures

2.1. Introduction

As shortly explained in the first chapter, within the framework of the empirical risk minimization (ERM) for supervised learning, generalization bounds upper-bound the gap between the empirical and the true risks, i.e., between $\hat{R}_S(h)$ and $R(h)$. In this thesis, we focus on *uniform* generalization bounds, which for a class of functions denoted by \mathcal{H} , bound the generalization gap uniformly for any hypothesis $h \in \mathcal{H}$, as a function of the training sample size and of the complexity of \mathcal{H} . While there are many ways of measuring the complexity of \mathcal{H} , including VC-dimension, Rademacher complexity, metric entropy and covering numbers, we focus on the *VC-dimension* for classification tasks and its counterpart for real-valued functions, the *pseudo-dimension*, for completion and regression tasks. We begin with studying the calculation of the generalization bound for the classification task in detail. After that, we briefly review a similar problem for the regression task, as well as for the completion.

2.2. Classical Generalization Bounds for Classification

Let $S = \{(x_1, y_1) \dots, (x_n, y_n)\}$ be a sample drawn i.i.d. from an unknown distribution D . Based on this subset S , we want to find a function $h : \mathbb{R}^d \rightarrow \mathbb{R}$, such that for all $i \in [n]$, $y_i = \text{sign}(h(x_i)) \in \{-1, 1\}$. It is not always guaranteed that all (x_i, y_i) 's satisfy this relation; so, this will be an approximation and we are interested in quantifying its accuracy; we define the loss: $L(y, \hat{y}) = \mathbb{I}_{y \neq \hat{y}}$. From that, we can write $L_i = L(y_i, \text{sign}(h(x_i)))$ to denote the loss on the i -th example. The empirical risk of a function (or hypothesis) is defined as the average

of the loss function L over the entire sample S

$$\hat{R}_S(h) = \sum_{i=1}^n \frac{1}{n} L(y_i, \text{sign}(h(x_i))) \quad (2.2.1)$$

However, what we finally care about is the true risk defined as $R(h) = \mathbb{E}_{(x,y) \sim D}[L(y, \text{sign}(h(x)))]$. Given that we do not have access to all samples in the distribution D , we cannot calculate it. Instead, we ask the following question :

Does the empirical risk $\hat{R}_S(h)$ give an upper bound on the true risk, $R(h)$?

To answer this question, we first remind some facts regarding the loss function. First, we observe that the loss function $L_i = L(y_i, \text{sign}(h(x_i))) = \mathbb{I}_{y_i \neq \text{sign}(h(x_i))}$, as a random variable, has Bernoulli distribution with probability $p = R(h)$. This is a result of the expectation of the loss L being the true risk, from the definition of the true risk, and the fact that the 0 – 1 loss takes only two values.

Knowing that the loss has a Bernoulli distribution, we consider the implication of Theorem 19 for the loss as the random variable.

By replacing X_i 's of this theorem with losses L_i , and noting that the sample average of losses is the empirical risk, the theorem straightforwardly results in the following upper bound on the probability of the difference between the empirical risk and the true risk exceeding an arbitrary real value ϵ

$$\mathbb{P} \left[|\hat{R}_S(h) - R(h)| > \epsilon \right] \leq 2 \exp(-2n\epsilon^2), \quad (2.2.2)$$

with n being the size of the sample S . So far, we have considered only one hypothesis. However, most of the time the question is to find a bound when the hypothesis h is just one function out of a class of functions \mathcal{H} . Then, the problem would be to put an upper bound on the probability of having at least one hypothesis in \mathcal{H} like h_i for which $|\hat{R}_S(h_i) - R(h_i)| > \epsilon$. In most cases the set of hypotheses is infinite and our goal is to finally obtain a generalization bound for that case. Nevertheless, here we start with the finite class which is more straightforward and then generalize this formulation to the infinite case.

2.2.1. Finite Class of Hypotheses

We use the union bound which states that for a finite set of k events e_1, \dots, e_k , the probability that at least one of the k events happens, i.e., $\mathbb{P}(\cup_{i=1}^k e_i)$, is no greater than the sum of the probabilities of the individual events:

$$\mathbb{P} \left[\cup_{i=1}^k e_i \right] \leq \sum_{i=1}^k \mathbb{P}[e_i] \quad (2.2.3)$$

From the union bound and our earlier result (2.2.2) for one hypothesis, we find out the following relation, known as a uniform convergence bound

$$\mathbb{P} \left[\exists h_i \in \mathcal{H}, |\hat{R}_S(h_i) - R(h_i)| > \epsilon \right] \leq \sum_{i=1}^{|\mathcal{H}|} \mathbb{P} \left[|\hat{R}_S(h_i) - R(h_i)| > \epsilon \right] \leq 2|\mathcal{H}| \exp(-2n\epsilon^2) \quad (2.2.4)$$

As a side note, one might wonder about the bias-variance trade-off through the lens of this relation. To comment on this point, note that for highly complex hypothesis classes, that is for large values of $|\mathcal{H}|$, the above bound on the probability gets loose if the data sample is relatively small. That means, the event of the generalization error being remarkably different than the empirical error, becomes more likely. In this case, the empirical error is not a good estimation of the true error and we get a high *approximation error* due to a high-variance model. As we know from the standard statistical learning theory, in order to deal with models diagnosed with high variance, we either provide samples with more data points or regularize the function class by decreasing the cardinality of the set of hypotheses. We now observe how these two approaches result in tightening the generalization gap through the corresponding probability relation in Equation (2.2.4). Moreover, the bias-variance trade-off is seen here from the fact that although one can arbitrarily diminish $|\mathcal{H}|$ to tighten the bound in Equation (2.2.4), the hypothesis class needs to be large enough so that the target function is included in it or at least is close enough to this class. In fact, while the above probability bound is uniform, we are finally concerned with this bound for a learned function $h \in \mathcal{H}$ that should be as close as possible to a target function h^* . That means, in the complete learning setup, the objective to be minimized is $|\hat{R}_S(\hat{h}) - R(h^*)|$. This expression is equivalent to

$$|\hat{R}_S(\hat{h}) - R(h^*)| = |\hat{R}_S(\hat{h}) - R(h) + R(h) - R(h^*)| \leq |\hat{R}_S(\hat{h}) - R(h)| + |R(h) - R(h^*)| \quad (2.2.5)$$

The first term on the right-hand side is called the *estimation error* which is due to the use of a limited number of data points, i.e., is high for high-variance models. The second term on the other hand, which is the difference between the true errors of the learned model and the target model, estimates the best performance that we can get from a model in our model class. For this second term, known as the *approximation error*, to be small, we need to have a big enough hypothesis class to avoid high-bias models.

The bound on the likelihood in Equation (2.2.4), can be recast into a bound on the gap between the empirical and true risk. We define δ as the probability of this gap exceeding the value ϵ . Then, the inequality in (2.2.4) is equivalent to the following statement: with probability at least $1 - \delta$ over the choice of the sample S , the difference between the empirical

and the true risk is bounded as

$$|\hat{R}_S(h) - R(h)| \leq \sqrt{\frac{\log |2\mathcal{H}| - \log \delta}{2n}} \quad (2.2.6)$$

To extend this result to the infinite-size hypothesis class, we first notice that, in that case the above bound will be uninformative. As explained in the next section, to deal with this case, we introduce the notion of *restriction* of the infinite class to a finite sample.

2.2.2. Infinite Class of Hypotheses

Definition 2. *The restriction of a class of hypotheses \mathcal{H} to a set $S = \{x_1, \dots, x_n\}$ is defined as $\mathcal{H}_S = \{(h(x_1), \dots, h(x_n)) \mid h \in \mathcal{H}\}$.*

We remind that \mathcal{H} is a subset of all Boolean-valued functions on the space of input data points. Each member of \mathcal{H}_S is called a dichotomy or a sign pattern. Evidently, not all \mathcal{H} have enough capacity to generate all 2^n possible sign patterns for n data inputs. Therefore, in general, we get an upper bound on the cardinality of \mathcal{H}_S : $|\mathcal{H}_S| \leq 2^n$. In case of equality, we say that the class \mathcal{H} *shatters* the set S . It is important to note that \mathcal{H}_S and consequently the shattering property are strongly dependent on the data points by definition; while one set of n points can be shattered by a class of functions, typically there exist many other sets of size n that are not shattered by this same class \mathcal{H} . Based on this fact, we introduce the *growth function* .

Definition 3. *Let $\mathcal{H} \subset \{-1, +1\}^{\mathcal{X}}$ be a hypothesis class. The growth function $\Pi_{\mathcal{H}} : \mathbb{N} \rightarrow \mathbb{N}$ of \mathcal{H} is defined by*

$$\Pi_{\mathcal{H}}(n) = \sup_{S=\{x_1, \dots, x_n\} \subset \mathcal{X}} |\{(h(x_1), \dots, h(x_n)) \mid h \in \mathcal{H}\}|.$$

That is, the growth function of a hypothesis class for a given number n is the maximum cardinality of the restriction of that hypothesis class to a sample S of size n .

Growth function generalization bound Equipped with the new concept of the growth function, we prove a generalization bound for infinite hypothesis classes. We use the symmetrization lemma which is based on the concept of *restriction* introduced in Definition 2. As we see later, using this lemma interestingly results in a useful reduction of the analysis of the infinite class to that of a finite set.

The following theorem gives the generalization bound in terms of the growth function.

Theorem 4. *(Vapnik-Chervonenkis). Let \mathcal{H} be an infinite class of hypotheses. For any $\delta > 0$ with probability at least $1 - \delta$ over a random draw of a sample of finite size n , we have*

$$\forall h \in \mathcal{H}, \quad R(h) < \hat{R}_S(h) + 2\sqrt{\frac{\log \Pi_{\mathcal{H}}(2n) + \log \frac{4}{\delta}}{n}} \quad (2.2.7)$$

PROOF. To prove this theorem we use a symmetrization lemma and a corollary of Hoeffding's inequality.

Lemma 2.2.1. (*Symmetrization Lemma*) *Let S and S' be two random samples of size n drawn from a distribution D . Then for any $t > 0$, with large enough n such that $nt^2 \geq 2$, we have*

$$\mathbb{P}_{S \sim D} \left[\sup_{h \in \mathcal{H}} (R(h) - \hat{R}_S(h)) \geq t \right] \leq 2 \mathbb{P}_{S, S' \sim D} \left[\sup_{h \in \mathcal{H}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \frac{t}{2} \right], \quad (2.2.8)$$

This lemma relates the difference between the true risk and the empirical risk of one given sample, to the difference between empirical risks of that sample and another random sample of the same size, sometimes called the ghost sample. The proof can be found in Appendix B.

Next, we give a corollary of Hoeffding's theorem.

Corollary 2.2.2. *If $Z_1, \dots, Z_n, Z'_1, \dots, Z'_n$ are $2n$ i.i.d. random variables drawn from a Bernoulli distribution, then for all $\epsilon > 0$ we have*

$$\mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n Z'_i > \epsilon \right] \leq 2 \exp \left(-\frac{n\epsilon^2}{2} \right) \quad (2.2.9)$$

The proof is again in Appendix B.

From Lemma 2.2.1 we have

$$\mathbb{P}_{S \sim D} \left[\sup_{h \in \mathcal{H}} (R(h) - \hat{R}_S(h)) \geq 2\epsilon \right] \leq 2 \mathbb{P}_{S \sim D, S' \sim D} \left[\sup_{h \in \mathcal{H}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \epsilon \right] \quad (2.2.10)$$

Now, since on the right-hand side, both risk terms are empirical risks over samples of finite size, we can restate the supremum over infinite set of hypotheses as the maximum operation over the *restriction* of the hypothesis class to the set of data points. So, the right-hand side becomes

$$2 \mathbb{P}_{S, S' \sim D} \left[\sup_{h \in \mathcal{H}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \epsilon \right] = 2 \mathbb{P}_{S, S' \sim D} \left[\max_{h \in \mathcal{H}_{S, S'}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \epsilon \right], \quad (2.2.11)$$

where on the right, we have implicitly changed the meaning of h as a hypothesis to the projection of that hypothesis onto the subset $S \cup S'$ taken from the distribution D .

We can restate this expression as below

$$2 \mathbb{P}_{S, S' \sim D} \left[\max_{h \in \mathcal{H}_{S, S'}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \epsilon \right] = 2 \mathbb{P}_{S, S' \sim D} \left[\exists h \in \mathcal{H}_{S, S'} \mid (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \epsilon \right], \quad (2.2.12)$$

because the realization of each side implies the other side as well. We note that now the hypothesis function is chosen among the restriction of the infinite class to the finite set of $S \cup S'$ and therefore it has finite size. Hence, we can apply the union bound as we did in the

finite case.

The maximum possible value occurs when the restriction of all hypotheses in \mathcal{H} to the $2n$ -sized set $S \cup S'$, i.e., all members of the set $\mathcal{H}_{S,S'}$ reach the maximum possible probability $\mathbb{P}[\hat{R}_{S'}(h) - \hat{R}_S(h) \geq \epsilon]$. The number of these $2n$ -sized sets is at most equal to the growth function $\Pi_{\mathcal{H}}(2n)$. So the above probability is at most

$$2 \Pi_{\mathcal{H}}(2n) \max_{h \in \mathcal{H}} \left(\mathbb{P}_{S,S' \sim D} [\hat{R}_{S'}(h) - \hat{R}_S(h) \geq \epsilon] \right)$$

On the other hand, the probability term is upper-bounded according to Corollary 2.2.2, where we have taken it into account that L has Bernoulli distribution. Therefore, we get

$$\begin{aligned} 2 \mathbb{P}_{S,S' \sim D} \left[\max_{h \in \mathcal{H}_{S,S'}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \geq \epsilon \right] &\leq 2 \Pi_{\mathcal{H}}(2n) \max_{h \in \mathcal{H}} \left(\mathbb{P}_{S,S' \sim D} [\hat{R}_{S'}(h) - \hat{R}_S(h) \geq \epsilon] \right) \\ &\leq 2 \Pi_{\mathcal{H}}(2n) 2 \exp \left(-\frac{n\epsilon^2}{2} \right) \end{aligned} \quad (2.2.13)$$

Combining equations (2.2.10) and (2.2.13), we obtain the following relation

$$\mathbb{P}_{S \sim D} \left[\sup_{h \in \mathcal{H}} (R(h) - \hat{R}_S(h)) \geq 2\epsilon \right] \leq 4 \Pi_{\mathcal{H}}(2n) \exp \left(-\frac{n\epsilon^2}{2} \right) \quad (2.2.14)$$

As before, by taking this probability as δ , i.e., by putting $4 \Pi_{\mathcal{H}}(2n) \exp \left(-\frac{n\epsilon^2}{2} \right) = \delta$, we find the following generalization bound for any hypothesis h from the infinite class \mathcal{H} with probability at least $1 - \delta$ over the choice of a random sample

$$R(h) < \hat{R}_S(h) + 2 \sqrt{2 \frac{\log \Pi_{\mathcal{H}}(2n) + \log \frac{4}{\delta}}{n}}$$

So, Theorem 4 is proved. ■

VC-Dimension From Definition 3, the growth function $\Pi_{\mathcal{H}}(m)$ is an increasing function of its argument which is always upper-bounded by 2^m . The important point is that if for a given \mathcal{H} and m the equality holds, then the equality will hold for all smaller m 's as well. In other words, for a given \mathcal{H} and m if there exists a set of m points shattered by \mathcal{H} , then, for any $n < m$, there exists a set of n points shattered by \mathcal{H} as well. Now, we are at the right point to define the VC dimension of a class of hypotheses.

Definition 5. Let $\mathcal{H} \subset \{-1, +1\}^{\mathcal{X}}$ be a hypothesis class. The VC-dimension of \mathcal{H} , $d_{\text{VC}}(\mathcal{H})$, is the largest number of points x_1, \dots, x_n shattered by \mathcal{H} , i.e., for which $|\{(h(x_1), \dots, h(x_n)) \mid h \in \mathcal{H}\}| = 2^n$. In other words: $d_{\text{VC}}(\mathcal{H}) = \sup\{n \mid \Pi_{\mathcal{H}}(n) = 2^n\}$.

As an example of an infinite hypothesis class, consider the set of lines in 2-dimensional Euclidean space. We want to see how many two-dimensional points can be shattered by this class. Obviously, for any two (non-coinciding) points we can always find three distinct lines giving all four possible sign patterns. Figure 2.1a shows how three points can also be shattered by the set of lines in two dimensions. On the other hand, if we consider four

points in two dimensions, it is shown that for any set of four points in two dimensions, there always exists at least one sign pattern that is not linearly separable; that means there exists no line that can perfectly separate the two classes. Figure 2.1b illustrates a specific binary labelling of four points (more precisely, two binary labelling of four points, for two possible configurations of four points in *general position*) which is not linearly separable and therefore, prevents the set of four points from being shattered. It can be shown that in general, the VC-dimension of hyperplanes in d -dimensional space is equal to $d + 1$. The proof of this fact is given in Appendix C.

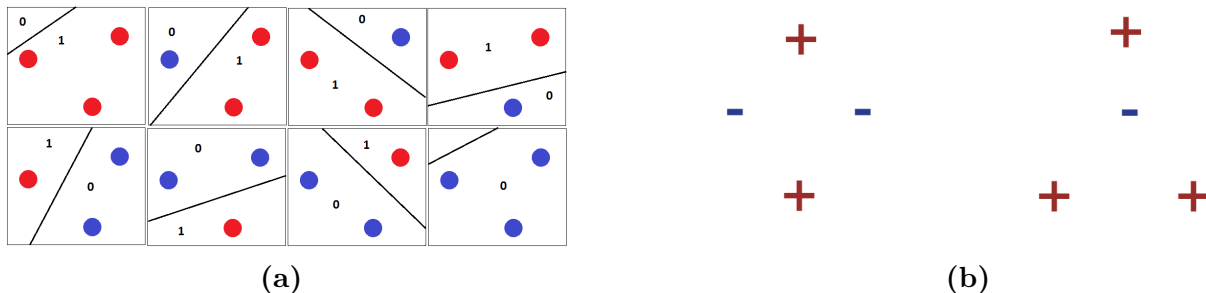


Figure 2.1. From [8]. (a) Illustration of shattering of three points in 2-dim by lines. (b) Two categories of four points in general position in two dimensions. The two sign patterns illustrated here are the ones that are not linearly separable. On the left, the four points lie on a convex hull. On the right one point lies inside the convex hull of the other three points.

From Definition 5 and our discussion above on the growth function, an important lemma follows:

Lemma 2.2.3. *If \mathcal{H} has VC-dimension d_{VC} , then for all $m \leq d_{\text{VC}}$, $\Pi_{\mathcal{H}}(m) = 2^m$. On the other hand, for all $m > d_{\text{VC}}$, $2^{d_{\text{VC}}} \leq \Pi_{\mathcal{H}}(m) < 2^m$.*

The upper bound in this lemma is in fact trivial. If we know the VC-dimension of a hypothesis class, there is a lemma, known as Sauer’s lemma, that tightens this bound to a more informative one.

Lemma 2.2.4. (*Sauer’s Lemma*) *Let \mathcal{H} be a hypothesis set of VC-dimension d_{VC} . Then for all $n \in \mathbb{N}$, the following relation holds*

$$\Pi_{\mathcal{H}}(n) \leq \sum_{i=0}^{d_{\text{VC}}} \binom{n}{i} \tag{2.2.15}$$

The proof of this lemma can be found in Appendix B.

Following Lemma 2.2.4, it can be shown that for $n > d_{\text{VC}}$, the growth function of a hypothesis class \mathcal{H} is bounded as $\Pi_{\mathcal{H}}(n) \leq \left(\frac{ne}{d_{\text{VC}}}\right)^{d_{\text{VC}}}$ [8]. By applying this upper-bound to Equation (2.2.7), we get the generalization bound in terms of the VC-dimension. Let \mathcal{H} be an infinite class of hypotheses with the VC-dimension d_{VC} . For any $\delta > 0$ with probability at least $1 - \delta$ over a random draw of a sample of finite size n , the generalization error (true

error) is bounded as below

$$\forall h \in \mathcal{H}, \quad R(h) < \hat{R}_S(h) + 2\sqrt{\frac{2}{n} \left(d_{\text{VC}} \log \frac{2ne}{d_{\text{VC}}} + \log \frac{4}{\delta} \right)} \quad (2.2.16)$$

2.3. Generalization Bounds for Regression

In this section, we go through the subject of generalization bound and complexity of the hypothesis class for the regression problem, i.e., for a supervised learning problem where the data points are labeled by real numbers and the loss function is accordingly a measure of the difference between the predicted label and the true label.

Let $S = \{(x_1, y_1) \dots, (x_n, y_n)\}$ be a sample drawn i.i.d. from an unknown distribution D . Based on this subset S , we want to find a function $h : \mathbb{R}^d \rightarrow \mathbb{R}$, such that for all $i \in [n]$, $y_i \approx h(x_i)$, with \approx denoting the approximation. The loss is usually the squared loss, defined as $L(y, \hat{y}) = (y - \hat{y})^2$ which gives the mean squared error as the empirical risk for the sample, i.e., $\hat{R} = \frac{1}{n} \sum_{i=1}^n (y_i - h(x_i))^2$. Similarly to the classification case, we consider both finite and infinite classes of hypotheses for regression. Also, we only consider bounded losses with $L \leq M \in \mathbb{R}$.

2.3.1. Finite Class of Hypotheses

For finite set of hypotheses, for each function $h \in \mathcal{H}$, from the Hoeffding's inequality in Corollary A.1.1 and Equation (A.1.19) in the proof of Theorem 19, we obtain

$$\mathbb{P}(\hat{R}(h) - R(h) \geq \epsilon) \leq \exp\left(-\frac{2n\epsilon^2}{M^2}\right) \quad (2.3.1)$$

Here, we have only used the fact that the generalization gap can be written as a sum over n centered loss terms, i.e., $L(y_i, h(x_i)) - \frac{R(h)}{n}$, which are all upper-bounded by M ; then, by applying the union bound, we get

$$\mathbb{P}(\exists h \in \mathcal{H} \mid R(h) - \hat{R}(h) \geq \epsilon) \leq |\mathcal{H}| \exp\left(-\frac{2n\epsilon^2}{M^2}\right) \quad (2.3.2)$$

As we saw in the classification case, this probability bound is equivalent to an upper bound on the generalization gap which is valid with probability at least $1 - \delta$ for an arbitrary $0 < \delta \leq 1$

$$R(h) - \hat{R}(h) \leq M \sqrt{\frac{\log |\mathcal{H}| - \log \delta}{2n}}. \quad (2.3.3)$$

2.3.2. Infinite Class of Hypotheses

The subject of generalization bounds for problems with real-valued loss functions is more involved than the classification case. As we saw in the previous section, VC-dimension is a

combinatorial measure of the complexity of the class of binary-valued functions. For real-valued functions that are pertinent to the regression problem, a similar quantity is defined which is called pseudo-dimension. Similar to the VC-dimension which is defined based on the notion of *shattering*, the pseudo-dimension is related to *pseudo-shattering*. The following definitions introduce these concepts.

Definition 6. For a real-valued hypothesis class $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$, we say that \mathcal{H} pseudo-shatters the points $x_1, \dots, x_n \in \mathcal{X}$ with thresholds $t_1, \dots, t_n \in \mathbb{R}$, if for every binary labeling of the points $(s_1, \dots, s_n) \in \{-1, +1\}^n$, there exists $h \in \mathcal{H}$ s.t. $h(x_i) < t_i$ if and only if $s_i = -1$.

Definition 7. The pseudo-dimension of a real-valued hypothesis class $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$, $\text{Pdim}(\mathcal{H})$, is the supremum over n for which there exist n points that are pseudo-shattered by \mathcal{H} (with some thresholds).

In other words, the class of functions \mathcal{H} pseudo-shatters the points $x_1, \dots, x_n \in \mathcal{X}$ with thresholds $t_1, \dots, t_n \in \mathbb{R}$ if for each of the 2^n subsets of the set of these n points, there exists a function $h \in \mathcal{H}$ which passes above those threshold values $t_i, i \in [n]$ which correspond to the points x_i belonging to that subset, and goes below other thresholds. Figure 2.2 illustrates the pseudo-shattering of two points in \mathbb{R} by the class of threshold functions. In this figure, we have chosen the points $p_1 = 1$ and $p_2 = 3$ and the thresholds $t_1 = 2$, $t_2 = 4$. The four linear functions that we choose to pseudo-shatter p_1 and p_2 with thresholds t_1 and t_2 , are: $f_1(x) = 2x - 1$, $f_2(x) = -x + 5$, $f_3(x) = 1$, $f_4(x) = 5$, which are shown by the green, blue, yellow and red lines respectively. It can be verified that three points in one dimension cannot be pseudo-shattered by linear functions. Therefore, the pseudo-dimension of the class of linear functions in one dimension is two. This result generalizes to higher dimensions and gives $d + 1$ as the pseudo-dimension of linear functions in \mathbb{R}^d , similar to the VC-dimension of the class of *linear* classifiers. More generally, it is shown that for any real-valued hypothesis class \mathcal{H} from \mathcal{X} to \mathbb{R} , the VC-dimension of the classifiers $\text{sign}(\mathcal{H}) = \{\text{sign}(h) \mid h \in \mathcal{H}\}$ is upper-bounded by the pseudo-dimension of \mathcal{H} [75]. This is because, from the definitions of shattering and pseudo-shattering, the pseudo-dimension is related to the VC-dimension by the relation

$$\text{Pdim}(\mathcal{H}) = d_{\text{VC}}(\{(x, t) \mapsto \text{sign}(h(x) - t) \mid h \in \mathcal{H}\})$$

which holds for any $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ [8].

In [8], it is shown how the generalization gap can be bounded in terms of the pseudo-dimension of a class of bounded *loss functions* associated to a hypothesis class of real-valued regression functions. In order to write this upper bound in terms of the pseudo-dimension of the corresponding *hypothesis class*, the loss function needs to satisfy some constraints, e.g., if the loss is a monotonic function of the hypothesis, then the pseudo-dimension of the class of loss functions $\{L(h(X), y) \mid h \in \mathcal{H}\}$ is equal to the pseudo-dimension of the hypothesis class \mathcal{H} [8]. These intricacies of the real-valued loss functions are beyond the scope of this

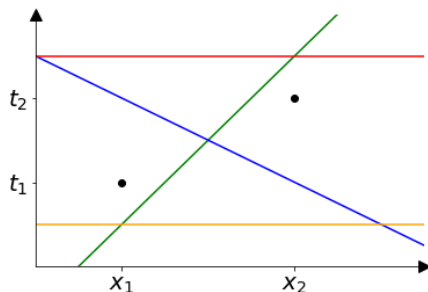


Figure 2.2. Pseudo-shattering of two points in one dimension, with thresholds t_1 and t_2 witnessing the pseudo-shattering.

thesis and therefore we do not go through the discussion of generalization bounds for the regression problem in terms of the pseudo-dimension. The main point of the current section is to emphasize on the pseudo-dimension as a combinatorial measure of complexity for real-valued hypothesis classes. Based on the above discussion, in the next chapter, we provide a unified approach for bounding both VC-dimension and pseudo-dimension for different supervised learning tasks, i.e., classification, regression and completion. In fact, pseudo-dimension can equivalently be defined for completion problems as we briefly see in the rest of this subsection.

Consider a completion problem; we are given a $m \times n$ matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, where we only observe a subset of S entries of the matrix with $S < mn$. Assuming that the matrix is of low rank $r < m, n$, we want to estimate the unobserved entries of the matrix. The binary version of this problem is when the matrix contains only plus and minus signs, and we want to find the correct sign for the missing entries. The relation of this problem to the classification problem is that this can be seen as a binary-classification problem with data points being the set of tuples containing the indices of the matrix entries. Therefore, the function class is a set of functions that maps from the space of matrix indices to $\{-1, +1\}$ [4]. Clearly, this problem extends to the tensor completion case with a similar interpretation in terms of a classification problem. Then, we can define a VC-dimension for the class of completion functions, similar to what we have in classification. In this case, the VC-dimension of the completion task gives the maximum number of entries of a $m \times n$ matrix of rank at most r to which any configuration of plus and minus signs can be assigned. Lemma 3.4.1 in the next chapter is a result of this definition and sheds more light on this point.

In a similar way, the pseudo-dimension is defined for matrix/tensor completion task when we are not only concerned with the sign of the entries, but also with their real values; in that case the completion function would map the indices to the real space and hence the pertinence of the pseudo-dimension.

Chapter 3

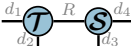
Generalization Bound and VC-dimension of Tensor Networks

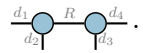
As already mentioned in the introduction chapter, the content of this chapter is the main contribution of the thesis and has been published in NeurIPS 2021 [70].

3.1. Introduction

We start this chapter by formally introducing tensor network learning models and showcasing some examples of such models. Then, we give a general upper bound on the VC-dimension and pseudo-dimension of hypothesis classes parameterized by *arbitrary* TN structures for linear regression, classification and completion. We then discuss corollaries of this general upper bound for common TN models including low-rank matrices and TT tensors, and compare them with existing results. Examples of particular upper bounds that can be derived from our general result can be found in Table 3.1. The last section of this chapter provides some lower bounds on several TN models. This is our first step towards tightening our general upper bound.

3.1.1. Tensor Network structures

In this section, we introduce a notation that simplifies defining the hypothesis class of tensor network models. In general, tensor networks can become too involved and contain too many vertices and edges. In such a case, the definition of the corresponding hypothesis class in a precise way can become too lengthy and unpractical. This notation resolves this problem and provides a way to concretely define the hypothesis class of any tensor network model in a neat way. To introduce our notation, we note that a tensor network (TN) can be fundamentally decomposed in two constituent parts: a tensor network structure, which describes its graphical structure, and a set of core tensors assigned to each node. For example, the tensor in $\mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$ represented by  is obtained by assigning the core tensors

$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times R}$ and $\mathcal{S} \in \mathbb{R}^{R \times d_3 \times d_4}$ to the nodes of the TN structure . This decomposition is illustrated in Figure 3.1.

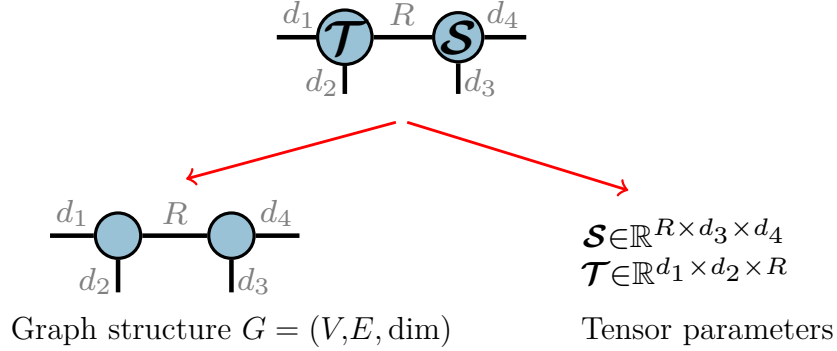


Figure 3.1. Disentangling the graph structure of a tensor network from its parameters

Formally, a *tensor network structure* is given by a graph $G = (V, E, \dim)$ where edges are labeled by integers: V is the set of vertices, $E \subset V \cup (V \times V)$ is a set of edges containing both classical edges ($e \in V \times V$) and singleton edges ($e \in V$) and $\dim : E \rightarrow \mathbb{N}$ assigns a dimension to each edge in the graph. The set of singleton edges $\delta_G = E \cap V$ corresponds to the dangling legs of a TN; it follows that δ_G has as many members as the order of the tensor. Given a TN structure G , one obtains a tensor by assigning a core tensor $\mathcal{T}^v \in \bigotimes_{e \in E_v} \mathbb{R}^{\dim(e)}$ to each vertex v in the graph, where $E_v = \{e \in E \mid v \in e\}$. The resulting tensor, denoted by $TN(G, \{\mathcal{T}^v\}_{v \in V})$, is a tensor of order $|\delta_G|$ in the tensor product space $\bigotimes_{e \in \delta_G} \mathbb{R}^{\dim(e)}$. Given a tensor structure $G = (V, E, \dim)$, the set of all tensors that can be obtained by assigning core tensors to the vertices of G is denoted by $\mathcal{T}(G) \subset \bigotimes_{e \in \delta_G} \mathbb{R}^{\dim(e)}$:

$$\mathcal{T}(G) = \{TN(G, \{\mathcal{T}^v\}_{v \in V}) : \mathcal{T}^v \in \bigotimes_{e \in E_v} \mathbb{R}^{\dim(e)}, v \in V\}. \quad (3.1.1)$$

As an illustration, one can check that the set of $m \times n$ matrices of rank at most r is equal to $\mathcal{T}(\text{---} \overset{m}{\circ} \text{---} \overset{r}{\circ} \text{---} \overset{n}{\circ} \text{---})$. Similarly, the set of all 4th order d -dimensional tensors of TT rank at most r is equal to $\mathcal{T}(\text{---} \overset{r}{\underset{d}{|}} \text{---} \overset{r}{\underset{d}{|}} \text{---} \overset{r}{\underset{d}{|}} \text{---} \overset{r}{\underset{d}{|}} \text{---})$. Finally, for a given graph structure G , the number of parameters of any member of the family $\mathcal{T}(G)$ in Equation (3.1.1) (which is the total number of entries of the core tensors $\{\mathcal{T}^v\}_{v \in V}$) is given by

$$N_G = \sum_{v \in V} \prod_{e \in E_v} \dim(e) \quad (3.1.2)$$

This will be a central quantity in the generalization bounds and bounds on the VC-dimension of TN models that we derive in subsequent sections. Graph structure of some common tensor networks are illustrated in Figure 3.2.

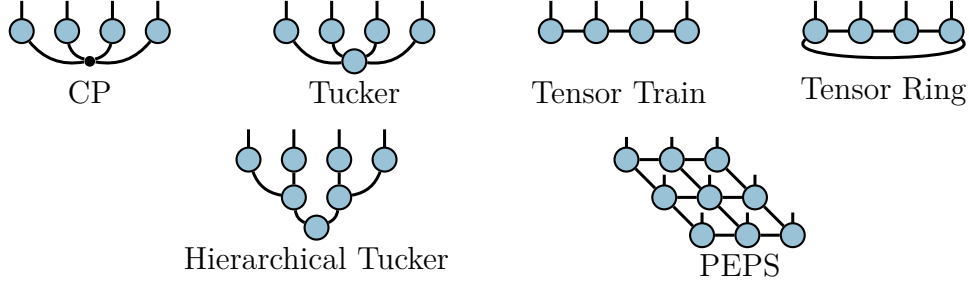


Figure 3.2. Graph structures of TN representation of common decomposition models for 4th order and 9th order tensors. For CP, the black dot represents a hyperedge corresponding to a joint contraction over 4 indices. For the ease of representation, the edge weights, i.e., the dimensions of the core tensors are not shown.

3.2. Tensor Network Learning Models

In this section, we formalize the general notion of *tensor network models* for supervised learning tasks. We then show how it encompasses classical models such as low-rank matrix completion [76, 77, 78, 79], classification [50, 51, 52], and tensor-train-based models [1, 27, 61, 62, 63, 64]. Consider a classification problem where the input space \mathcal{X} is the space of p -th order tensors $\mathbb{R}^{d_1 \times d_2 \times \dots \times d_p}$. One motivation for TN models is that the tensor product space \mathcal{X} can be exponentially large, thus learning a linear model in this space is often not feasible. Indeed, the number of parameters of a linear classifier $h : \mathcal{X} \mapsto \text{sign}(\langle \mathcal{X}, \mathcal{W} \rangle)$, where $\mathcal{W} \in \mathbb{R}^{d_1 \times \dots \times d_p}$ is the tensor weight, grows exponentially with p . TN models parameterize \mathcal{W} as a low-rank TN, thus reducing the number of parameters needed to represent a model h . Our objective is to derive generalization bounds for the class of such hypotheses parameterized by low-rank tensor networks for classification, regression and completion tasks.

Formally, let $G = (V, E, \text{dim})$ be a TN structure for tensors of shape $d_1 \times \dots \times d_p$, i.e. where the set of singleton edges is $\delta_G = E \cap V = \{v_1, \dots, v_p\}$ and $\text{dim}(v_i) = d_i$ for each $i \in [p]$. We are interested in the class of models whose weight tensors are represented in the TN structure G :

$$\mathcal{H}_G^{\text{regression}} = \{h : \mathcal{X} \mapsto \langle \mathcal{W}, \mathcal{X} \rangle \mid \mathcal{W} \in \mathcal{T}(G)\} \quad (3.2.1)$$

$$\mathcal{H}_G^{\text{classif}} = \{h : \mathcal{X} \mapsto \text{sign}(\langle \mathcal{W}, \mathcal{X} \rangle) \mid \mathcal{W} \in \mathcal{T}(G)\} \quad (3.2.2)$$

$$\mathcal{H}_G^{\text{completion}} = \{h : (i_1, \dots, i_p) \mapsto \mathcal{W}_{i_1, \dots, i_p} \mid \mathcal{W} \in \mathcal{T}(G)\} \quad (3.2.3)$$

Notice how by disentangling the graph structure of tensor networks from their entries, the notation that we introduced in the earlier section allows us to define these hypothesis classes in a neat and concrete way for any arbitrary TN structure G . In Equation (3.2.3) for the completion hypothesis class, p -th order tensors are interpreted as real-valued functions $f : [d_1] \times \dots \times [d_p] \rightarrow \mathbb{R}$ over the indices of the tensor. $\mathcal{H}_G^{\text{completion}}$ is thus a class of functions

over the indices domain, for which the notion of pseudo-dimension is well-defined. This treatment of completion as a supervised learning task was considered previously to derive generalization bounds for matrix and tensor completion [4, 48].

As mentioned before, the benefit of TN models comes from the drastic reduction in parameters when the TN structure G is *low-rank*, in the sense that the number of parameters N_G is small compared to $d_1 d_2 \cdots d_p$. In addition to allowing one to represent linear models in exponentially large spaces, this compression controls the capacity of the corresponding hypothesis class \mathcal{H}_G .

3.2.1. Examples

To illustrate some TN models, we now present several examples of models based on common TN structures: low-rank matrices and tensor trains.

Low-rank matrices As discussed in Section 3.1.1, if we define the TN structure $G_{\text{mat}}(r) = \text{---} \textcircled{d_1} \textcircled{r} \textcircled{d_2} \text{---}$, then $\mathcal{T}(G_{\text{mat}}(r))$ is the set of matrices in $\mathbb{R}^{d_1 \times d_2}$ of rank at most r . The hypothesis class $\mathcal{H}_{G_{\text{mat}}(r)}^{\text{completion}}$ then corresponds to the classical problem of low-rank matrix completion [76, 77, 78, 79]. Similarly $\mathcal{H}_{G_{\text{mat}}(r)}^{\text{classif}}$ corresponds to the hypothesis class of low-rank matrix classifiers. This hypothesis class was previously considered, notably to compactly represent the parameters of support vector machines for matrix inputs [50, 51, 52]. Lastly, for the regression case, $\mathcal{H}_{G_{\text{mat}}(r)}^{\text{regression}}$ is the set of functions $\{h : \mathbf{X} \mapsto \text{Tr}(\mathbf{W}\mathbf{X}^\top) \mid \text{rank}(\mathbf{W}) \leq r\}$. Learning hypotheses from this class is relevant in, e.g., quantum tomography, where it is known as the *low-rank trace regression* problem [80, 81, 82, 83].

Tensor train tensors As explained in Chapter 1, the tensor train (TT) decomposition model [12], has a number of parameters that grows only linearly with the order of the tensor and this makes the TT format an appealing model for compressing the parameters of ML models [1, 26, 35, 18]. Let us remind the tensor train classifier model that we reviewed in Subsection 1.5, which was introduced in [1] and subsequently explored in [27]. Given a vector input $\mathbf{x} \in \mathbb{R}^p$, Stoudenmire and Schwab [1] propose to map \mathbf{x} into a high-dimensional space of p -th order tensors $\mathcal{X} = \mathbb{R}^{d \times \cdots \times d}$ by applying a local feature map $\phi : \mathbb{R} \rightarrow \mathbb{R}^d$ to each component of the vector \mathbf{x} and taking their outer product: $\Phi(\mathbf{x}) = \phi(\mathbf{x}_1) \otimes \phi(\mathbf{x}_2) \otimes \cdots \otimes \phi(\mathbf{x}_p) \in (\mathbb{R}^d)^{\otimes p}$.

Instead of relying on the so-called kernel trick, Stoudenmire and Schwab propose to directly learn the parameters \mathcal{W} of a linear model $h : \mathbf{x} \mapsto \text{sign}(\langle \mathcal{W}, \Phi(\mathbf{x}) \rangle)$ in the exponentially large feature space \mathcal{X} . The learning problem is made tractable by parameterizing \mathcal{W} as a low-rank TT tensor (see Equation (1.4.15)). Letting $G_{\text{TT}}(r_1, \dots, r_{p-1}) = \text{---} \textcircled{d_1} \textcircled{r_1} \textcircled{r_2} \cdots \textcircled{r_{p-2}} \textcircled{r_{p-1}} \textcircled{d_p} \text{---}$,

the hypothesis class considered in [1] is $\mathcal{H}_{G_{\text{TT}}(r_1, \dots, r_{p-1})}^{\text{classif}}$. In addition to the approach of [1], which was extended in [27] and [61], tensor train classifiers were also previously considered in [62, 63, 64]. Similarly, the hypothesis class $\mathcal{H}_{G_{\text{TT}}(r_1, \dots, r_{p-1})}^{\text{completion}}$ corresponds to the low-rank TT completion problem [84, 19, 85].

Other TN models Lastly, we mention that our formalism can be applied to any tensor models having a low-rank structure, including CP, Tucker, tensor ring and PEPS. As mentioned previously, for the case of the CP decomposition, the graph G of the TN structure is in fact a hyper-graph with $|V| = p$ nodes and $N_G = pdr$ parameters for a weight tensor in $(\mathbb{R}^d)^{\otimes p}$ with CP rank at most r . Several TN learning models using these decomposition models have been proposed previously, including [58, 59] for regression in the Tucker format, [65] for classification using the PEPS model, [54, 55] for classification with the CP decomposition and [20, 86] for tensor completion with TR.

3.3. Bounds on the VC/Pseudo-dimension and the Generalization Gap

The following theorem states one of our main results which upper bounds the VC and pseudo-dimension of models parameterized by arbitrary TN structures.

Theorem 8. *Let $G = (V, E, \text{dim})$ be a tensor network structure and let $\mathcal{H}_G^{\text{regression}}$, $\mathcal{H}_G^{\text{classif}}$, $\mathcal{H}_G^{\text{completion}}$ be the corresponding hypothesis classes defined in Equations (3.2.1-3.2.3), where each model has N_G parameters (see Equation (3.1.2)). Then, $\text{Pdim}(\mathcal{H}_G^{\text{regression}})$, $d_{\text{VC}}(\mathcal{H}_G^{\text{classif}})$ and $\text{Pdim}(\mathcal{H}_G^{\text{completion}})$ are all upper-bounded by $2N_G \log(12|V|)$.*

These bounds naturally relate the capacity of the TN classes $\mathcal{H}_G^{\text{regression}}$, $\mathcal{H}_G^{\text{classif}}$, $\mathcal{H}_G^{\text{completion}}$ to the number of parameters N_G of the underlying TN structure G .

PROOF. Following the analysis of [4] for matrix completion and its extension to the Tucker decomposition model presented in [48], the proof of this theorem leverages Warren’s theorem which bounds the number of sign patterns a system of polynomial equations can take.

Theorem 9. [46] *The number of sign patterns of n real polynomials, each of degree at most v , over N variables is at most $\left(\frac{4evn}{N}\right)^N$ for all $n > N > 2$ (where e is Euler’s number).*

We start with the pseudo-dimension introduced in Definition 7. Consider n input tensors $\mathcal{X}_1, \dots, \mathcal{X}_n$ and arbitrary threshold values t_1, \dots, t_n . To upper-bound $\text{Pdim}(\mathcal{H}_G^{\text{regression}})$, it is enough to show that for any set $S = \{\mathcal{X}_1, \dots, \mathcal{X}_n\}$ and threshold values t_1, \dots, t_n , the number of *relative sign patterns* realized by the class of functions $\mathcal{H}_G^{\text{regression}}$ is bounded by a value depending only on n and the tensor network structure G . Formally, we define the

maximal number of sign patterns as follows:

$$f(n, G) := \sup_{\substack{\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X} \\ t_1, \dots, t_n \in \mathbb{R}}} \left| \left\{ \begin{pmatrix} \text{sign}(h(\mathbf{x}_1) - t_1) \\ \vdots \\ \text{sign}(h(\mathbf{x}_n) - t_n) \end{pmatrix} \mid h \in \mathcal{H}_G^{\text{regression}} \right\} \right| \quad (3.3.1)$$

Let $G = (V, E)$ be an arbitrary TN structure. For $h \in \mathcal{H}_G^{\text{regression}}$, by definition, $h : \mathcal{X} \mapsto \langle \mathcal{W}, \mathcal{X} \rangle$ for some weight tensor $\mathcal{W} \in \mathcal{T}(G)$. Consequently, there exists a collection of core tensors $\mathcal{T}^v \in \otimes_{e \in E_v} \mathbb{R}^{\dim(e)}$ such that $\mathcal{W} = TN(G, \{\mathcal{T}^v\}_{v \in V})$ (see Equation (3.1.1)) and it follows that $h(\mathcal{X})$ is a polynomial of degree $|V|$ over N_G variables. The variables of the polynomial are the entries of the core tensors $\{\mathcal{T}^v\}_{v \in V}$.

Now, given a set of input tensors $S = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the value $f(n, G)$ in Equation (3.3.1) is thus bounded by the number of sign patterns that a system of n polynomial equations (one for each input data point) of order $|V|$ over N_G variables can take. It then follows from Warren's theorem (Theorem 9) that

$$f(n, G) \leq \left(\frac{4en|V|}{N_G} \right)^{N_G}. \quad (3.3.2)$$

Bound on the pseudo-dimension To extract a bound on the pseudo-dimension from the above bound on the number of *relative sign patterns*, we follow the line of the proof of Theorem 8.3 in [47]. First observe that by the definition of the pseudo-dimension, if $f(n, N_G) < 2^n$ for some n , then $\text{Pdim}(\mathcal{H}_G^{\text{regression}}) < n$. Using the bound on $f(n, N_G)$, we have $f(n, N_G) \leq \left(\frac{4en|V|}{N_G} \right)^{N_G} < 2^n$ if and only if

$$N_G \left(\log n + \log \frac{4e|V|}{N_G} \right) < n. \quad (3.3.3)$$

Using the classical inequality $\ln n \leq nb + \ln \frac{1}{b} - 1$, or equivalently $\log n \leq \frac{nb}{\ln 2} + \log \frac{1}{eb}$, it follows that

$$\log n \leq \frac{n}{2N_G} + \log \frac{2N_G}{e \ln 2}.$$

Consequently, Equation (3.3.3) is implied by $n > 2N_G \log \frac{8|V|}{\ln 2}$, which is in turn implied by $n > 2N_G \log(12|V|)$.

We thus have shown that $\text{Pdim}(\mathcal{H}_G^{\text{regression}}) \leq 2N_G \log(12|V|)$. Since for any hypothesis class \mathcal{H} , $\text{Pdim}(\mathcal{H}) = d_{\text{VC}}(\{(x, t) \mapsto \text{sign}(h(x) - t) \mid h \in \mathcal{H}\})$ this upper bound implies that there exists no set of $k \geq 2N_G \log(12|V|)$ points that are shattered by the hypothesis class

$$\{(\mathcal{X}, t) \mapsto \text{sign}(h(\mathcal{X}) - t) \mid h \in \mathcal{H}_G^{\text{regression}}\} = \{(\mathcal{X}, t) \mapsto \text{sign}(\langle \mathcal{W}, \mathcal{X} \rangle - t) \mid \mathcal{W} \in \mathcal{T}(G)\}.$$

In particular, no set of k points with thresholds $t_1 = \dots = t_k = 0$ is shattered by $\mathcal{H}_G^{\text{regression}}$, which is equivalent to no set of k points being shattered by $\mathcal{H}_G^{\text{classif}}$, hence $d_{\text{VC}}(\mathcal{H}_G^{\text{classif}}) \leq$

$2N_G \log(12|V|)$.

Similarly, for the completion case we argue that the maximum number of multiples of indices shattered by the function class $\mathcal{H}_G^{\text{completion}}$ is bounded by the same value as $\text{Pdim}(\mathcal{H}_G^{\text{regression}})$. The *Pseudo-dimension* of $\mathcal{H}_G^{\text{completion}}$ is by definition, the maximum number of indices, i.e., the maximum number of the entries of the tensor, that could be pseudo-shattered (with thresholds zero) by the class of tensors $\mathcal{H}_G^{\text{completion}}$. Each component of the tensor $\mathcal{T}_{i_1, \dots, i_p}$ can be written as the following inner product

$$\mathcal{T}_{i_1, \dots, i_p} = \langle \mathcal{T}, \mathbf{e}_{i_1}^{(1)} \otimes \mathbf{e}_{i_2}^{(2)} \otimes \dots \otimes \mathbf{e}_{i_p}^{(p)} \rangle$$

where each $\mathbf{e}_i^{(j)} \in \mathbb{R}^{d_j}$ is the i -th vector of the canonical basis of \mathbb{R}^{d_j} . Thus, no set of more than $2N_G \log(12|V|)$ indices is shattered by $\mathcal{H}_G^{\text{completion}}$, since otherwise the corresponding set of points $\mathbf{e}_{i_1}^{(1)} \otimes \dots \otimes \mathbf{e}_{i_p}^{(p)}$ would be shattered by $\mathcal{H}_G^{\text{regression}}$. Therefore, $\text{Pdim}(\mathcal{H}_G^{\text{completion}}) \leq 2N_G \log(12|V|)$. \square

The bounds on the VC-dimension and pseudo-dimension presented in Theorem 8 can be leveraged to derive bounds on the generalization error of the corresponding learning models; see for example [8]. In the following theorem, we derive such a generalization bound for classifiers parameterized by arbitrary TN structures.

Theorem 10. *Let S be a sample of size n drawn from a distribution D and let ℓ be a loss bounded by 1, including the 0 – 1 loss. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of S , for any $h \in \mathcal{H}_G^{\text{classif}}$,*

$$R(h) < \hat{R}_S(h) + 2\sqrt{\frac{2}{n} \left(N_G \log \frac{8en|V|}{N_G} + \log \frac{4}{\delta} \right)}. \quad (3.3.4)$$

The proof takes into account the general formula in Theorem 4 as well as the bound on the growth function which follows from Theorem 9, i.e., $f(n, G) \leq \left(\frac{4en|V|}{N_G} \right)^{N_G}$ as in Equation (3.3.2). It follows from this theorem that, with high probability, the generalization gap $R(h) - \hat{R}_S(h)$ of any hypothesis $h \in \mathcal{H}_G^{\text{classif}}$ is in $\mathcal{O} \left(\sqrt{\frac{N_G \log(n)}{n}} \right)$. This bound naturally relates the sample complexity of the hypothesis class with its expressiveness. The notion of richness of the hypothesis class appearing in this bound reflects the structure of the underlying TN through the number of parameters N_G . Using classical results (see, e.g., Theorem 10.6 in [8]), similar generalization bounds for regression and classification with arbitrary TN structures can be obtained from the bounds on the pseudo-dimension of $\mathcal{H}_G^{\text{regression}}$ and $\mathcal{H}_G^{\text{completion}}$ derived in Theorem 8. As detailed in Subsection 3.3.2, to examine this upper bound in practice, we perform an experiment with low-rank TT classifiers on synthetic data. In the next subsection, we present corollaries of our results for particular TN structures, including low-rank matrix completion and the TT classifiers introduced in [1].

3.3.1. Special cases

We now discuss special cases of Theorems 8 and 10 and compare them with existing results.

Low-rank matrices Let $G_{\text{mat}}(r) = \begin{array}{c} d_1 \text{---} \text{---} r \text{---} \text{---} d_2 \\ \bullet \quad \bullet \end{array}$ and $\mathcal{T}(G_{\text{mat}}(r))$ be the set of $d_1 \times d_2$ matrices of rank at most r . In this case, we have $|V| = 2$ and $N_{G_{\text{mat}}(r)} = r(d_1 + d_2)$, and Theorems 8 and 10 give the following result.

Corollary 3.3.1. *Pdim($\mathcal{H}_{G_{\text{mat}}(r)}^{\text{regression}}$), $d_{\text{VC}}(\mathcal{H}_{G_{\text{mat}}(r)}^{\text{classif}})$ and Pdim($\mathcal{H}_{G_{\text{mat}}(r)}^{\text{completion}}$) are all upper-bounded by $10r(d_1 + d_2)$. Moreover, with high probability over the choice of a sample S of size n drawn i.i.d. from a distribution D , the generalization gap $R(h) - \hat{R}_S(h)$ of any hypothesis $h \in \mathcal{H}_{G_{\text{mat}}(r)}^{\text{classif}}$ is in $\mathcal{O}\left(\sqrt{\frac{r(d_1+d_2)\log(n)}{n}}\right)$.*

This bound improves on the one given in [52] where the VC-dimension of $\mathcal{H}_{G_{\text{mat}}(r)}^{\text{classif}}$ is bounded by $r(d_1 + d_2) \log(r(d_1 + d_2))$ (see Theorem 2 in [52]). For the matrix completion case, our upper bound improves on the bound $\text{Pdim}(\mathcal{H}_{G_{\text{mat}}(r)}^{\text{completion}}) \leq r(d_1 + d_2) \log \frac{16ed_1}{r}$ derived in [4]; recall that this improvement is the result of extracting our VC-dimension upper bound from the upper bound on the number of sign patterns using a more sophisticated approach than the one in [4] (Our approach is demonstrated in the proof of Theorem 8). In Section 3.4, we will derive lower bounds showing that the upper bounds on the VC/pseudo-dimension of Corollary 3.3.1 are tight up to the constant factor 10 for matrix completion, regression and classification.

Tensor train Let $G_{\text{TT}}(r) = \begin{array}{c} d_1 \text{---} \text{---} r \text{---} \text{---} d_2 \text{---} \dots \text{---} r \text{---} \text{---} d_{p-1} \text{---} \text{---} d_p \\ \bullet \quad \bullet \quad \quad \quad \bullet \quad \bullet \end{array}$ and $\mathcal{T}(G_{\text{TT}}(r))$ be the set of tensors of TT rank at most r . In this case, we have $|V| = p$ and $N_G = \mathcal{O}(dpr^2)$ where $d = \max_i d_i$. For this class of hypotheses, Theorems 8 and 10 give the following result.

Corollary 3.3.2. *Pdim($\mathcal{H}_{G_{\text{TT}}(r)}^{\text{regression}}$), $d_{\text{VC}}(\mathcal{H}_{G_{\text{TT}}(r)}^{\text{classif}})$ and Pdim($\mathcal{H}_{G_{\text{TT}}(r)}^{\text{completion}}$) are all in $\mathcal{O}(dpr^2 \log(p))$, where $d = \max_i d_i$. Moreover, with high probability over the choice of a sample S of size n drawn i.i.d. from a distribution D , the generalization gap $R(h) - \hat{R}_S(h)$ of any hypothesis $h \in \mathcal{H}_{G_{\text{TT}}(r)}^{\text{classif}}$ is in $\mathcal{O}\left(\sqrt{\frac{dpr^2 \log(n)}{n}}\right)$.*

This result applies for the MPS model introduced in [1] and thus answers the open problem listed as Question 13 in [2]. To the best of our knowledge, the VC-dimension of tensor train classifier models has not been studied previously and our work is the first to address this open question. The lower bounds we derive in Section 3.4 show that the upper bounds on the VC/pseudo-dimension of Corollary 3.3.2 are tight up to a $\mathcal{O}(\log(p))$ factor.

Tucker We briefly compare our result with the ones proved in [48] for tensor completion and in [59] for tensor regression using the Tucker decomposition. For a Tucker

decomposition with maximum rank r for tensors of size $d_1 \times \dots \times d_p$ with maximal dimension $d = \max_i d_i$, the number of parameters is in $\mathcal{O}(r^p + dpr)$ and the number of vertices in the TN structure is $p + 1$. In this case, Theorems 8 and 10 show that the VC/pseudo-dimensions are in $\mathcal{O}((r^p + dpr) \log(p))$ and the generalization gap is in $\mathcal{O}\left(\sqrt{\frac{(r^p + dpr) \log(n)}{n}}\right)$ with high probability for any classifier parameterized by a low-rank Tucker tensor. It is worth observing that in contrast with the tensor train decomposition, all bounds have an exponential dependency on the tensor order p . In [48], the authors give an upper bound on the analogue of the growth function for tensor completion problems which is equivalent to ours. In [59], the pseudo-dimension of regression functions whose weight parameters have low Tucker rank is upper-bounded by $\mathcal{O}((r^p + dpr) \log(pd^{p-1}))$, which is looser than our bound due to the term d^{p-1} (though a similar argument to the one we use in the proof of Theorem 10 can be used to tighten the bound given in [59]).

Tree tensor networks Lastly, we compare our result with the ones presented in [69] where the authors study the complexity of learning with tree tensor networks using metric entropy and covering numbers. The results presented in [69] only hold for TN structures whose underlying graph G is a tree. Let G be a tree and ℓ be a loss function which is both bounded and Lipschitz. Under these assumptions, it is shown in [69] that, for any $h \in \mathcal{H}_G^{\text{regression}}$, with high probability over the choice of a sample S of size n drawn i.i.d. from a distribution D , the generalization gap $R(h) - \hat{R}(h)$ is in $\tilde{\mathcal{O}}(\sqrt{N_G}/n)$. Theorem 10 gives a similar upper bound in $\tilde{\mathcal{O}}(\sqrt{N_G}/n)$ on the generalization gap of low-rank tensor classifiers. However, our results hold for *any* TN structure G . Thus, in contrast with our general upper bound (Theorem 8), the bounds from [69] cannot be applied to TN structures containing cycles such as tensor ring and PEPS.

3.3.2. Experiments

To evaluate the theoretical upper bound provided in Theorem 10, we perform a simple binary classification experiment with synthetic data. We draw a random low-rank TT target tensor $\mathcal{W} \in \mathbb{R}^{4 \times 4 \times 4 \times 4}$ of rank 8 by drawing the components of the cores of the TT decomposition i.i.d. from a uniform distribution between -1 and 1. Input-output data is generated with $y_i = \text{sign}(\langle \mathcal{W}, \mathcal{X}_i \rangle)$ for training and testing, where the components of \mathcal{X}_i are drawn i.i.d. from a normal distribution. Using the cross-entropy as loss function, we optimize the empirical risk using stochastic gradient descent with a learning rate of 10^{-2} to learn a TT hypothesis of rank r .

In Figure 3.3, we report the generalization gap of the learned hypothesis h , $R(h) - \hat{R}_S(h)$, where the true risk $R(h)$ is estimated on a test set of size 4,000 for different scenarios. In Figure 3.3 (left), we show how the sample size affects the generalization gap for learned

hypothesis of rank $r = 2$ and $r = 4$. As expected, the generalization gap decreases as the sample size grows, and is smaller for $r = 2$ than $r = 4$ which is also expected from Theorem 10. In Figure 3.3 (right), we show how the rank r of the learned hypothesis affects the generalization for sample sizes 2,000 and 4,000. As expected, the higher the rank of the TT weight tensor, the larger the model complexity and hence the generalization gap. In both figures, we observe that the theoretical upper bound and the experimental results follow a similar trend as a function of the sample size and hypothesis rank.

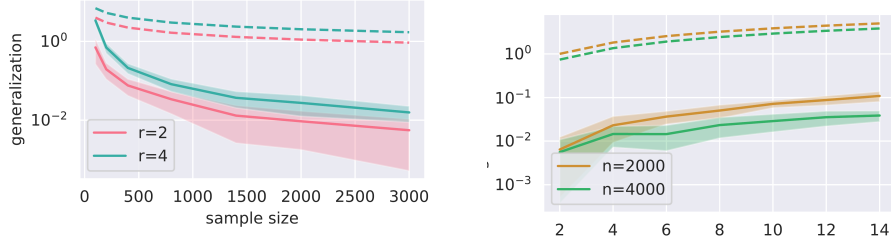


Figure 3.3. Dashed lines represent the theoretical bound, full lines represent the generalization gap (averaged over 20 runs for both experiments), and shaded areas show the standard deviation. (left) Generalization error for two models with ranks $r = 2$ and $r = 4$ as a function of training size. (right) Generalization error for two sample sizes $n = 2000$ and $n = 4000$ as a function of the rank of the learned hypothesis.

3.4. Lower Bounds

We now present lower bounds on the VC and pseudo-dimensions of standard TN models: rank-one, CP, Tucker, TT and TR.

	rank one	CP	Tucker	TT / TR
Decomposition				
Lower Bound (condition)	$(d - 1)p$	rd ($r \leq d^{p-1}$)	r^p ($r \leq d$)	r^2d ($r \leq d^{\lfloor \frac{p-1}{2} \rfloor}, p \geq 3$) $\frac{p(r^2d-1)}{3}$ ($r = d, \frac{p}{3} \in \mathbb{N}$)
Upper bound	$2dp \log(12p)$	$2prd \log(12p)$	$2(r^p + prd) \log(24p)$	$2pr^2d \log(12p)$

Table 3.1. Summary of our results for common TN structures. Both lower and upper bounds hold for the VC/pseudo-dimension of $\mathcal{H}_G^{\text{classif}}$, $\mathcal{H}_G^{\text{completion}}$ and $\mathcal{H}_G^{\text{regression}}$ for the corresponding TN structure G (see Equations (3.2.1-3.2.3)). The upper bounds follow from applying our general upper bound (Theorem 8) to each TN structure. The lower bounds are proved for each TN structure specifically. Each lower bound is followed by the condition under which it holds in parenthesis (small font). Note that the two bounds for TT and TR hold for both TN structures.

Theorem 11. *The VC-dimension and pseudo-dimension of the classification, regression and completion hypothesis classes defined in Equations (3.2.1-3.2.3) for the rank-one, CP,*

Tucker, TT and TR tensor network structures satisfy the lower bounds presented in Table 3.1. These lower bounds show that the general upper bound of Theorem 8 is tight up to a $\mathcal{O}(\log(p))$ factor for rank-one, TT and TR tensors and is tight up to a constant for low-rank matrices.

We devote the next section to the proof of this theorem. These lower bounds show that our general upper bound is nearly optimal (up to a log factor in p) for rank-one, TT and TR tensors. Indeed, for rank-one tensors we have $(d-1)p \leq \mathcal{C}^{\text{rank-one}} \leq 2dp \log(12p)$ and for TT and TR tensors of rank $r = d$ whose order p is a multiple of 3 we have $p(r^2d-1)/3 \leq \mathcal{C}_r^{\text{TT/TR}} \leq pr^2d \cdot 2 \log(12p)$, where $\mathcal{C}^{\text{rank-one}}$ (resp. $\mathcal{C}_r^{\text{TT/TR}}$) denotes any of the VC/pseudo-dimension of the regression, classification and completion hypothesis classes associated with rank-one tensors (resp. rank r TT and TR tensors). In addition, the lower bound for the CP case shows that our general upper bounds are tight up to a constant for matrices. Indeed, for $p = 2$ and $r \leq d$ the bounds for the CP case give $rd \leq \mathcal{C}_r^{\text{matrix}} \leq 20rd$ where $\mathcal{C}_r^{\text{matrix}}$ denotes the VC/pseudo-dimension of the hypothesis classes associated with $d \times d$ matrices of rank at most r .

3.4.1. Proof of Theorem 11

In the remaining sections, we give the proofs of all the lower bounds appearing in Table 3.1. All proofs rely on the following lemma which gives a useful way for jointly deriving lower bounds on the pseudo-dimension and VC-dimension of the hypothesis classes of linear models for regression, completion and classification defined in Equations (3.2.1-3.2.3).

Lemma 3.4.1. *Let $V \subset \mathbb{R}^d$ and define the hypothesis classes*

$$\begin{aligned} \mathcal{H}^{\text{completion}} &= \{h : i \mapsto \mathbf{w}_i \mid \mathbf{w} \in V\} \\ \mathcal{H}^{\text{regression}} &= \{h : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle \mid \mathbf{w} \in V\} \\ \mathcal{H}^{\text{classif}} &= \{h : \mathbf{x} \mapsto \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) \mid \mathbf{w} \in V\} . \end{aligned}$$

If there exist k indices $i_1, \dots, i_k \in [d]$ that are shattered by V , i.e., such that

$$|\{(\text{sign}(\mathbf{w}_{i_1}), \text{sign}(\mathbf{w}_{i_2}), \dots, \text{sign}(\mathbf{w}_{i_k})) \mid \mathbf{w} \in V\}| = 2^k ,$$

then $d_{\text{VC}}(\mathcal{H}^{\text{classif}})$, $\text{Pdim}(\mathcal{H}^{\text{regression}})$ and $\text{Pdim}(\mathcal{H}^{\text{completion}})$ are all lower bounded by k .

PROOF. Let $\mathbf{e}_1, \dots, \mathbf{e}_d$ be the canonical basis of \mathbb{R}^d and let $i_1, \dots, i_k \in [d]$ be a set of indices shattered by V . Since $\langle \mathbf{w}, \mathbf{e}_i \rangle = \mathbf{w}_i$ for all $i \in [d]$, the points $\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}$ are shattered by $\mathcal{H}^{\text{classif}}$ and thus $d_{\text{VC}}(\mathcal{H}^{\text{classif}}) \geq k$.

Similarly, since $\text{Pdim}(\mathcal{H}) = d_{\text{VC}}(\{(x, t) \mapsto \text{sign}(h(x) - t) \mid h \in \mathcal{H}\})$ for any hypothesis class \mathcal{H} , the set of points $\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}$ with thresholds $t_1 = t_2 = \dots = t_k = 0$ is shattered by the hypothesis class $\{(\mathbf{x}, t) \mapsto \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle - t) \mid \mathbf{w} \in V\}$, and thus $\text{Pdim}(\mathcal{H}^{\text{regression}}) \geq k$.

Lastly, the set of indices i_1, \dots, i_k with thresholds $t_1 = t_2 = \dots = t_k = 0$ is shattered by the class $\{(i, t) \mapsto \text{sign}(\mathbf{w}_i - t) \mid \mathbf{w} \in V\}$, and thus $\text{Pdim}(\mathcal{H}^{\text{completion}}) \geq k$. \square

3.4.2. Rank-One Tensors

Theorem 12. Let $G_{\text{rank-one}} = \overset{d_1}{\bullet} \overset{d_1}{\bullet} \dots \overset{d_1}{\bullet} \overset{d_1}{\bullet}$ be the tensor network structure corresponding to p -th order rank-one tensors, i.e., $\mathcal{T}(G_{\text{rank-one}}) = \{\mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \dots \otimes \mathbf{u}_p \mid \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p \in \mathbb{R}^d\}$. The VC-dimension and pseudo-dimensions $d_{\text{VC}}(\mathcal{H}_{G_{\text{rank-one}}}^{\text{classif}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{rank-one}}}^{\text{regression}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{rank-one}}}^{\text{completion}})$ are all lower-bounded by $(d-1)p$.

PROOF. We show that the set of indices

$$S = \left\{ \underbrace{(d, \dots, d)}_{i-1 \text{ times}}, j, \underbrace{(d, \dots, d)}_{p-i \text{ times}} \mid i \in [p], j \in [d-1] \right\}$$

is shattered by $\mathcal{T}(G_{\text{rank-one}})$, the result then follows from Lemma 3.4.1. More precisely, we show that S is shattered by the set of rank-one tensors

$$A = \left\{ \begin{pmatrix} \mathbf{v}_1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} \mathbf{v}_2 \\ 1 \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} \mathbf{v}_p \\ 1 \end{pmatrix} \mid \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p \in \mathbb{R}^{d-1} \right\} \subset \mathcal{T}(G_{\text{rank-one}}).$$

Indeed, for any multi-index $\underbrace{(d, \dots, d)}_{i-1 \text{ times}}, j, \underbrace{(d, \dots, d)}_{p-i \text{ times}} \in S$ and any rank one tensor $\mathbf{X} = \begin{pmatrix} \mathbf{v}_1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} \mathbf{v}_2 \\ 1 \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} \mathbf{v}_p \\ 1 \end{pmatrix} \in A$, we have

$$\mathcal{X}_{\underbrace{d, \dots, d}_{i-1 \text{ times}}, j, \underbrace{d, \dots, d}_{p-i \text{ times}}} = \left(\begin{pmatrix} \mathbf{v}_1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} \mathbf{v}_2 \\ 1 \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} \mathbf{v}_p \\ 1 \end{pmatrix} \right)_{d, \dots, d, j, d, \dots, d} = (\mathbf{v}_i)_j.$$

It follows that the $(d-1)p$ components $\mathcal{X}_{i_1, \dots, i_p}$ for $\mathcal{X} \in A$ and $(i_1, \dots, i_p) \in S$ can take any arbitrary values (the entries of the vectors $\mathbf{v}_1, \dots, \mathbf{v}_p \in \mathbb{R}^{d-1}$) and thus, that S is shattered by A and accordingly by $\mathcal{T}(G_{\text{rank-one}})$. The result then directly follows from Lemma 3.4.1. \square

3.4.3. Tensor Train and Tensor Ring

Theorem 13. Let $r \leq d^{\lfloor \frac{p-1}{2} \rfloor}$, let $G_{TT}(r) = \overset{r}{\bullet} \overset{r}{\bullet} \dots \overset{r}{\bullet} \overset{r}{\bullet}$ be the tensor network structure corresponding to p th order tensors of tensor train rank at most r , and let $G_{TR}(r) = \overset{r}{\bullet} \overset{r}{\bullet} \dots \overset{r}{\bullet} \overset{r}{\bullet}$ be the tensor network structure corresponding to p th order tensors of tensor ring rank at most r . Then, the VC-dimension and pseudo-dimensions $d_{\text{VC}}(\mathcal{H}_{G_{TT}(r)}^{\text{classif}})$, $d_{\text{VC}}(\mathcal{H}_{G_{TR}(r)}^{\text{classif}})$, $\text{Pdim}(\mathcal{H}_{G_{TT}(r)}^{\text{regression}})$, $\text{Pdim}(\mathcal{H}_{G_{TR}(r)}^{\text{regression}})$, $\text{Pdim}(\mathcal{H}_{G_{TT}(r)}^{\text{completion}})$ and $\text{Pdim}(\mathcal{H}_{G_{TR}(r)}^{\text{completion}})$ are all lower-bounded by $r^2 d$.

Moreover, in the particular case where $r = d$ and $p = 3k$ for some $k \in \mathbb{N}$, the VC-dimension and pseudo-dimensions $d_{\text{VC}}(\mathcal{H}_{G_{\text{TT}}(r)}^{\text{classif}})$, $d_{\text{VC}}(\mathcal{H}_{G_{\text{TR}}(r)}^{\text{classif}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{TT}}(r)}^{\text{regression}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{TR}}(r)}^{\text{regression}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{TT}}(r)}^{\text{completion}})$ and $\text{Pdim}(\mathcal{H}_{G_{\text{TR}}(r)}^{\text{completion}})$ are all lower-bounded by $\frac{p(r^2d-1)}{3}$.

PROOF. We start with the tensor train case, the tensor ring case will be handled similarly. Let $r \leq d^{\lfloor \frac{p-1}{2} \rfloor}$. We will show that there exists a set of r^2d indices $(i_1, \dots, j_1), \dots, (i_{r^2d}, \dots, j_{r^2d})$ that is shattered by $\mathcal{T}(G_{\text{TT}}(r))$ (the set of tensors of tensor train rank at most r), i.e., such that

$$\left| \{(\text{sign}(\mathcal{W}_{i_1, \dots, j_1}), \text{sign}(\mathcal{W}_{i_2, \dots, j_2}), \dots, \text{sign}(\mathcal{W}_{i_{r^2d}, \dots, j_{r^2d}})) \mid \mathcal{W} \in \mathcal{T}(G_{\text{TT}}(r))\} \right| = 2^{r^2d} .$$

In order to do so, we will consider a tensor train tensor \mathcal{T} with cores $\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(p)}$, where the $(k+1)$ -th core $\mathcal{G}^{(k+1)}$ will be free while the other cores are fixed in such a way that each component of $\mathcal{G}^{(k+1)}$ appears exactly once in the entries of \mathcal{T} .

Let $\mathbf{e}_1, \dots, \mathbf{e}_r$ be the canonical basis of \mathbb{R}^r and let $\mathbf{e}_i = \mathbf{0}$ for any $i > r$. Let $k = \lfloor \frac{p}{2} \rfloor$ and let $\mathcal{G}^{(k)}$ be the k -th core of the tensor train tensor \mathcal{T} (i.e., the middle core). The other cores of \mathcal{T} are defined as follows: for each $j \in [d]$,

$$\begin{aligned} \mathcal{G}_{j,:}^{(1)} &= \mathbf{e}_j^\top \\ \mathcal{G}_{:,j}^{(s)} &= \mathbf{e}_1 \mathbf{e}_{(j-1)d^{s-1}+1}^\top + \mathbf{e}_2 \mathbf{e}_{(j-1)d^{s-1}+2}^\top + \dots + \mathbf{e}_r \mathbf{e}_{(j-1)d^{s-1}+r}^\top && \text{for } s = 2, \dots, k-1 \\ \mathcal{G}_{:,j}^{(s)} &= \mathbf{e}_{(j-1)d^{p-s}+1} \mathbf{e}_1^\top + \mathbf{e}_{(j-1)d^{p-s}+2} \mathbf{e}_2^\top + \dots + \mathbf{e}_{(j-1)d^{p-s}+r} \mathbf{e}_r^\top && \text{for } s = k+1, \dots, p-1 \\ \mathcal{G}_{:,j}^{(p)} &= \mathbf{e}_j . \end{aligned}$$

With these definitions, one can check that

$$\mathcal{G}_{i_1,:}^{(1)} \mathcal{G}_{:,i_2}^{(2)} \mathcal{G}_{:,i_3}^{(3)} \dots \mathcal{G}_{:,i_{k-1},:}^{(k-1)} = \mathbf{e}_{i_1+(i_2-1)d+(i_3-1)d^2+\dots+(i_{k-1}-1)d^{k-2}}^\top$$

for any $i_1, \dots, i_{k-1} \in [d]$ and

$$\mathcal{G}_{:,i_{k+1},:}^{(k+1)} \mathcal{G}_{:,i_{k+2},:}^{(k+2)} \dots \mathcal{G}_{:,i_{p-1},:}^{(p-1)} \mathcal{G}_{:,i_p}^{(p)} = \mathbf{e}_{i_p+(i_{p-1}-1)d+(i_{p-2}-1)d^2+\dots+(i_{k+1}-1)d^{p-k-1}}$$

for any $i_{k+1}, \dots, i_p \in [d]$. Letting $\llbracket j_0, \dots, j_t \rrbracket = j_0 + (j_1 - 1)d + (j_2 - 1)d^2 + \dots + (j_t - 1)d^t$ for any $j_0, \dots, j_t \in [d]$, it follows that for any $i_1, \dots, i_p \in [d]$,

$$\mathcal{T}_{i_1, \dots, i_p} = \begin{cases} \mathcal{G}_{\llbracket i_1, i_2, \dots, i_{k-1} \rrbracket, i_k, \llbracket i_p, i_{p-1}, \dots, i_{k+1} \rrbracket}^{(k)} & \text{if } \llbracket i_1, i_2, \dots, i_{k-1} \rrbracket \leq r \text{ and } \llbracket i_p, i_{p-1}, \dots, i_{k+1} \rrbracket \leq r \\ 0 & \text{otherwise.} \end{cases}$$

Before continuing, we write a lemma that will be frequently used in our proofs for the lower bounds.

Lemma 3.4.2. *For any integer base, $b \geq 2$, every natural number has a unique base representation.*

One result of this lemma is that, if we take base b , then the non-negative integer number $n = a_0 + a_1b + a_2b^2 + \dots + a_{t-1}b^{t-1}$ with all integers $0 \leq a_i < b$ for $0 \leq i \leq t-1$, is always lower than or equal to $b^t - 1$. This lemma implies that for any natural number $n < b^t$, there exists one and only one tuple of corresponding coefficients $(a_0, a_1, \dots, a_{t-1})$ which generates n in base b as described above. Note that this is consistent with the fact that there are b^t of such coefficient tuples. Then, since one of the counted generated numbers is zero, which happens when all a_i 's are zero, the uniqueness property as stated in Lemma 3.4.2 results in the largest generated number being $b^t - 1$.

Back to the proof, Since $r \leq d^{\lfloor \frac{p-1}{2} \rfloor}$ and $k = \lfloor \frac{p}{2} \rfloor$, this implies that for any k -th core $\mathcal{G}^{(k)}$, the tensor train tensor \mathcal{T} contains all the r^2d entries of $\mathcal{G}^{(k+1)}$. Thus, the set of r^2d indices $\{(i_1, \dots, i_p) \mid \llbracket i_1, i_2, \dots, i_{k-1} \rrbracket \leq r, i_k \in [d], \llbracket i_p, i_{p-1}, \dots, i_{k+1} \rrbracket \leq r\}$ is shattered by $\mathcal{T}(G_{\text{TT}}(r))$ and the first part of the theorem follows from Lemma 3.4.1.

We now prove the second part of the theorem for the TT case, using a different construction. Let $r = d$ and $p = 3k$ for some $k \in \mathbb{N}$. We will construct a family of tensors in $\mathcal{T}(G_{\text{TT}}(r))$ where a third of the $p = 3k$ cores will be free while the other cores are fixed in such a way that the resulting tensor \mathcal{T} can be seen as the outer product of k 3rd order tensor of size $d \times d \times d$. By observing that such tensors can be interpreted as rank one k -th order tensors in $\mathbb{R}^{d^3 \times d^3 \times \dots \times d^3}$, the second part of the theorem will follow from Theorem 12.

Let $\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(p)}$ be the core tensors of the TT decomposition. The core tensors $\mathcal{G}^{(3s+2)} \in \mathbb{R}^{d \times d \times d}$ for $s = 0, \dots, p-1$ are free while the other cores are defined as follows: for any $j \in [d]$,

$$\begin{aligned} \mathcal{G}_{j,:}^{(1)} &= \mathbf{e}_j^\top \\ \mathcal{G}_{:,j,:}^{(3s+3)} &= \mathbf{e}_j \mathbf{e}_1^\top && \text{for } s = 0, \dots, k-2 \\ \mathcal{G}_{:,j,:}^{(3s+1)} &= \mathbf{e}_1 \mathbf{e}_j^\top && \text{for } s = 1, \dots, k-1 \\ \mathcal{G}_{:,j}^{(p)} &= \mathbf{e}_j \end{aligned}$$

It follows that, for any $i_1, \dots, i_p \in [d]$, we have

$$\begin{aligned} \mathcal{T}_{i_1, \dots, i_p} &= \mathcal{G}_{i_1,:}^{(1)} \mathcal{G}_{:,i_2,:}^{(2)} \dots \mathcal{G}_{:,i_{p-1},:}^{(p-1)} \mathcal{G}_{:,i_p}^{(p)} \\ &= (\mathbf{e}_{i_1}^\top) (\mathcal{G}_{:,i_2,:}^{(2)}) (\mathbf{e}_{i_3} \mathbf{e}_1^\top) (\mathbf{e}_1 \mathbf{e}_{i_4}^\top) (\mathcal{G}_{:,i_5,:}^{(5)}) (\mathbf{e}_{i_6} \mathbf{e}_1^\top) \dots (\mathbf{e}_1 \mathbf{e}_{i_{p-2}}^\top) (\mathcal{G}_{:,i_{p-1},:}^{(p-1)}) (\mathbf{e}_{i_p}) \\ &= \mathcal{G}_{i_1, i_2, i_3}^{(2)} \mathcal{G}_{i_4, i_5, i_6}^{(5)} \dots \mathcal{G}_{i_{p-2}, i_{p-1}, i_p}^{(p-1)} \end{aligned}$$

which implies that $\mathcal{T} = \mathcal{G}^{(2)} \otimes \mathcal{G}^{(5)} \otimes \dots \otimes \mathcal{G}^{(p-1)} = \bigotimes_{s=0}^{k-1} \mathcal{G}^{(3s+2)}$. By reshaping the set of tensors constructed in this way into k -th order tensors in $\mathbb{R}^{d^3 \times \dots \times d^3}$, one can see that

this set of tensors is exactly the set of rank-one k -th order tensors of size $\underbrace{d^3 \times \dots \times d^3}_{k \text{ times}}$, for which the corresponding VC-dimension and pseudo-dimensions are lower-bounded by $k(d^3 - 1) = p(r^2d - 1)/3$ from Theorem 12.

To see a simple example of how the proof for this second part works, Figure 3.4 illustrates the proof for a 9-th order uniform tensor train with rank $r = d$. In this figure, the core tensors highlighted in red are free. The other cores are fixed according to Equations (3.4.1).

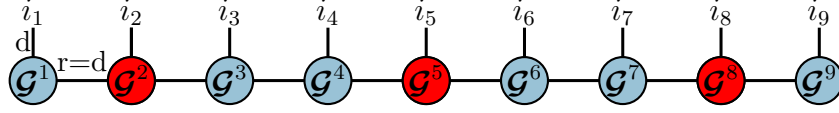


Figure 3.4. Visualization of the proof of the lower bound on the VC-dimension of a tensor train tensor

$$\begin{aligned} \mathcal{G}_{i_1, :}^{(1)} &= \mathbf{e}_{i_1}^\top, & \mathcal{G}_{:, i_9}^{(9)} &= \mathbf{e}_{i_9} \\ \mathcal{G}_{:, j, :}^{(3)} &= \mathcal{G}_{:, j, :}^{(6)} = \mathbf{e}_j \mathbf{e}_1^\top, & \mathcal{G}_{:, j, :}^{(4)} &= \mathcal{G}_{:, j, :}^{(7)} = \mathbf{e}_1 \mathbf{e}_j^\top \end{aligned} \quad (3.4.1)$$

It follows that, for any $i_1, \dots, i_9 \in [d]$, we have

$$\begin{aligned} \mathcal{T}_{i_1, \dots, i_9} &= \mathcal{G}_{i_1, :}^{(1)} \mathcal{G}_{:, i_2, :}^{(2)} \mathcal{G}_{:, i_3, :}^{(3)} \mathcal{G}_{:, i_4, :}^{(4)} \mathcal{G}_{:, i_5, :}^{(5)} \mathcal{G}_{:, i_6, :}^{(6)} \mathcal{G}_{:, i_7, :}^{(7)} \mathcal{G}_{:, i_8, :}^{(8)} \mathcal{G}_{:, i_9}^{(9)} \\ &= (\mathbf{e}_{i_1}^\top) (\mathcal{G}_{:, i_2, :}^{(2)}) (\mathbf{e}_{i_3} \mathbf{e}_1^\top) (\mathbf{e}_1 \mathbf{e}_{i_4}^\top) (\mathcal{G}_{:, i_5, :}^{(5)}) (\mathbf{e}_{i_6} \mathbf{e}_1^\top) (\mathbf{e}_1 \mathbf{e}_{i_7}^\top) (\mathcal{G}_{:, i_8, :}^{(8)}) (\mathbf{e}_{i_9}) \\ &= \mathcal{G}_{i_1, i_2, i_3}^{(2)} \mathcal{G}_{i_4, i_5, i_6}^{(5)} \mathcal{G}_{i_7, i_8, i_9}^{(8)} \end{aligned}$$

This expression is equivalent to

$$\mathcal{T} = \mathcal{G}^2 \otimes \mathcal{G}^5 \otimes \mathcal{G}^8, \quad (3.4.2)$$

and from Theorem 12 results in the following bound on the VC-dimension

$$d_{VC} \geq p(r^2d - 1)/3 \quad (3.4.3)$$

Finally, turning to the tensor ring, the proof for this case uses the exact same constructions with the difference in the definition of the first and last core tensors which are defined by $\mathcal{G}_{:, j, :}^{(1)} = \mathbf{e}_1 \mathbf{e}_j^\top$ and $\mathcal{G}_{:, j, :}^{(p)} = \mathbf{e}_j \mathbf{e}_1^\top$ for each $j \in [d]$. With these definitions, one can check that

$$\mathcal{G}_{:, i_1, :}^{(1)} \mathcal{G}_{:, i_2, :}^{(2)} \mathcal{G}_{:, i_3, :}^{(3)} \dots \mathcal{G}_{:, i_{k-1}, :}^{(k-1)} = \mathbf{e}_1 \mathbf{e}_{[i_1, i_2, \dots, i_{k-1}]}^\top$$

for any $i_1, \dots, i_{k-1} \in [d]$ and

$$\mathcal{G}_{:, i_{k+1}, :}^{(k+1)} \mathcal{G}_{:, i_{k+2}, :}^{(k+2)} \dots \mathcal{G}_{:, i_{p-1}, :}^{(p-1)} \mathcal{G}_{:, i_p, :}^{(p)} = \mathbf{e}_{[i_p, i_{p-1}, i_{k+1}]} \mathbf{e}_1^\top$$

for any $i_{k+1}, \dots, i_p \in [d]$. It follows that for any $i_1, \dots, i_p \in [d]$,

$$\begin{aligned}
\mathcal{T}_{i_1, \dots, i_p} &= \text{Tr} \left(\mathcal{G}_{:,i_1,:}^{(1)} \mathcal{G}_{:,i_2,:}^{(2)} \mathcal{G}_{:,i_3,:}^{(3)} \cdots \mathcal{G}_{:,i_{k-1},:}^{(k-1)} \mathcal{G}_{:,i_k,:}^{(k)} \mathcal{G}_{:,i_{k+1},:}^{(k+1)} \mathcal{G}_{:,i_{k+2},:}^{(k+2)} \cdots \mathcal{G}_{:,i_{p-1},:}^{(p-1)} \mathcal{G}_{:,i_p,:}^{(p)} \right) \\
&= \begin{cases} \mathcal{G}_{[[i_1, i_2, \dots, i_{k-1}], i_k, [[i_p, i_{p-1}, \dots, i_{k+1}]]}^{(k)} & \text{if } [[i_1, i_2, \dots, i_{k-1}]] \leq r \text{ and } [[i_p, i_{p-1}, \dots, i_{k+1}]] \leq r \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

The proof of the first part of the theorem then follows the exact same argument as for the TT case. The second part of the theorem for TR is proved exactly as the one for TT by replacing the first and last cores again by $\mathcal{G}_{:,j,:}^{(1)} = \mathbf{e}_1 \mathbf{e}_j^\top$ and $\mathcal{G}_{:,j,:}^{(p)} = \mathbf{e}_j \mathbf{e}_1^\top$ for each $j \in [d]$. \square

3.4.4. Tucker

Theorem 14. *Let $r \leq d$ and let $G_{\text{Tucker}}(r) = \text{diag}_r^d \text{diag}_r^d \cdots \text{diag}_r^d \text{diag}_r^d$ be the tensor network structure corresponding to p -th order tensors of Tucker rank at most r . Then, the VC-dimension and pseudo-dimensions $d_{\text{VC}}(\mathcal{H}_{G_{\text{Tucker}}(r)}^{\text{classif}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{Tucker}}(r)}^{\text{regression}})$ and $\text{Pdim}(\mathcal{H}_{G_{\text{Tucker}}(r)}^{\text{completion}})$ are all lower-bounded by r^p .*

PROOF. Let $r \leq d$. We show that there exists a set of r^p indices $(i_1, \dots, j_1), \dots, (i_{r^p}, \dots, j_{r^p})$ that is shattered by $\mathcal{T}(G_{\text{Tucker}}(r))$ (the set of tensors of Tucker rank at most r), i.e., such that

$$|\{(\text{sign}(\mathcal{W}_{i_1, \dots, j_1}), \text{sign}(\mathcal{W}_{i_2, \dots, j_2}), \dots, \text{sign}(\mathcal{W}_{i_{r^p}, \dots, j_{r^p}})) \mid \mathcal{W} \in \mathcal{T}(G_{\text{Tucker}}(r))\}| = 2^{r^p} .$$

Let $\mathbf{P} = \begin{pmatrix} \mathbf{I}_{r \times r} & \mathbf{0}_{r \times (d-r)} \end{pmatrix}^\top \in \mathbb{R}^{d \times r}$. We consider the following subset of $\mathcal{T}(G_{\text{Tucker}}(r))$:

$$A = \{\mathcal{G} \times_1 \mathbf{P} \times_2 \mathbf{P} \times_3 \cdots \times_p \mathbf{P} \mid \mathcal{G} \in \mathbb{R}^{r \times r \times \cdots \times r}\} \subset \mathcal{T}(G_{\text{Tucker}}(r))$$

where \times_k denotes the mode- k product (see, e.g., [11]). It is easy to see that any tensor $\mathcal{T} = \mathcal{G} \times_1 \mathbf{P} \times_2 \mathbf{P} \times_3 \cdots \times_p \mathbf{P} \in A$ will have entries $\mathcal{T}_{i_1, \dots, i_p} = \mathcal{G}_{i_1, \dots, i_p}$ for any $i_1, \dots, i_p \in [r]$. Hence the set of r^p indices $[r] \times [r] \times \cdots \times [r] \subset [d] \times [d] \times \cdots \times [d]$ is shattered by $\mathcal{T}(G_{\text{Tucker}}(r))$ and the result directly follows from Lemma 3.4.1. \square

3.4.5. CP

Theorem 15. *Let $r \leq d^{p-1}$ and let $G_{\text{CP}}(r) = \text{diag}_r^d \text{diag}_r^d \cdots \text{diag}_r^d \text{diag}_r^d$ be the tensor network structure corresponding to p -th order tensors of CP rank at most r . Then, the VC-dimension and pseudo-dimensions $d_{\text{VC}}(\mathcal{H}_{G_{\text{CP}}(r)}^{\text{classif}})$, $\text{Pdim}(\mathcal{H}_{G_{\text{CP}}(r)}^{\text{regression}})$ and $\text{Pdim}(\mathcal{H}_{G_{\text{CP}}(r)}^{\text{completion}})$ are all lower-bounded by rd .*

PROOF. Let $r \leq d^{p-1}$. We show that there exists a set of rd indices $(i_1, \dots, j_1), \dots, (i_{rd}, \dots, j_{rd})$ that is shattered by $\mathcal{T}(G_{\text{CP}}(r))$ (the set of tensors of CP rank at most r), i.e., such that

$$|\{(\text{sign}(\mathcal{W}_{i_1, \dots, j_1}), \text{sign}(\mathcal{W}_{i_2, \dots, j_2}), \dots, \text{sign}(\mathcal{W}_{i_{rd}, \dots, j_{rd}})) \mid \mathcal{W} \in \mathcal{T}(G_{\text{CP}}(r))\}| = 2^{rd}.$$

We construct a tensor \mathcal{T} of CP rank at most r such that each component of a matrix $\mathbf{A} \in \mathbb{R}^{d \times r}$ appears at least once in the entries of \mathcal{T} . Similarly to the previous proofs, \mathbf{A} will be a free parameter allowed to take any value while the other components of the parametrization of \mathcal{T} will be fixed.

Let $\mathbf{A} \in \mathbb{R}^{d \times r}$, we define p tensors $\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(p)} \in \mathbb{R}^{d \times \dots \times d}$ of order p as follows: for all $i_1, \dots, i_p, \tau_1, \dots, \tau_{p-1} \in [d]$,

$$\mathcal{A}_{i_1, \tau_1, \dots, \tau_{p-1}}^{(1)} = \begin{cases} \mathbf{A}_{i_1, \tau_1 + (\tau_2 - 1)d + \dots + (\tau_{p-1} - 1)d^{p-2}} & \text{if } \tau_1 + (\tau_2 - 1)d + \dots + (\tau_{p-1} - 1)d^{p-2} \leq r \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{A}_{i_s, \tau_1, \dots, \tau_{p-1}}^{(s)} = \delta_{i_s, \tau_{s-1}} \quad \text{for } s = 2, \dots, p$$

where δ is the Kronecker symbol. Let $S = \{(\tau_1, \dots, \tau_{p-1}) \in [d] \times \dots \times [d] \mid \tau_1 + (\tau_2 - 1)d + \dots + (\tau_{p-1} - 1)d^{p-2} \leq r\}$. Note that since $r \leq d^{p-1}$ and $\tau_1, \dots, \tau_{p-1} \in [d]$, Lemma 3.4.2 implies that

$$|S| = r \tag{3.4.4}$$

Let $\mathcal{T} \in \mathbb{R}^{d \times \dots \times d}$ be the p -th order tensor defined by

$$\mathcal{T}_{i_1, i_2, \dots, i_p} = \sum_{\tau_1=1}^d \sum_{\tau_2=1}^d \dots \sum_{\tau_{p-1}=1}^d \mathcal{A}_{i_1, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(1)} \mathcal{A}_{i_2, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(2)} \dots \mathcal{A}_{i_p, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(p)}$$

for all $i_1, \dots, i_p \in [d]$. It can easily be checked that \mathcal{T} is a tensor of CP rank at most r , i.e., $\mathcal{T} \in \mathcal{T}(G_{\text{CP}}(r))$. Indeed, from the definition of $\mathcal{A}^{(1)}$, we have

$$\begin{aligned} \mathcal{T}_{i_1, i_2, \dots, i_p} &= \sum_{\tau_1=1}^d \sum_{\tau_2=1}^d \dots \sum_{\tau_{p-1}=1}^d \mathcal{A}_{i_1, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(1)} \mathcal{A}_{i_2, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(2)} \dots \mathcal{A}_{i_p, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(p)} \\ &= \sum_{(\tau_1, \dots, \tau_{p-1}) \in S} \mathcal{A}_{i_1, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(1)} \mathcal{A}_{i_2, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(2)} \dots \mathcal{A}_{i_p, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(p)} \end{aligned}$$

where from Equation (3.4.4), the sum is over at most r terms. At the same time, we have

$$\begin{aligned}
\mathcal{T}_{i_1, i_2, \dots, i_p} &= \sum_{(\tau_1, \dots, \tau_{p-1}) \in S} \mathcal{A}_{i_1, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(1)} \mathcal{A}_{i_2, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(2)} \cdots \mathcal{A}_{i_p, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(p)} \\
&= \sum_{(\tau_1, \dots, \tau_{p-1}) \in S} \mathcal{A}_{i_1, \tau_1, \tau_2, \dots, \tau_{p-1}}^{(1)} \delta_{i_2, \tau_1} \delta_{i_3, \tau_2} \cdots \delta_{i_p, \tau_{p-1}} \\
&= \begin{cases} \mathbf{A}_{i_1, i_2 + (i_3 - 1)d + \dots + (i_p - 1)d^{p-2}} & \text{if } i_2 + (i_3 - 1)d + \dots + (i_p - 1)d^{p-2} \leq r \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Since for all values of $n \in [r]$ with $r \leq d^{p-1}$, there exists one tuple (i_2, i_3, \dots, i_p) for which $i_2 + (i_3 - 1)d + \dots + (i_p - 1)d^{p-2} = n$, each one of the components of \mathbf{A} appears exactly once in \mathcal{T} . In particular, this implies that the set of indices

$$\{(i_1, \dots, i_p) \in [d] \times \dots \times [d] \mid i_2 + (i_3 - 1)d + \dots + (i_p - 1)d^{p-2} \leq r\}$$

of size rd is shattered by $\mathcal{T}(G_{\text{CP}}(r))$. The theorem then directly follows from Lemma 3.4.1. \square

Before finishing this section, let us see two examples of the above proof. The first one is a 3-rd order tensor \mathcal{T} of uniform dimension d and rank $r = d$ and the second one is a 3-rd order tensor of uniform dimension d and rank $r = d^2$. In both cases, the CP-decomposition (1.4.3) takes the following form

$$\mathcal{T}_{i_1, i_2, i_3} = \sum_{k_1, k_2, k_3}^r \delta_{k_1, k_2, k_3} (\mathbf{T}_1)_{i_1, k_1} (\mathbf{T}_2)_{i_2, k_2} (\mathbf{T}_3)_{i_3, k_3} \quad (3.4.5)$$

In the first example, we define three matrices $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3 \in \mathbb{R}^{d \times d}$ as follows: \mathbf{T}_1 is a free matrix and \mathbf{T}_2 and \mathbf{T}_3 are both identity matrices $I_{d \times d}$. Then Equation (3.4.5) becomes

$$\begin{aligned}
\mathcal{T}_{i_1, i_2, i_3} &= \sum_{k_1=1}^r (\mathbf{T}_1)_{i_1, k_1} \sum_{k_2, k_3=1}^r \delta_{k_1, k_2, k_3} \delta_{i_2, k_2} \delta_{i_3, k_3} \\
&= \sum_{k_1=1}^{r=d} (\mathbf{T}_1)_{i_1, k_1} \delta_{k_1, i_2, i_3} = (\mathbf{T}_1)_{i_1, i_2} \delta_{i_2, i_3}
\end{aligned} \quad (3.4.6)$$

We observe that tensor \mathcal{T} has exactly as many entries as the free matrix \mathbf{T}_1 , i.e., $dr = d^2$ entries and therefore the proof for this first example is completed.

In the second example, we define three matrices $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3 \in \mathbb{R}^{d \times d^2}$ as follows: first off, \mathbf{T}_1 is a free matrix. Also, from Equation (3.4.5) we have

$$\mathcal{T}_{i_1, i_2, i_3} = \sum_{k=1}^r (\mathbf{T}_1)_{i_1, k} (\mathbf{T}_2)_{i_2, k} (\mathbf{T}_3)_{i_3, k} \quad (3.4.7)$$

Now, let's say we want each of the d^3 entries of $\mathbf{T}_1 \in \mathbb{R}^{d \times d^2}$ to appear once and only once in $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$. This is possible if for any $k \in [d^2]$ we have a unique pair (i_2, i_3) for which $(\mathbf{T}_2)_{i_2, k} (\mathbf{T}_3)_{i_3, k} = 1$. More precisely, if we make a 1-to-1 map between the index k on the one hand and (i_2, i_3) on the other hand, then our goal is realized. We identify each $k \in [d^2]$

with a pair (i_2, i_3) that satisfies $k = i_2 + (i_3 - 1)d$. Since $k \in [d^2]$, from Lemma 3.4.2, this is a 1-to-1 map. Having this relation, we can now construct the two matrices \mathbf{T}_2 and \mathbf{T}_3 ; each $k \in [d]$ is mapped to $(i_2, i_3) = (k, 1)$, each $k \in [d + 1, 2d]$ is mapped to $(i_2, i_3) = (k - d, 2)$, etc., each $k \in [d^2 - d + 1, d^2]$ is mapped to $(i_2, i_3) = (k - d(d - 1), d)$. Therefore, we have

$$(\mathbf{T}_2)_{i_2, k} = \begin{cases} 1 & \text{if } i_2 = k \pmod{d} \\ 0 & \text{otherwise} \end{cases}, \quad (\mathbf{T}_3)_{i_3, k} = \begin{cases} 1 & \text{if } i_3 = \lceil \frac{k}{d} \rceil \\ 0 & \text{otherwise} \end{cases}$$

or more explicitly

$$\mathbf{T}_2 = (\mathbf{I}_{d \times d} \quad \mathbf{I}_{d \times d} \quad \cdots \quad \mathbf{I}_{d \times d}) \in \mathbb{R}^{d \times d^2} \quad (3.4.8)$$

and

$$\mathbf{T}_3 = \begin{pmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots & \cdots & \vdots & & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & & 1 & \cdots & 1 & 1 \end{pmatrix}_{d \times d^2} \quad (3.4.9)$$

Note that, matrices \mathbf{T}_2 and \mathbf{T}_3 above are constructed in such a way that by fixing indices i_2 and i_3 , there is only one index k in the sum for which $(T_2)_{i_2, k}(T_3)_{i_3, k}$ is non-zero, and is equal to 1 (only one non-zero entry at each column).

Before finishing this chapter, it is worth mentioning that the approach that we took in this section to show the tightness of our upper bound on the VC/pseudo-dimension is different from the one in some similar works in the tensor network literature, such as [4, 48]. In order to examine the tightness of their bound, the authors of these works calculate some lower-bounds on the number of sign patterns produced by the corresponding hypothesis class. The reason why we did not use this method is that, it does not seem straightforward to extract a lower-bound on the VC-dimension from this lower-bound on the number of sign patterns. Therefore, we opted to directly lower-bound the VC/pseudo-dimensions. In spite of that, since the other approach involves some interesting proof techniques that worth being reviewed, in Appendix D, we write the proof for the matrix case [4] and also, we use the same technique to derive some lower bounds on the number of sign patterns produced by TT-based models.

Chapter 4

Conclusion and Future Directions

We derived a general upper bound on the VC and pseudo-dimension of a large class of tensor models parameterized by *arbitrary* tensor network structures for classification, regression and completion. We showed that this general bound can be applied to obtain bounds on the complexity of relevant machine learning models such as matrix and tensor completion, trace regression and TT-based linear classifiers. In particular, our result leads to an improved upper bound on the VC-dimension of low-rank matrices for completion tasks. As a corollary of our results, we answer the open question listed in [2] on the VC-dimension of the MPS classification model introduced in [1]. To demonstrate the tightness of our general upper bound, we derived a series of lower bounds for specific TN structures, notably showing that our bound is tight up to a constant for low-rank matrix models for completion, regression and classification.

Future directions include deriving tighter upper bounds and/or lower bounds for specific TN structures. This includes investigating whether our general upper bound can be tightened by removing the log factor in the number of vertices of the TN structure, deriving a stronger lower bound for CP and Tucker, and loosening the condition under which our stronger lower bound holds for TT and TR. Especially, in Appendix D.1.2 we discuss some evidence for the possibility of the tightness of the bound for the TT case in larger parts of the parameter space than the one shown in Table 3.1.

One limitation of the combinatorial complexity measures like VC/pseudo-dimension is their independence to the data distribution and the data samples. Studying other data-dependent complexity measures (e.g. Rademacher complexity [45]) and extending recent data-dependant generalization bounds for overparameterized deep neural networks, such as the ones used in [87, 88], to TN learning models is worth pursuing. Finally, building upon the connection between the depth of convolutional arithmetic circuits and tensor network structures introduced in [7], it is interesting to connect our result on the VC-dimension of tensor networks to the expressiveness and generalization ability of neural networks.

To elaborate more on this final direction, in [7] the well-known *depth efficiency* in neural networks, was examined theoretically for convolutional arithmetic circuits (CAC). Using the equivalence of these neural networks with some specific tensor networks, i.e., CP and hierarchical tucker, the authors use tensor network considerations to show that in general, representing a function that is realized by a deep CAC network of polynomial size, requires a shallow CAC of exponential size. One interesting question is whether this property has some implications for the VC-dimension of the corresponding tensor network models as well. Especially, to make analogy with [7], we need to consider tensor network learning models with rank-1 input data, i.e., the input tensorial data of order p represented as the tensor products of p vectors. Note that in our current work, we had no such constraint on the input data, while in [7], the fact that they consider rank-1 data, makes it possible to relate the levels of the tree of the hierarchical tucker to the layers of the neural network. Adding to this story, the dependence of the VC-dimension of *ReLU* neural networks on their depth [75], this question comes up: does there exist a similar dependence of the VC-dimension on the equivalent notion of depth in tensor networks as well? This being said, one interesting next step would be to study the dependence of the VC-dimension of CAC neural networks on their depth.

Bibliography

- [1] E. Stoudenmire and D. J. Schwab, “Supervised learning with tensor networks,” *Advances in Neural Information Processing Systems*, vol. 29, pp. 4799–4807, 2016.
- [2] J. I. Cirac, J. Garre-Rubio, and D. Pérez-García, “Mathematical open problems in projected entangled pair states,” *Revista Matemática Complutense*, vol. 32, no. 3, pp. 579–599, 2019.
- [3] T. Popoviciu, “Sur les équations algébriques ayant toutes leurs racines réelles,” *Mathematica*, vol. 9, pp. 129–145, 1935.
- [4] N. Srebro, N. Alon, and T. S. Jaakkola, “Generalization error bounds for collaborative prediction with low-rank matrices,” in *Advances In Neural Information Processing Systems*, pp. 1321–1328, 2005.
- [5] T. M. Cover, “Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition,” *IEEE transactions on electronic computers*, no. 3, pp. 326–334, 1965.
- [6] S. Miron, Y. Zniyed, R. Boyer, A. de Almeida, G. Favier, D. Brie, and P. Comon, “Tensor methods for multisensor signal processing,” *IET signal processing*, 2020.
- [7] N. Cohen, O. Sharir, and A. Shashua, “On the expressive power of deep learning: A tensor analysis,” in *Conference on learning theory*, pp. 698–728, PMLR, 2016.
- [8] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.
- [9] “Svd and data compression using low-rank matrix approximation.” <https://dustinstansbury.github.io/theclevermachine/svd-data-compression>.
- [10] F. L. Hitchcock, “The expression of a tensor or a polyadic as a sum of products,” *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.
- [11] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [12] I. V. Oseledets, “Tensor-train decomposition,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [13] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Annals of Physics*, vol. 349, pp. 117–158, 2014.
- [14] J. Biamonte and V. Bergholm, “Tensor networks in a nutshell,” *arXiv preprint arXiv:1708.00006*, 2017.
- [15] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Annals of physics*, vol. 326, no. 1, pp. 96–192, 2011.
- [16] R. Penrose, “Applications of negative dimensional tensors,” *Combinatorial mathematics and its applications*, vol. 1, pp. 221–244, 1971.

- [17] R. P. Feynman, “Quantum mechanical computers,” *Foundations of physics*, vol. 16, no. 6, pp. 507–531, 1986.
- [18] A. Novikov, D. Podoprikin, A. Osokin, and D. P. Vetrov, “Tensorizing neural networks,” in *Advances in neural information processing systems*, pp. 442–450, 2015.
- [19] H. N. Phien, H. D. Tuan, J. A. Bengua, and M. N. Do, “Efficient tensor completion: Low-rank tensor train,” *arXiv preprint arXiv:1601.01083*, 2016.
- [20] W. Wang, V. Aggarwal, and S. Aeron, “Efficient low rank tensor ring completion,” in *IEEE International Conference on Computer Vision*, 2017.
- [21] M. Hashemizadeh, M. Liu, J. Miller, and G. Rabusseau, “Adaptive tensor learning with tensor networks,” *arXiv preprint arXiv:2008.05437*, 2020.
- [22] Y. Yang, D. Krompass, and V. Tresp, “Tensor-train recurrent neural networks for video classification,” in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017.
- [23] A. Novikov, A. Rodomanov, A. Osokin, and D. P. Vetrov, “Putting mrfs on a tensor train,” in *Proceedings of the 31th International Conference on Machine Learning*, vol. 32, 2014.
- [24] P. Izmailov, A. Novikov, and D. Kropotov, “Scalable gaussian processes with billions of inducing inputs via tensor train decomposition,” in *International Conference on Artificial Intelligence and Statistics*, vol. 84, 2018.
- [25] R. Yu, M. G. Li, and Y. Liu, “Tensor regression meets gaussian processes,” in *International Conference on Artificial Intelligence and Statistics*, vol. 84, 2018.
- [26] A. Novikov, M. Trofimov, and I. Oseledets, “Exponential machines,” *arXiv preprint arXiv:1605.03795*, 2016.
- [27] I. Glasser, N. Pancotti, and J. I. Cirac, “From probabilistic graphical models to generalized tensor networks for supervised learning,” *IEEE Access*, vol. 8, pp. 68169–68182, 2020.
- [28] E. M. Stoudenmire, “Learning relevant features of data with multi-scale tensor networks,” *Quantum Science and Technology*, vol. 3, no. 3, p. 034003, 2018.
- [29] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, “Unsupervised generative modeling using matrix product states,” *Physical Review X*, vol. 8, no. 3, p. 031012, 2018.
- [30] J. Miller, G. Rabusseau, and J. Terilla, “Tensor networks for probabilistic sequence modeling,” in *The 24th International Conference on Artificial Intelligence and Statistics*, vol. 130, 2021.
- [31] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, “Tensor decompositions for learning latent variable models,” *Journal of machine learning research*, vol. 15, pp. 2773–2832, 2014.
- [32] N. Cohen and A. Shashua, “Convolutional rectifier networks as generalized tensor decompositions,” in *International Conference on Machine Learning*, pp. 955–963, PMLR, 2016.
- [33] O. Sharir and A. Shashua, “On the expressive power of overlapping architectures of deep learning,” *arXiv preprint arXiv:1703.02065*, 2017.
- [34] V. Khrulkov, A. Novikov, and I. V. Oseledets, “Expressive power of recurrent neural networks,” in *Proc. of ICLR*, 2018.
- [35] I. Glasser, R. Sweke, N. Pancotti, J. Eisert, and I. Cirac, “Expressive power of tensor-network factorizations for probabilistic modeling,” in *Advances in Neural Information Processing Systems*, pp. 1498–1510, 2019.

- [36] S. Adhikary, S. Srinivasan, J. Miller, G. Rabusseau, and B. Boots, “Quantum tensor networks, stochastic processes, and weighted automata,” in *The 24th International Conference on Artificial Intelligence and Statistics*, vol. 130, 2021.
- [37] L. Grasedyck, D. Kressner, and C. Tobler, “A literature survey of low-rank tensor approximation techniques,” *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013.
- [38] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [39] L. Grasedyck, “Hierarchical singular value decomposition of tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2029–2054, 2010.
- [40] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” *Journal of Fourier analysis and applications*, vol. 15, no. 5, pp. 706–722, 2009.
- [41] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, “Tensor ring decomposition,” *arXiv preprint arXiv:1606.05535*, 2016.
- [42] F. Verstraete, V. Murg, and J. I. Cirac, “Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems,” *Advances in Physics*, vol. 57, no. 2, pp. 143–224, 2008.
- [43] T. F. M. Anthony and P. L. Bartlett, “Neural network learning theoretical foundations,”
- [44] V. N. Vapnik and A. Y. Chervonenkis, “On the uniform convergence of relative frequencies of events to their probabilities,” in *Measures of complexity*, pp. 11–30, Springer, 2015.
- [45] V. Koltchinskii and D. Panchenko, “Rademacher processes and bounding the risk of function learning,” in *High dimensional probability II*, pp. 443–457, Springer, 2000.
- [46] H. E. Warren, “Lower bounds for approximation by nonlinear manifolds,” *Transactions of the American Mathematical Society*, vol. 133, no. 1, pp. 167–178, 1968.
- [47] M. Anthony and P. L. Bartlett, *Neural network learning: Theoretical foundations*. cambridge university press, 2009.
- [48] M. Nickel and V. Tresp, “An analysis of tensor models for learning on structured data,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 272–287, Springer, 2013.
- [49] M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Physical Review A*, vol. 103, no. 3, p. 032430, 2021.
- [50] L. Luo, Y. Xie, Z. Zhang, and W.-J. Li, “Support matrix machines,” in *International conference on machine learning*, pp. 938–947, 2015.
- [51] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, “Bilinear classifiers for visual recognition,” in *Advances in neural information processing systems*, pp. 1482–1490, 2009.
- [52] L. Wolf, H. Jhuang, and T. Hazan, “Modeling appearances with low-rank svm,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, IEEE, 2007.
- [53] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations for incomplete data,” *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.
- [54] K. Makantasis, A. D. Doulamis, N. D. Doulamis, and A. Nikitakis, “Tensor-based classification models for hyperspectral data analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 6884–6898, 2018.

- [55] D. Cai, X. He, J.-R. Wen, J. Han, and W.-Y. Ma, “Support tensor machines for text categorization,” tech. rep., 2006.
- [56] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” in *IEEE 12th International Conference on Computer Vision*, 2009.
- [57] S. Gandy, B. Recht, and I. Yamada, “Tensor completion and low-n-rank tensor recovery via convex optimization,” *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.
- [58] Z. He, J. Hu, and Y. Wang, “Low-rank tensor learning for classification of hyperspectral image with limited labeled samples,” *Signal Processing*, vol. 145, pp. 12–25, 2018.
- [59] G. Rabusseau and H. Kadri, “Low-rank regression with tensor responses,” in *Advances in Neural Information Processing Systems*, pp. 1867–1875, 2016.
- [60] C. Chen, Z.-B. Wu, Z.-T. Chen, Z.-B. Zheng, and X.-J. Zhang, “Auto-weighted robust low-rank tensor completion via tensor-train,” *Information Sciences*, vol. 567, pp. 100–115, 2021.
- [61] R. Selvan and E. B. Dam, “Tensor networks for medical image classification,” *arXiv preprint arXiv:2004.10076*, 2020.
- [62] Z. Chen, K. Batselier, J. A. Suykens, and N. Wong, “Parallelized tensor train learning of polynomial classifiers,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4621–4632, 2017.
- [63] Y. Wang, W. Zhang, Z. Yu, Z. Gu, H. Liu, Z. Cai, C. Wang, and S. Gao, “Support vector machine based on low-rank tensor train decomposition for big data applications,” in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 850–853, IEEE, 2017.
- [64] X. Xu, Q. Wu, S. Wang, J. Liu, J. Sun, and A. Cichocki, “Whole brain fmri pattern analysis based on tensor neural network,” *IEEE Access*, vol. 6, pp. 29297–29305, 2018.
- [65] S. Cheng, L. Wang, and P. Zhang, “Supervised learning with projected entangled pair states,” *arXiv preprint arXiv:2009.09932*, 2020.
- [66] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima, “Statistical performance of convex tensor decomposition,” in *Advances in Neural Information Processing Systems*, 2011.
- [67] R. Tomioka and T. Suzuki, “Convex tensor decomposition via structured Schatten norm regularization,” in *Advances in Neural Information Processing Systems*, 2013.
- [68] M. Imaizumi, T. Maehara, and K. Hayashi, “On tensor train rank minimization : Statistical efficiency and scalable algorithm,” in *Advances in Neural Information Processing Systems*, pp. 3930–3939, 2017.
- [69] B. Michel and A. Nouy, “Learning with tree tensor networks: complexity estimates and model selection,” *arXiv preprint arXiv:2007.01165*, 2020.
- [70] B. Khavari and G. Rabusseau, “Lower and upper bounds on the pseudo-dimension of tensor network models,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [71] C. J. Hillar and L.-H. Lim, “Most tensor problems are np-hard,” *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–39, 2013.
- [72] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [73] W. Wang, Y. Sun, B. Eriksson, W. Wang, and V. Aggarwal, “Wide compression: Tensor ring nets,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

- [74] L. Yuan, J. Cao, X. Zhao, Q. Wu, and Q. Zhao, “Higher-dimension tensor completion via low-rank tensor ring decomposition,” in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1071–1076, IEEE, 2018.
- [75] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, “Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks,” *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 2285–2301, 2019.
- [76] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.
- [77] E. J. Candès and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.
- [78] D. Gross, “Recovering low-rank matrices from few coefficients in any basis,” *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1548–1566, 2011.
- [79] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.
- [80] N. Hamidi and M. Bayati, “On low-rank trace regression under general sampling distribution,” *arXiv preprint arXiv:1904.08576*, 2019.
- [81] Y. Wang *et al.*, “Asymptotic equivalence of quantum state tomography and noisy matrix completion,” *The Annals of Statistics*, vol. 41, no. 5, pp. 2462–2504, 2013.
- [82] H. Kadri, S. Ayache, R. Huusari, A. Rakotomamonjy, and R. Liva, “Partial trace regression and low-rank kraus decomposition,” in *International Conference on Machine Learning*, pp. 5031–5041, PMLR, 2020.
- [83] V. Koltchinskii and D. Xia, “Optimal estimation of low rank density matrices.,” *J. Mach. Learn. Res.*, vol. 16, no. 53, pp. 1757–1792, 2015.
- [84] L. Grasedyck, M. Kluge, and S. Kramer, “Variants of alternating least squares tensor completion in the tensor train format,” *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. A2424–A2450, 2015.
- [85] W. Wang, V. Aggarwal, and S. Aeron, “Tensor completion by alternating minimization under the tensor train (tt) model,” *arXiv preprint arXiv:1609.05587*, 2016.
- [86] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao, “Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9151–9158, 2019.
- [87] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, “Stronger generalization bounds for deep nets via a compression approach,” in *International Conference on Machine Learning*, pp. 254–263, PMLR, 2018.
- [88] J. Li, Y. Sun, J. Su, T. Suzuki, and F. Huang, “Understanding generalization in deep learning via tensor methods,” in *International Conference on Artificial Intelligence and Statistics*, pp. 504–515, PMLR, 2020.
- [89] L. Babai and P. Frankl, *Linear algebra methods in combinatorics*. University of Chicago, 1988.

Appendix A

Useful Formulas

A.1. Essential Inequalities

We introduce some concentration inequalities in probability theory that have been used frequently in the text.

A.1.1. Markov's inequality

Theorem 16. (*Markov's inequality*). *Let X be a non-negative random variable. For any $t > 0$,*

$$\mathbb{P}[X > t] \leq \frac{\mathbb{E}[X]}{t}$$

PROOF. We use the definition of the expectation value of the random variable X , and we split the integral as follows

$$\mathbb{E}[X] = \int_0^\infty X\mathbb{P}[X]dX = \int_0^t X\mathbb{P}[X]dX + \int_t^\infty X\mathbb{P}[X]dX \quad (\text{A.1.1})$$

From that we have

$$\int_t^\infty X\mathbb{P}[X]dX = \mathbb{E}[X] - \int_0^t X\mathbb{P}[X]dX \leq \mathbb{E}[X] \quad (\text{A.1.2})$$

On the other hand we have

$$\int_t^\infty X\mathbb{P}[X]dX \geq t \int_t^\infty \mathbb{P}[X]dX \quad (\text{A.1.3})$$

Combining the two inequalities of Equation (A.1.2) and Equation (A.1.3), we get

$$t\mathbb{P}[X > t] = t \int_t^\infty \mathbb{P}[X]dX \leq \int_t^\infty X\mathbb{P}[X]dX \leq \mathbb{E}[X] \quad (\text{A.1.4})$$

or equivalently

$$\mathbb{P}[X > t] \leq \frac{\mathbb{E}[X]}{t} \quad (\text{A.1.5})$$

□

A.1.2. Chebyshev's inequality

Theorem 17. (*Chebyshev's inequality*) Let $\mu = \mathbb{E}[X]$ and $\sigma^2 = \mathbb{V}[X]$. Then, for any $t > 0$ we have

$$\mathbb{P}[|X - \mu| \geq t] \leq \frac{\sigma^2}{t^2} \quad (\text{A.1.6})$$

Equivalently, for any $k > 0$ we have

$$\mathbb{P}\left[\frac{|X - \mu|}{\sigma} \geq k\right] \leq \frac{1}{k^2} \quad (\text{A.1.7})$$

PROOF. We define positive random variable $(|X - \mu|)^2$. Its expectation value is by the definition of the variance, equal to $\mathbb{V}[X] = \sigma^2$. Using the fact that for any $t > 0$,

$$\mathbb{P}[|X - \mu| \geq t] = \mathbb{P}[(|X - \mu|)^2 \geq t^2], \quad (\text{A.1.8})$$

we get from Markov's inequality

$$\mathbb{P}[|X - \mu| \geq t] = \mathbb{P}[(|X - \mu|)^2 \geq t^2] \leq \frac{\mathbb{E}[(|X - \mu|)^2]}{t^2} = \frac{\mathbb{V}[X]}{t^2} = \frac{\sigma^2}{t^2}$$

and therefore

$$\mathbb{P}[|X - \mu| \geq t] \leq \frac{\sigma^2}{t^2} \quad (\text{A.1.9})$$

□

A.1.3. Hoeffding's inequality

Theorem 18. (*Hoeffding's inequality*). Let Y_1, \dots, Y_n be independent random variables such that $\mathbb{E}[Y_i] = 0$ and $Y_i \in [a_i, b_i]$ for all $i = 1, \dots, n$. Let $\epsilon > 0$. Then for any $t > 0$

$$\mathbb{P}\left[\sum_{i=1}^n Y_i \geq \epsilon\right] \leq \exp(-t\epsilon) \prod_{i=1}^n \exp\left(\frac{t^2(b_i - a_i)^2}{8}\right) \quad (\text{A.1.10})$$

PROOF. First, we consider the positive variable $e^{t \sum_i Y_i}$ and use Markov's inequality to obtain

$$\mathbb{P}\left[\sum_i Y_i \geq \epsilon\right] = \mathbb{P}\left[e^{t \sum_i Y_i} \geq e^{t\epsilon}\right] \leq \frac{\mathbb{E}\left[e^{t \sum_i Y_i}\right]}{e^{t\epsilon}} = \frac{\prod_i \mathbb{E}\left[e^{tY_i}\right]}{e^{t\epsilon}}, \quad (\text{A.1.11})$$

where we used the independence of Y_i 's to write the last equality. In order to upper-bound $\mathbb{E}[e^{tY_i}]$, we use the property that Y_i is bounded as $Y_i \in [a_i, b_i]$; therefore, it can be written as a convex combination of a_i and b_i as $Y_i = \alpha a_i + (1 - \alpha)b_i$, with $\alpha = \frac{Y_i - b_i}{a_i - b_i}$ (note that $\alpha \in [0, 1]$). Since e^{tY_i} is a convex function of Y_i , and a convex function f satisfies $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$ for $0 < \alpha < 1$, we have

$$e^{tY_i} = e^{\alpha t a_i + (1 - \alpha)t b_i} \leq \alpha e^{t a_i} + (1 - \alpha)e^{t b_i}$$

Taking expectations of both sides we obtain

$$\mathbb{E} \left[e^{tY_i} \right] \leq \mathbb{E}[\alpha]e^{ta_i} + (1 - \mathbb{E}[\alpha])e^{tb_i} \quad (\text{A.1.12})$$

and using $\mathbb{E}[Y_i] = 0$ we get

$$\mathbb{E}[\alpha] = \mathbb{E} \left[\frac{Y_i - b_i}{a_i - b_i} \right] = \frac{-b_i}{a_i - b_i}. \quad (\text{A.1.13})$$

Equation (A.1.12) can then be rewritten as

$$\mathbb{E} \left[e^{tY_i} \right] \leq \frac{-b_i}{a_i - b_i} e^{ta_i} + \frac{a_i}{a_i - b_i} e^{tb_i}. \quad (\text{A.1.14})$$

Finally, we use (one form of) Taylor's theorem which states that for a differentiable function g , there is a number $\eta \in (0, x)$ such that $g(x) = g(0) + xg'(0) + \frac{x^2}{2}g''(\eta)$.

To apply this theorem, we do the following change of variable to write $\frac{-b_i}{a_i - b_i} e^{ta_i} + \frac{a_i}{a_i - b_i} e^{tb_i} = e^{g(u)}$:

$$u = t(b_i - a_i) \quad , \quad g(u) = -\gamma u + \log(1 - \gamma + \gamma e^u) \quad \text{with} \quad \gamma = -\frac{a_i}{b_i - a_i}. \quad (\text{A.1.15})$$

We observe that $g(0) = g'(0) = 0$. The second derivative is given by

$$g''(u) = \frac{\gamma(1 - \gamma)e^u}{(1 - \gamma + \gamma e^u)^2}$$

which in terms of u , a_i and b_i is $g''(u) = \frac{-a_i b_i e^u}{(b_i - a_i e^u)^2}$. It can easily be seen that its maximum value is $\frac{1}{4}$. Therefore, we have shown that

$$g''(u) \leq \frac{1}{4}$$

Now, Taylor's theorem states that there exists $\eta \in (0, u)$ such that

$$g(u) = \frac{u^2}{2} g''(\eta) \leq \frac{u^2}{8} = \frac{t^2(b_i - a_i)^2}{8}$$

From Equation (A.1.14) we get

$$\mathbb{E}[e^{tY_i}] \leq \exp(g(u)) \leq \exp\left(\frac{t^2(b_i - a_i)^2}{8}\right)$$

and Hoeffding's theorem is proved. □

As we will see below, Hoeffding's inequality results in the following theorem.

Theorem 19. *If X_1, \dots, X_n are n random variables drawn i.i.d. from a Bernoulli distribution Bernoulli(p), or just to have values in $[0,1]$, then for all $\epsilon > 0$ we have*

$$\mathbb{P} \left[\left| \frac{1}{n} \sum_{i=1}^n X_i - p \right| > \epsilon \right] \leq 2 \exp(-2n\epsilon^2). \quad (\text{A.1.16})$$

or

$$\mathbb{P} \left[\left| \frac{1}{n} \sum_{i=1}^n X_i - \mathbb{E}[X_i] \right| > \epsilon \right] \leq 2 \exp(-2n\epsilon^2). \quad (\text{A.1.17})$$

PROOF. We introduce the new random variable $Y_i = \frac{1}{n}(X_i - p)$, for which $\mathbb{E}[Y_i] = 0$ and $Y_i \in [a = -\frac{p}{n}, b = \frac{1-p}{n}]$ hold. Then, $b - a = \frac{1}{n}$ and from Hoeffding's theorem we have

$$\mathbb{P} \left[\sum_{i=1}^n Y_i \geq \epsilon \right] \leq \exp(-t\epsilon) \prod_{i=1}^n \exp\left(t^2 \frac{1}{8n^2}\right) = \exp(-t\epsilon) \exp\left(\frac{t^2}{8n}\right). \quad (\text{A.1.18})$$

Since Hoeffding's inequality holds for arbitrary $t > 0$, we can put $t = \arg \min \left(\exp(-t\epsilon) \exp\left(\frac{t^2}{8n}\right) \right)$ to get the tightest bound in terms of t . This results in $t = 4n\epsilon$ which gives the following bound

$$\mathbb{P} \left[\sum_{i=1}^n Y_i \geq \epsilon \right] \leq e^{-2n\epsilon^2}. \quad (\text{A.1.19})$$

On the other hand, from the union bound, we have

$$\mathbb{P} \left[\left| \frac{1}{n} \sum_{i=1}^n X_i - p \right| \geq \epsilon \right] \leq \mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n X_i - p \geq \epsilon \right] + \mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n X_i - p \leq -\epsilon \right].$$

So, it remains for us to bound the following quantity

$$\mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n X_i - p \leq -\epsilon \right] = \mathbb{P} \left[\sum_{i=1}^n Y_i \leq -\epsilon \right] = \mathbb{P} \left[\sum_{i=1}^n (-Y_i) \geq \epsilon \right].$$

We can again use Hoeffding's theorem; we notice that $\mathbb{E}[(-Y_i)] = 0$ and $(-Y_i) \in [-b = -\frac{1-p}{n}, -a = \frac{p}{n}]$. For arbitrary $t > 0$, we have

$$\mathbb{P} \left[\sum_{i=1}^n (-Y_i) \geq \epsilon \right] \leq e^{-t\epsilon} \prod_{i=1}^n \exp\left(\frac{t^2}{8n}\right). \quad (\text{A.1.20})$$

As before, by choosing $t = 4n\epsilon$ we obtain the same bound as the one on $\mathbb{P}[\sum_{i=1}^n (Y_i) \geq \epsilon]$:

$$\mathbb{P} \left[\sum_{i=1}^n (-Y_i) \geq \epsilon \right] \leq e^{-2n\epsilon^2}. \quad (\text{A.1.21})$$

Replacing the above result in Equation (A.1.20), we find

$$\mathbb{P} \left[\left| \frac{1}{n} \sum_{i=1}^n X_i - p \right| \geq \epsilon \right] \leq \mathbb{P} \left[\left(\frac{1}{n} \sum_{i=1}^n X_i - p \geq \epsilon \right) \right] + \mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n X_i - p \leq -\epsilon \right] \leq 2e^{-2n\epsilon^2}. \quad (\text{A.1.22})$$

□

Also, it is straightforward to verify that this theorem holds for any bounded random variables. In this case, we have $b - a = \frac{M^2}{n^2}$ and the Hoeffding's expression is minimized by the value $t = \frac{4n\epsilon}{M^2}$ and gives the following result

Corollary A.1.1. *If X_1, \dots, X_n are n random variables drawn i.i.d. from a bounded distribution with values in $[0, M]$, then for all $\epsilon > 0$ we have*

$$\mathbb{P} \left[\left| \frac{1}{n} \sum_{i=1}^n X_i - p \right| > \epsilon \right] \leq 2 \exp \left(-\frac{2n\epsilon^2}{M^2} \right). \quad (\text{A.1.23})$$

or

$$\mathbb{P} \left[\left| \frac{1}{n} \sum_{i=1}^n X_i - \mathbb{E}[X_i] \right| > \epsilon \right] \leq 2 \exp \left(-\frac{2n\epsilon^2}{M^2} \right). \quad (\text{A.1.24})$$

A.1.4. Popoviciu's inequality [3]

Theorem 20. *Let b and a be the upper and lower bounds on the values of a random variable X with a particular probability distribution. Then Popoviciu's inequality states:*

$$\mathbb{V}[x] \leq \frac{(b-a)^2}{4} \quad (\text{A.1.25})$$

We will not prove this inequality. We only mention that this inequality is consistent with the intuition that the distribution with highest variance corresponds to having half of the data on one end, say a , and half of the data on the other end of the interval, b . In that case the expectation value of the distribution will be $\frac{a+b}{2}$ and the distance between every point and the expectation value is equal to $\frac{b-a}{2}$. Therefore, from the definition of variance, the highest possible value is given by $(\frac{b-a}{2})^2$, as stated by the theorem.

Appendix B

Proofs for Chapter 2

B.1. Proof of Lemma 2.2.1

Lemma. (*Symmetrization Lemma*) Let S and S' be two random samples of size n drawn from a distribution D . Then for any $t > 0$, with large enough n such that $nt^2 \geq 2$, we have

$$\mathbb{P}_{S \sim D} \left[\sup_{h \in \mathcal{H}} (\hat{R}_S(h) - R(h)) \geq t \right] \leq 2 \mathbb{P}_{S, S' \sim D} \left[\sup_{h \in \mathcal{H}} (\hat{R}_S(h) - \hat{R}_{S'}(h)) \geq \frac{t}{2} \right], \quad (\text{B.1.1})$$

PROOF. First we consider \tilde{h} as the hypothesis maximizing $(\hat{R}_S(h) - R(h))$:

$$\tilde{h} = \arg \sup_{h \in \mathcal{H}} (\hat{R}_S(h) - R(h))$$

and we define the corresponding errors for \tilde{h} as below

$$\epsilon = \hat{R}_S(\tilde{h}) - R(\tilde{h}) = \sup_{h \in \mathcal{H}} (\hat{R}_S(h) - R(h)) \quad , \quad \epsilon' = \hat{R}_{S'}(\tilde{h}) - R(\tilde{h})$$

We notice that if $\epsilon \geq t$ and $\epsilon' < \frac{t}{2}$, we have $\epsilon - \epsilon' \geq \frac{t}{2}$. As this last inequality is a deterministic result of the two first inequalities, we have $\mathbb{P} \left[\epsilon \geq t \text{ and } \epsilon' < \frac{t}{2} \right] \leq \mathbb{P} \left[\epsilon - \epsilon' \geq \frac{t}{2} \right]$. Since the two events $\epsilon \geq t$ and $\epsilon' < \frac{t}{2}$ are independent, we can rewrite this inequality as

$$\mathbb{P} [\epsilon \geq t] \mathbb{P} \left[\epsilon' < \frac{t}{2} \right] \leq \mathbb{P} \left[\epsilon - \epsilon' \geq \frac{t}{2} \right] \quad (\text{B.1.2})$$

Notice that as the events ϵ and ϵ' are defined in terms of different samples, the above probabilities are accordingly over S , S' or the mixed samples $S \cup S'$. Now, we notice from the definition of ϵ and ϵ' that $\epsilon - \epsilon' = \hat{R}_S(\tilde{h}) - \hat{R}_{S'}(\tilde{h})$. That is, for the rhs of equation (B.1.2), we can write

$$\begin{aligned} \mathbb{P} \left[\epsilon - \epsilon' \geq \frac{t}{2} \right] &= \mathbb{P} \left[\hat{R}_S(\tilde{h}) - \hat{R}_{S'}(\tilde{h}) \geq \frac{t}{2} \right] \leq \mathbb{P} \left[\sup_h (\hat{R}_S(h) - \hat{R}_{S'}(h)) \geq \frac{t}{2} \right] \\ &= \mathbb{P} \left[\exists h \in \mathcal{H} \mid \hat{R}_S(h) - \hat{R}_{S'}(h) \geq \frac{t}{2} \right] \quad (\text{B.1.3}) \end{aligned}$$

Therefore, up to now, we have found the following relation

$$\mathbb{P}[\epsilon \geq t] \mathbb{P}\left[\epsilon' < \frac{t}{2}\right] \leq \mathbb{P}\left[\sup_h(\hat{R}_S(h) - \hat{R}_{S'}(h)) \geq \frac{t}{2}\right] \quad (\text{B.1.4})$$

From (B.1.4), if we can lower-bound the expression $\mathbb{P}\left[\epsilon' < \frac{t}{2}\right]$, it will result in upper-bounding $\mathbb{P}[\epsilon \geq t]$. To do this, we can alternatively upper-bound the complementary event, which is $\epsilon' = \hat{R}_{S'}(\tilde{h}) - R(\tilde{h}) > \frac{t}{2}$. By taking this alternative approach we are able to again profit from some concentration inequalities. We start by considering the following natural inequality

$$\mathbb{P}\left[\epsilon' > \frac{t}{2}\right] \leq \mathbb{P}\left[|\epsilon'| > \frac{t}{2}\right] \quad (\text{B.1.5})$$

Then, from the fact that $\mathbb{E}[\epsilon'] = 0$ and by using Chebyshev's inequality given in Theorem 17, we have

$$\mathbb{P}\left[\epsilon' > \frac{t}{2}\right] \leq \mathbb{P}\left[|\epsilon'| > \frac{t}{2}\right] \leq \frac{\mathbb{V}[\epsilon']}{\left(\frac{t}{2}\right)^2} \quad (\text{B.1.6})$$

Now, in order to deal with the term $\mathbb{V}[\epsilon']$, we take into account that it is the variance of an average over *independent* random variables $Z_i = L(y_i, \tilde{h}(x'_i)) - R(\tilde{h})$ and rewrite the event $\epsilon' > \frac{t}{2}$ as $\frac{1}{n} \sum_{i=1}^n Z_i > \frac{t}{2}$ or equivalently $\sum_i Z_i > \frac{nt}{2}$. By definition of Z_i and since the true risk is nothing but the expectation value of the loss function, we see that the expectation value of the random variables Z_i is zero. Then from the fact that loss values are always in the interval $[0,1]$ and by using Popoviciu's inequality (A.1.25), the variance of Z_i is bounded as $\mathbb{V}[Z_i] \leq \frac{1}{4}$. therefore, we can rewrite Equation (B.1.6) as below

$$\mathbb{P}\left[\frac{1}{n} \sum_i Z_i > \frac{t}{2}\right] = \mathbb{P}\left[\sum_i Z_i > \frac{nt}{2}\right] \leq \mathbb{P}\left[|\sum_i Z_i| > \frac{nt}{2}\right] \leq \frac{\mathbb{V}[\sum_i Z_i]}{n^2 \left(\frac{t}{2}\right)^2} \quad (\text{B.1.7})$$

As Z_i 's are *i.i.d.* variables, this reduces to

$$\mathbb{P}\left[\sum_i Z_i > \frac{nt}{2}\right] \leq \frac{n \mathbb{V}[Z_i]}{n^2 \left(\frac{t}{2}\right)^2} \leq \frac{1}{4} \frac{1}{n \left(\frac{t}{2}\right)^2} = \frac{1}{nt^2} \quad (\text{B.1.8})$$

So, we have shown that

$$\mathbb{P}\left[\epsilon' > \frac{t}{2}\right] = \mathbb{P}\left[\hat{R}_{S'}(\tilde{h}) - R(\tilde{h}) > \frac{t}{2}\right] = \mathbb{P}\left[\frac{1}{n} \sum_{i=1}^n Z_i > \frac{t}{2}\right] \leq \frac{1}{nt^2} \quad (\text{B.1.9})$$

As explained after Equation (B.1.4) we are interested in the probability of the complimentary event, i.e. $\epsilon' < \frac{t}{2}$.

$$\mathbb{P}\left[\epsilon' < \frac{t}{2}\right] = \mathbb{P}\left[\hat{R}_{S'}(\tilde{h}) - R(\tilde{h}) < \frac{t}{2}\right] \geq 1 - \frac{1}{nt^2} \geq \frac{1}{2} \quad (\text{B.1.10})$$

where the last inequality comes from the Lemma's assumption, $nt^2 \geq \frac{1}{2}$. By replacing the above result in equation (B.1.4), we get

$$\mathbb{P}[\epsilon \geq t] \leq 2 \mathbb{P} \left[\sup_h \left(\hat{R}_S(h) - \hat{R}_{S'}(h) \right) \geq \frac{t}{2} \right] \quad (\text{B.1.11})$$

To put everything in place, we replace $\mathbb{P}[\epsilon \geq t]$ by $\mathbb{P} \left[\sup_{h \in \mathcal{H}} \left(\hat{R}_S(h) - R(h) \right) \geq t \right]$ to get

$$\mathbb{P} \left[\sup_{h \in \mathcal{H}} \left(\hat{R}_S(h) - R(h) \right) \geq t \right] \leq 2 \mathbb{P} \left[\sup_{h \in \mathcal{H}} \left(\hat{R}_S(h) - \hat{R}_{S'}(h) \right) \geq \frac{t}{2} \right] \quad (\text{B.1.12})$$

□

We close this section by mentioning that it is easy to show that the same approach could be taken to prove a similar symmetrization lemma as below

$$\mathbb{P} \left[\sup_{h \in \mathcal{H}} \left(R(h) - \hat{R}_S(h) \right) \geq t \right] \leq 2 \mathbb{P} \left[\sup_{h \in \mathcal{H}} \left(\hat{R}_{S'}(h) - \hat{R}_S(h) \right) \geq \frac{t}{2} \right] \quad (\text{B.1.13})$$

B.2. Proof of Corollary 2.2.2

Corollary. *If $Z_1, \dots, Z_n, Z'_1, \dots, Z'_n$ are $2n$ i.i.d. random variables drawn from a Bernoulli distribution, then for all $\epsilon > 0$ we have*

$$\mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n Z'_i > \epsilon \right] \leq 2 \exp \left(-\frac{n\epsilon^2}{2} \right)$$

PROOF. We first rewrite the left-hand side of the above relation as below

$$\mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n Z'_i > \epsilon \right] = \mathbb{P} \left[\sum_{i=1}^n \frac{1}{n} (Z_i - p) - \sum_{i=1}^n \frac{1}{n} (Z'_i - p) > \epsilon \right], \quad (\text{B.2.1})$$

with p being the expected value of random variables Z_i . As shown in the proof of Theorem 19, by defining $Y_i = \frac{1}{n}(Z_i - p)$, we have

$$\mathbb{P} \left[\sum_{i=1}^n (-Y_i) \geq \epsilon \right] \leq e^{-2n\epsilon^2} \quad \text{and} \quad \mathbb{P} \left[\sum_{i=1}^n Y_i \geq \epsilon \right] \leq e^{-2n\epsilon^2}$$

So we can write

$$\mathbb{P} \left[\sum_{i=1}^n \frac{1}{n} (Z_i - p) - \sum_{i=1}^n \frac{1}{n} (Z'_i - p) > \epsilon \right] = \mathbb{P} \left[\sum_{i=1}^n Y_i - \sum_{i=1}^n Y'_i > \epsilon \right] \leq \mathbb{P} \left[\left(\sum_{i=1}^n Y_i > \frac{\epsilon}{2} \right) \cup \left(-\sum_{i=1}^n Y'_i > \frac{\epsilon}{2} \right) \right] \quad (\text{B.2.2})$$

The last inequality comes from the fact that in order for the event $\sum_{i=1}^n Y_i - \sum_{i=1}^n Y'_i > \epsilon$ to hold, it is necessary that either $\sum_{i=1}^n Y_i > \frac{\epsilon}{2}$ or $\sum_{i=1}^n (-Y'_i) > \frac{\epsilon}{2}$ is satisfied. However, it is only a necessary, but not sufficient condition and hence the inequality.

By using the union bound, we have

$$\mathbb{P} \left[\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n Z'_i > \epsilon \right] \leq \mathbb{P} \left[\left(\sum_{i=1}^n Y_i > \frac{\epsilon}{2} \right) \right] + \mathbb{P} \left[\left(\sum_{i=1}^n -Y'_i > \frac{\epsilon}{2} \right) \right] \leq 2 \exp \left(-2n \left(\frac{\epsilon}{2} \right)^2 \right) = 2 \exp \left(-\frac{n\epsilon^2}{2} \right)$$

□

B.3. Proof of Lemma 2.2.4

Lemma. (*Sauer's Lemma*) *Let \mathcal{H} be a hypothesis set of VC-dimension d . Then for all $n \in \mathbb{N}$, the following relation holds*

$$\Pi_{\mathcal{H}}(n) \leq \sum_{i=0}^d \binom{n}{i} \tag{B.3.1}$$

PROOF. The proof is by induction in both dataset size, n , and VC-dimension. Intuitively, we show that the restriction of any hypothesis class \mathcal{H} to a dataset S , i.e., \mathcal{H}_S , can be decomposed into the restriction of two different subclasses of \mathcal{H} to a subset of the data with one data point less than the initial data set. This decomposition is constructed in such a way that on the one hand, the VC-dimensions of these two subclasses are upper-bounded by d and $d - 1$, and on the other hand, the union of the two smaller restrictions is equal to \mathcal{H}_S . For the induction step, we use the following identity

$$\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1} \tag{B.3.2}$$

According to this short description of the proof, we need two base cases as $n = 1, d = 1$ and $n = 1, d = 0$. For both cases we see that the inequality in Sauer's lemma holds. The induction assumption is that the inequality in Equation (B.3.1) is valid for the case with the dataset size $m - 1$ and VC-dimension d as well as the case with the dataset size $m - 1$ and VC-dimension $d - 1$. If we then show that this assumption results in the validity of the result for the case of dataset size m and VC-dimension d , then the lemma is proved.

Let's consider the restriction of \mathcal{H} to $S = \{x_1, \dots, x_m\}$. Also, consider all sign patterns realized by this class \mathcal{H} over data points $S' = \{x_1, \dots, x_{m-1}\}$. Now we construct two different (representative) subsets of \mathcal{H} . The first one includes a representative function h for any possible sign pattern realized by \mathcal{H} on S' . That is, its cardinality is equal to the number of all possible sign patterns on S' formed by the hypothesis class \mathcal{H} . We call this subset \mathcal{H}_1 . Then, we form \mathcal{H}_2 as all representative functions in \mathcal{H} that are not included in \mathcal{H}_1 . This means each members of \mathcal{H}_2 has a counterpart in \mathcal{H}_1 , which results in the same labelling of data points in S' , but a different label on x_m . Therefore, the cardinality of the restriction of \mathcal{H} to S is the sum of the cardinality of the restriction of \mathcal{H}_1 to S' and the cardinality of the

restriction of \mathcal{H}_2 to S' .

$$|\mathcal{H}_S| = |\mathcal{H}_{1,S'}| + |\mathcal{H}_{2,S'}| \quad (\text{B.3.3})$$

Also, by construction we have $|\mathcal{H}_{1,S'}| \geq |\mathcal{H}_{2,S'}|$. Another important observation is that since any labeling realized by \mathcal{H}_2 on S' has a counterpart in \mathcal{H}_1 which gives a different label for x_m , if a subset $S_{shattered} \subset S'$ is shattered by \mathcal{H}_2 , this implies $S_{shattered} \cup \{x_m\}$ is shattered by \mathcal{H} . From that, we have the inequality $d_{\text{VC}}(\mathcal{H}_2) \leq d_{\text{VC}}(\mathcal{H}) - 1 = d - 1$. On the other hand, since \mathcal{H}_1 is included in \mathcal{H} , we have $d_{\text{VC}}(\mathcal{H}_1) \leq d_{\text{VC}}(\mathcal{H}) = d$.

Now, applying the induction assumption on \mathcal{H}_2 over the set S' , we have

$$\Pi_{\mathcal{H}_2}(n-1) \leq \sum_{i=0}^{d-1} \binom{n-1}{i} \quad \text{and} \quad \Pi_{\mathcal{H}_1}(n-1) \leq \sum_{i=0}^d \binom{n-1}{i} \quad (\text{B.3.4})$$

Replacing these upper bounds into Equation (B.3.3) we get

$$\Pi_{\mathcal{H}}(n) \leq \sum_{i=0}^{d-1} \binom{n-1}{i} + \sum_{i=0}^d \binom{n-1}{i} = \sum_{i=1}^d \binom{n-1}{i-1} + \sum_{i=1}^d \binom{n-1}{i} + \binom{n-1}{0} = \sum_{i=0}^d \binom{n}{i}, \quad (\text{B.3.5})$$

where we have used the identity in Equation (B.3.2) as well as the fact that $\binom{n-1}{0} = \binom{n}{0} = 1$. \square

Appendix C

VC-dimension of Half-Spaces

In this section, we review the calculation of the VC-dimension for a prevalent class of hypotheses known as half-spaces. Consider data points with d number of features that reside in the d -dimensional real space \mathbb{R}^d . The points are labeled either 1 or -1 and we are concerned with the classification problem of these points. The class of hypotheses that we consider is the set of hyperplanes of dimension $d - 1$. These subspaces are called half-spaces. Now, the question is: what is the maximum number of dichotomies or sign patterns that can be realized for a dataset of size n under this class, i.e., the growth function of the set of half-spaces. This is tightly related with the VC-dimension of this hypothesis class.

Here, we consider the simpler case where half-spaces are constrained to pass through the origin. This is the homogeneous case. Using the rules of linear algebra, we show that the VC-dimension of homogeneous half-spaces of the space \mathbb{R}^d is equal to d . For that, we should first show that there always exists at least one dataset of d points which is shattered by homogeneous half-spaces. Secondly, we have to prove that no dataset of $d + 1$ points can be shattered by this class of functions.

In terms of linear algebra language, the set of hyperplane-classifiers in \mathbb{R}^d is defined as

$$\mathcal{H} = \left\{ h : \mathbf{x} \mapsto (\text{sign}(\mathbf{a} \cdot \mathbf{x})) \mid \mathbf{a} \in \mathbb{R}^d \right\} \quad (\text{C.0.1})$$

which from the geometry perspective is the set of all hyperplanes passing through the origin with normal vectors proportional to \mathbf{a} . To show that there exists at least one dataset of d points shattered by this class of hypotheses, we consider the canonical basis of the d -dimensional real space as the data points. We represent this dataset as $P = \{p_1, \dots, p_d\}$ with the following coordinate representation

$$\mathbf{p}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \dots, \quad \mathbf{p}_d = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (\text{C.0.2})$$

Now, let's consider the restriction of the half-space class defined in Equation (C.0.1) to this set of data points, denoted by \mathcal{H}_P . This is given by

$$\mathcal{H}_P = \{(\text{sign}(a_1), \text{sign}(a_2), \dots, \text{sign}(a_d)) \mid a_1, \dots, a_d \in \mathbb{R}\} \quad (\text{C.0.3})$$

with a_1, \dots, a_d , being the d components of the vector \mathbf{a} . From the above relation, it is clear that there are 2^d possibilities for the sign patterns of these points and in order to get any of the two possible signs for each data point \mathbf{p}_i , we only need to let the corresponding coefficient a_i have that same sign (without affecting the sign of the other data points). Therefore, we have shown that the VC-dimension of half-spaces of Equation (C.0.1) is at least d .

Next, we should show that no $d + 1$ points are shattered by this same hypothesis class. To keep the analysis general and inclusive, we write the coordinates of the i^{th} data point as $\mathbf{p}_i = [x_1^{(i)} \ x_2^{(i)} \ \dots \ x_d^{(i)}]^T$. Since in d -dimensional space we have at most d linearly independent vectors, the $(d + 1)$ -th point is a linear combination of the other d ones; so we write it as $\mathbf{p}_{d+1} = \alpha_1 \mathbf{p}_1 + \dots + \alpha_d \mathbf{p}_d$. Also, we define a function $f : \mathbf{x} \mapsto \mathbf{a} \cdot \mathbf{x}$. Then we consider the value of the function class on the point \mathbf{p}_{d+1} which gives $\text{sign}(\mathbf{a} \cdot \mathbf{p}_{d+1}) = \text{sign}(\alpha_1 \mathbf{a} \cdot \mathbf{p}_1 + \dots + \alpha_d \mathbf{a} \cdot \mathbf{p}_d) = \text{sign}(\alpha_1 f(\mathbf{p}_1) + \dots + \alpha_d f(\mathbf{p}_d))$.

A given \mathbf{p}_{d+1} is associated with unique values for $\alpha_1, \dots, \alpha_d$. For that set of α_i 's, any combination of $f(\mathbf{p}_1), \dots, f(\mathbf{p}_d)$, results in either $f(\mathbf{p}_{d+1}) = \alpha_1 f(\mathbf{p}_1) + \alpha_2 f(\mathbf{p}_2) + \dots + \alpha_d f(\mathbf{p}_d) > 0$ or $f(\mathbf{p}_{d+1}) = \alpha_1 f(\mathbf{p}_1) + \alpha_2 f(\mathbf{p}_2) + \dots + \alpha_d f(\mathbf{p}_d) < 0$. The fact that the dichotomy corresponding to the same combination of $f(\mathbf{p}_1), \dots, f(\mathbf{p}_d)$, cannot give rise to both $f(\mathbf{p}_{d+1}) > 0$ and $f(\mathbf{p}_{d+1}) < 0$ means that, the dataset of size $d + 1$ cannot be shattered by this class of functions and hence $d \leq d_{\text{VC}} < d + 1$. We conclude that for half-spaces, $d_{\text{VC}} = d$.

Appendix D

Lower Bounds on the Number of Sign Patterns

D.1. Main Results

We saw in Chapter 3, that for tensor trains with some constraints ($r = d$ and $p \bmod 3 = 0$), the upper-bound on the VC-dimension is tight up to the logarithmic factor $\log p$. We guess that this tightness should be the case for more general setups. Here, we want to give an evidence for this claim. This is based on an approach that the number of sign patterns realized by a hypothesis class, rather than its VC-dimension. Since lower-bounding the VC-dimension from the number of sign patterns does not seem straightforward, we did not discuss this approach in Chapter 3. However, the technicality involved in this method as well as the evidence it provides, make it worth mentioning it here.

D.1.1. Lower-bound on the Number of Sign Patterns of Low-rank Matrices [4]

We start by the low-rank matrix case. Let \mathbf{M} be a $m \times n$ matrix of rank k , so that it has a rank decomposition as illustrated in Figure D.1, with the left and right low-rank matrices called \mathbf{L}_k and \mathbf{R}_k .

Our goal is to find a lower bound on the number of sign patterns of \mathbf{M} given that it has this low-rank decomposition. We make some key observations. First, Each of the matrices \mathbf{L}_k and \mathbf{R}_k are free to take any values. That is, we can take \mathbf{L}_k to be quite arbitrary, and \mathbf{R}_k to be in *general position*.

Definition 21. *A set of vectors in \mathbb{R}^d is in general position if and only if every subset of exactly d vectors is linearly independent.*

Lemma D.1.1. *There are exactly $2 \sum_{k=0}^{d-1} \binom{N-1}{k}$ homogeneously linearly separable sign patterns of N points in general position in \mathbb{R}^d .*

PROOF. The proof of this lemma can be found in Section D.2.3. □

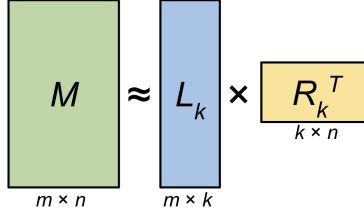


Figure D.1. Figure from [9]

Now, the above decomposition can be interpreted as follows: let us think of the n columns of the matrix \mathbf{R}_k as n points in \mathbb{R}^k , then each row of the matrix \mathbf{L}_k can be seen as a linear classifier of these n points. Therefore, Lemma D.1.1 says that for each classifier, i.e., each row of \mathbf{L}_k , we have exactly the following number of sign patterns (SP)

$$\#\text{SP}(\text{each row}) = 2 \sum_{i=0}^{k-1} \binom{n-1}{i}$$

Now, if we consider low-rank matrices for which $n > k$, we get

$$\#\text{SP}(\text{each row}) \geq \binom{n-1}{k-1} > \left(\frac{n-1}{k-1}\right)^{k-1}$$

Having this many sign patterns by each row of \mathbf{L}_k , the sign patterns realized by the whole matrix \mathbf{L}_k on the n points of \mathbf{R}_k are at least as many as

$$\left(\left(\frac{n-1}{k-1}\right)^{k-1}\right)^m \sim \left(\frac{n}{k}\right)^{mk}$$

Now, if we further constrain the matrix rank as $k^2 < n$, the above lower bound simplifies to

$$\#\text{SP}(\text{each row}) \geq n^{\frac{mk}{2}}$$

By comparing this lower bound with the upper bound coming from Warren's theorem 9, i.e., with $\#\text{SP} \leq \left(\frac{8e \cdot 2 \cdot nm}{k(n+m)}\right)^{k(n+m)} \leq \left(\frac{16en}{k}\right)^{k(n+m)}$, we observe the tightness of the upper bound up to a multiplicative factor in the exponent, for the case of $n > k^2$.

D.1.2. Lower-bound on the Number of Sign Patterns of Tensor Trains

Analogously to the above study for low-rank matrices, we can consider other tensor networks, like tensor train.

For the tensor completion and classification tasks with a tensor $\mathcal{G} \in \mathbb{R}^{d \times d \times d \times d}$ in TT representation, as $\mathcal{G}^1 - \mathcal{G}^2 - \mathcal{G}^3 - \mathcal{G}^4$, if we can find an optimal way of breaking the tensor into two parts, one which we can claim it to be in general position, and the other one that can

take any arbitrary value, then it is possible to take the same approach as in the previous part to estimate a lower bound on the number of sign patterns of the tensor.

To make it clear, let's consider a scenario for tensor completion task as illustrated in Figure D.2 part (1); the core tensor highlighted in red is assumed to take any arbitrary value. Then, the rest of the diagram needs to be in general position, so that one could do the same analysis as in the low-rank matrix case to find a lower bound on the number of dichotomies. Since this approach works based on the matricization of the tensor, in part (2) we have represented the corresponding matricization.



Figure D.2. (1) A 4-th order tensor train \mathcal{G} with the core highlighted in red being free to take any arbitrary values. (2) Mode- n Matricization of the same tensor \mathcal{G} w.r.t. the mode corresponding to the highlighted core.

We first consider the diagram on the left with \mathcal{G}_1 taking arbitrary values. Figure D.3 shows the rest of the diagram, which we will call *broken TT*.



Figure D.3. Part (1) The broken TT of Figure D.2. Part (2) illustrates the matricization of the broken TT.

Now, we claim that the matricization of the broken TT can be put in general position; that is, the broken TT can be seen as d^3 points in *general position* in \mathbb{R}^r . A constructive proof is provided in Appendix D.2.1 for the special case of $r = 2$. For general $r \leq d$, a constructive proof based on *moment curves* is given in Appendix D.2.2.

The upshot is that the first matricization in part (2) of Figure D.2 is the matrix product of an arbitrary matrix \mathcal{G}_1 of size $d \times r$ by a matrix G of size $r \times d^3$, interpreted as stacking d^3 points in general positions in \mathbb{R}^r . Each row of \mathcal{G}_1 as a homogeneous linear classifier on these points produces $2 \sum_{i=0}^{r-1} \binom{d^3-1}{i}$ dichotomies which gives the following lower bound on the total number of such dichotomies, when taking into account all independent rows of \mathcal{G}_1

$$\#\text{dichotomies} \geq \left(\frac{d^3 - 1}{r - 1} \right)^{d(r-1)} \quad (\text{D.1.1})$$

This result straightforwardly generalizes to higher-order tensors, giving the following lower bound on the number of sign patterns that a tensor train of arbitrary order p with a uniform TT-rank $r \leq d$ can take

$$\#\text{dichotomies} \geq \left(\frac{d^{p-1} - 1}{r - 1} \right)^{d(r-1)} \sim d^{(p-2)(r-1)d} \quad (\text{D.1.2})$$

This could be interpreted as an evidence for the possibility of tightening our lower bound on the VC-dimension of TT models that we have shown in Table 3.1 for the case $r < d$.

D.2. Proofs for Section D.1

To review the calculation of lower bound in earlier works for tensor completion task, we explain some concepts.

D.2.1. Proof of General Position for the Ranks $r = 2$

Theorem 22. *There exist a broken TT, as defined in Figure D.3, with uniform TT-rank 2 and arbitrary order, for which the columns of the first matricization could be seen as points in general position in 2 dimensions.*

PROOF.

Lemma D.2.1. *Points on a semi-circle in two dimensions, are in general position in the sense of Definitio 21.*

Let us call the broken TT, $\mathcal{G}^{>1}$, and assume that $\mathcal{G}^{>1}$ is of order n . Consider one element of $\mathcal{G}^{>1}$ as $\mathcal{G}_{r_1, i_2, \dots, i_n}^{>1} \in \mathbb{R}^{2 \times d_2 \times \dots \times d_n}$, with r_1 associated with the broken bond dimension and the rest of the indices corresponding to all physical legs. By construction, this tensor element is written in terms of the core tensors (matrices) as below

$$\mathcal{G}_{r_1, \dots, i_n}^{>1} = \sum_{r_2, \dots, r_{n-1}} \mathcal{G}_{r_1, i_2, r_2}^2 \mathcal{G}_{r_2, i_3, r_3}^3 \cdots \mathcal{G}_{r_{n-1}, i_n}^n \quad (\text{D.2.1})$$

with $\{\mathcal{G}_{:, i_k, :}^k\}_{k=2}^{n-1}$, being matrices of dimension 2×2 and $\mathcal{G}_{:, i_n}^n$ a 2-d vector. Next, we define $d_2 \cdots d_n$ different polar angles as below

$$\begin{aligned} \theta_1^{(2)} &= \frac{2\pi}{d_2}, \theta_2^{(2)} = 2\frac{2\pi}{d_2}, \dots, \theta_{d_2}^{(2)} = d_2 \frac{2\pi}{d_2} \\ \theta_1^{(3)} &= \frac{2\pi}{d_2 d_3}, \theta_2^{(3)} = 2\frac{2\pi}{d_2 d_3}, \dots, \theta_{d_3}^{(3)} = d_3 \frac{2\pi}{d_2 d_3} \\ &\dots \\ \theta_1^{(n)} &= \frac{2\pi}{d_2 d_3 \cdots d_n}, \theta_2^{(n)} = 2\frac{2\pi}{d_2 d_3 \cdots d_n}, \dots, \theta_{d_n}^{(n)} = d_n \frac{2\pi}{d_2 d_3 \cdots d_n} \end{aligned} \quad (\text{D.2.2})$$

Next, we construct $\{\mathcal{G}_{:, i_k, :}^k\}_{k=2}^{n-1}$ in terms of $\theta_j^{(2)}$'s up to $\theta_j^{(n-1)}$'s as below

$$\mathcal{G}_{:, i_k, :}^k = \begin{pmatrix} \cos \theta_{i_k}^{(k)} & \sin \theta_{i_k}^{(k)} \\ -\sin \theta_{i_k}^{(k)} & \cos \theta_{i_k}^{(k)} \end{pmatrix} \quad (\text{D.2.3})$$

and vector $\mathcal{G}_{:,i_n}^n$ as $\mathcal{G}_{:,i_n}^n = \begin{pmatrix} \sin \theta_{i_n}^{(n)} & \cos \theta_{i_n}^{(n)} \end{pmatrix}$. From the following matrix identities

$$\begin{aligned} \sin(\alpha + \beta) &= \sin \alpha \cos \beta + \cos \alpha \sin \beta \\ \cos(\alpha + \beta) &= \cos \alpha \cos \beta - \sin \alpha \sin \beta \end{aligned}$$

or equivalently

$$\begin{pmatrix} \sin(\alpha + \beta) \\ \cos(\alpha + \beta) \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} \sin \beta \\ \cos \beta \end{pmatrix}$$

it is easily verified that $\sum_{r_{n-1}} \mathcal{G}_{:,i_{n-1},r_{n-1}}^{n-1} \mathcal{G}_{r_{n-1},i_n}^n = \begin{pmatrix} \sin(\theta_{i_n}^{(n)} + \theta_{i_{n-1}}^{(n-1)}) & \cos(\theta_{i_n}^{(n)} + \theta_{i_{n-1}}^{(n-1)}) \end{pmatrix}^T$. This analysis is easily generalized to the whole TT and leads to the following result for every element $:,i_2, \dots, i_n$ of the broken TT

$$\sum_{r_2, \dots, r_{n-1}} \mathcal{G}_{:,i_2,r_3}^2 \cdots \mathcal{G}_{:,i_{n-1},r_{n-1}}^{n-1} \mathcal{G}_{r_{n-1},i_n}^n = \begin{pmatrix} \sin(\theta_{i_n}^{(n)} + \theta_{i_{n-1}}^{(n-1)} + \cdots + \theta_{i_2}^{(2)}) & \cos(\theta_{i_n}^{(n)} + \theta_{i_{n-1}}^{(n-1)} + \cdots + \theta_{i_2}^{(2)}) \end{pmatrix}^T \quad (\text{D.2.4})$$

which also comes from the fact that consecutive rotations of a point by several angles is equivalent to one rotation by the sum of those angles. This means each vector $\mathcal{G}_{:,i_2, \dots, i_n}$ is interpreted as a point on a unit circle centered at the origin at the radial angle $\theta_{i_n}^{(n)} + \theta_{i_{n-1}}^{(n-1)} + \cdots + \theta_{i_2}^{(2)}$. The point is that with the choice of angles as in Equation (D.2.2) no two angles out of all $d_2 d_3 \cdots d_n$ are the same; hence, all points are uniformly placed around the unit circle and according to Lemma D.2.1 are in general position. \square

D.2.2. Proof of General Position based on Moment Curve for Tensor Train

Proposition D.2.2. *For any dimension d_2, \dots, d_p and any rank R , there exist $\{\mathcal{G}^{(k)} \in \mathbb{R}^{R \times d_k \times R}\}_{k=2}^{p-1}$ and $\mathcal{G}^{(p)} \in \mathbb{R}^{R \times d_p}$, such that the $d_2 \cdots d_p$ points in \mathbb{R}^R defined by*

$$\mathbf{x}_{i_2, \dots, i_p} = \mathcal{G}_{:,i_2, :}^{(2)} \cdots \mathcal{G}_{:,i_p}^{(p)} \in \mathbb{R}^R \quad \text{for } i_2 \in [d_2], \dots, i_p \in [d_p]$$

are in general position.

PROOF.

Proposition D.2.3 (Proposition 3.3 in [89]). *The points of the moment curve $\{(1, \alpha, \alpha^2, \dots, \alpha^{d-1}) \mid \alpha \in \mathbb{R}\} \subset \mathbb{R}^d$ are in general position.*

We will show that the core tensors $\mathcal{G}^{(k)}$ can be chosen in such a way that the points $\mathbf{x}_{i_2, \dots, i_p}$ correspond to distinct points on the moment curve $\{(1, \alpha, \alpha^2, \dots, \alpha^{R-1}) \mid \alpha \in \mathbb{R}\}$, the result then follows from Proposition D.2.3.

For all $k \in [2 : p]$, $j_k = 0, \dots, d_k - 1$, let $\alpha_{k, j_k} = \exp(j_k d_2 \cdots d_{k-1})$ (for $k = 2$, we take

$\alpha_{2,j_2} = \exp(j_2)$). With considerations similar to Lemma 3.4.2, one can check that the products $\alpha_{2,j_2} \cdots \alpha_{p,j_p}$ are all distinct. More precisely, we have

$$\left\{ \prod_{k=2}^p \alpha_{k,j_k} \mid j_k \in [d_k] \right\} = \{ \exp(l-1) \mid l \in [d_2 \cdots d_p] \}$$

Now, for each $k = 2, \dots, p-1$, let $\mathcal{G}^{(k)}$ be defined by

$$\mathcal{G}_{:,i_k,:}^{(k)} = \text{diag}(1, \alpha_{k,j_k}, \alpha_{k,j_k}^2, \dots, \alpha_{k,j_k}^{R-1}) \quad \text{for each } i_k \in [d_k]$$

and let \mathcal{G}^p be defined by

$$\mathcal{G}_{:,i_p}^{(p)} = \left(1, \alpha_{p,j_p}, \alpha_{p,j_p}^2, \dots, \alpha_{p,j_p}^{R-1} \right) \quad \text{for each } i_p \in [d_p], r \in [R]$$

One can check that,

$$\mathcal{G}_{:,i_2,:}^{(2)} \cdots \mathcal{G}_{:,i_p}^{(p)} = \left(1, \beta_{i_1, \dots, i_p}, \beta_{i_1, \dots, i_p}^2, \dots, \beta_{i_1, \dots, i_p}^{R-1} \right)$$

where $\beta_{i_1, \dots, i_p} = \prod_{k=2}^p \alpha_{k,j_k}$. It follows that each point x_{i_2, \dots, i_p} is given by

$$x_{i_2, \dots, i_p} = \left(1, \beta_{i_1, \dots, i_p}, \beta_{i_1, \dots, i_p}^2, \dots, \beta_{i_1, \dots, i_p}^{R-1} \right) .$$

Since all the β_{i_1, \dots, i_p} are distinct, the points x_{i_1, i_2, \dots, i_p} are distinct points on the moment curve. Then, according to Proposition D.2.3, these points are in general position. \square

D.2.3. Dichotomy Counting [5]

Lemma. *For $n > d$, for n points in general position in \mathbb{R}^d , the number of homogeneously linearly separable sign patterns is given by*

$$C(n,d) = 2 \sum_{i=0}^{d-1} \binom{n-1}{i} \tag{D.2.5}$$

PROOF. The proof is based on induction on n and d and also uses the following lemma.

Lemma D.2.4. *Consider a set of n points in \mathbb{R}^d as $X = \{x_1, x_2, \dots, x_n\}$ with a given fixed dichotomy $\{X^+, X^-\}$. Consider a new point y such that $X \cup y$ is in general position in \mathbb{R}^d . Then the dichotomies $\{X^+ \cup y, X^-\}$ and $\{X^+, X^- \cup y\}$ are both homogeneously linearly separable if and only if $\{X^+, X^-\}$ is homogeneously linearly separable by a $(d-1)$ -dim subspace containing y . For later use, we call this $(d-1)$ -dim subspace \mathcal{V} .*

Note that in the following, whenever we talk about linearly separable or just separable dichotomies, we mean homogeneously linearly separable. To prove the theorem we consider $X = \{x_1, x_2, \dots, x_n\}$ as the set of n points in general position, with $C(n,d)$ the number of its separable dichotomies. We then take x_{n+1} s.t. $X \cup x_{n+1}$ are in general position as well. Then we consider the $C(n,d)$ separable dichotomies of X . For any dichotomy $\{X^+, X^-\}$, where X^+ (X^-) is the subset of X with all members positively (Negatively) labeled, either $\{X^+ \cup$

x_{n+1}, X^- or $\{X^+, X^- \cup x_{n+1}\}$ will also be linearly separable. According to Lemma D.2.4 there are also some dichotomies for which both $\{X^+ \cup x_{n+1}, X^-\}$ and $\{X^+, X^- \cup x_{n+1}\}$ are linearly separable. Let's call the number of such dichotomies D . From the above lemma, we know that such dichotomies, when projected onto the $(d-1)$ -dim hyperplane perpendicular to the subspace \mathcal{V} defined in Lemma D.2.4, are still linearly separable; that means D is equal to $C(n, d-1)$. From this observation, a recursive relation for $C(n, d)$ follows

$$C(n+1, d) = C(n, d) - D + 2D = C(n, d) + C(n, d-1) \quad (\text{D.2.6})$$

By repeatedly applying this identity to the right-hand side of it, we get Counting [5]

$$C(n, d) = \sum_{i=0}^{n-1} \binom{n-1}{i} C(1, d-i) \quad (\text{D.2.7})$$

Now, since $C(1, t)$ vanishes for $t < 1$, the above sum reduces to $\sum_{i=0}^{d-1} \binom{n-1}{i} C(1, d-i)$. Also, since $C(1, t) = 2$ for $t \geq 1$, we get

$$C(n, d) = 2 \sum_{i=0}^{d-1} \binom{n-1}{i} \quad (\text{D.2.8})$$

□