



UNIVERSITÀ
DEGLI STUDI
FIRENZE

UNIVERSITÀ DEGLI STUDI DI FIRENZE
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)
CORSO DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE
CURRICULUM: INFORMATICA – SSD: ING-INF/05

NOVEL NEURAL NETWORKS FOR STRUCTURED DATA

Candidate

Daniele Baracchi

Supervisor

Prof. Paolo Frasconi

PhD Coordinator

Prof. Luigi Chisci

CICLO XXX, 2014-2017

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

Thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Information Engineering. Copyright © 2018 by
Daniele Baracchi.

Abstract

Complex relational structures are used to represent data in many scientific fields such as chemistry, bioinformatics, natural language processing and social network analysis. It is often desirable to classify these complex objects, a problem which is increasingly being dealt with machine learning approaches. While a number of algorithms have been shown to be effective in solving this task for graphs of moderate size, dealing with large structures still poses significant challenges due to the difficulty in scaling exhibited by the existing techniques.

In this thesis we introduce a framework to approach supervised learning problems on structured data by extending the R-convolution concept used in graph kernels. We represent a graph (or, more in general, a relational structure) as a hierarchy of objects and we define how to unroll a template neural network on it. This approach is able to outperform state-of-the-art methods on large social networks datasets, while at the same time being competitive on small chemobiological datasets. We also introduce a lossless compression algorithm for the hierarchical decompositions that improves the temporal complexity of our approach by exploiting symmetries in the input data.

Another contribution of this thesis is an application of the aforementioned framework to the context-dependent claim detection task. Claim detection is the assessment of whether a sentence contains a claim, i.e. the thesis, or conclusion, of an argument; in particular we focus on context-dependent claims, where the context (i.e. the topic of the argument) is a determining factor in classifying a sentence. We show how our framework is able to take advantage of contextual information in a straightforward way and we present some preliminary results that indicates how this approach is viable on real world datasets.

A third contribution is a machine learning approach to aortic size nor-

malcy assesment. The definition of normalcy is crucial when dealing with thoracic aortas, as a dilatation of its diameter often precedes serious disease. We build a new estimator based on OC-SVM fitted on a cohort of 1024 healty individuals aging from 5 to 89 years, and we compare its results to those obtained on the same set of subjects by an approach based on linear regression. As a further novelty, we also build a second estimator that combines the diameters measured at multiple levels in order to assess the normalcy of the overall shape of the aorta.

Contents

Contents	v
1 Introduction	1
1.1 The objective	1
1.2 Contributions	2
I Relational learning	5
2 Background	7
2.1 Graphs	7
2.2 Machine learning	8
2.2.1 Support Vector Machines	9
2.2.2 Neural networks	12
2.3 Related works	14
2.3.1 Graph kernels	14
2.3.2 Neural networks on graphs	15
3 Shift aggregate extract networks	17
3.1 Introduction	17
3.2 \mathcal{H} -decompositions	19
3.3 Instances of \mathcal{H} -decompositions	22
3.3.1 Ego Graph Decomposition	22
3.3.2 Nested Ego Graph Decomposition	23
3.4 Learning representations with SAEN	23
3.5 Exploiting symmetries for domain compression	26
3.6 Experimental evaluation	33
3.6.1 Datasets	33

3.6.2	Experiments	35
3.6.3	Discussion	38
3.7	Related works	39
3.8	Conclusions	41
4	Context-dependent claim detection using SAEN	43
4.1	Introduction	43
4.2	Argumentation Mining	46
4.3	Context-aware hierarchical decomposition	47
4.4	Experimental Evaluation	48
4.4.1	Datasets	50
4.4.2	Experiments and results	51
4.5	Conclusions	52
II	Novelty detection in cardiology	55
5	A machine learning approach to assess normalcy of aortic size	57
5.1	Introduction	57
5.2	Materials and methods	60
5.2.1	Study population	60
5.2.2	Echocardiography	61
5.2.3	Regression analysis model	62
5.2.4	One-class support vector machine model	62
5.2.5	ROC analysis	65
5.3	Results	66
5.3.1	Characteristics of normalcy and deemed-at-risk cohorts	66
5.3.2	ROC analysis by linear regression model and OC-SVM model	70
5.3.3	Performance of linear regression and OC-SVM models in deemed-at-risk individuals	70
5.4	Discussion	71
6	Conclusions	75
6.1	Summary of contribution	75
6.2	Directions for future work	76
A	Publications	79

Bibliography

81

Chapter 1

Introduction

The amount of information produced by the human race saw an unprecedented growth in recent years, thanks to the ubiquity of devices capable of generating new data such as computers, smartphones, and connected devices. Many important tasks in science and technology involve analyzing and classifying this information, but employing human beings to make sense of this amount of data is often infeasible. Fortunately, the same advances in technology that allowed us to generate this data also brought us a marked improvement in computational power, enabling the development of algorithms that can automatically learn from data, i.e. that can *machine learn*.

1.1 The objective

Most machine learning algorithms are designed to handle independent objects, but real world data can often be understood only when also considering the relationships between them. As a consequence, in the last few years many researchers started to study how to tackle the problem of learning on structured (or relational) data, which is usually represented as graphs. Graph kernels decompose graphs into simpler substructures and compute similarity between structured objects as the similarity of their sets of parts. Recursive neural networks, on the other hand, exploit the graph's structure to build a vector representation for a node by composing the representations of its neighbors.

Our goal is to build a method to learn representations for structured data by combining the strengths of both families of algorithms, and more

specifically by using both the decomposition approach of graph kernels and the iterative vector representation building approach of neural networks.

1.2 Contributions

This thesis makes two contributions to relational learning by proposing a novel method to apply neural networks to structured data and a possible application to a natural language processing task. A third contribution involves an application of novelty detection in cardiology.

- The *first contribution* introduces a framework to approach supervised learning problems on structured data by extending the \mathcal{R} -convolution concept used in most graph kernels [38]. We describe how to represent a relational structure as a hierarchy of objects and we define how to unroll a template neural network on it. This approach is able to outperform state-of-the-art methods on large social networks datasets, while at the same time being competitive on small chemobiological datasets. We also introduce a lossless compression algorithm for the hierarchical decompositions that improves the temporal and spatial complexities of our approach by exploiting symmetries in the input data.
- The *second contribution* describes an application of the aforementioned method to context-dependent claim detection task. Claim detection is the assessment of whether a sentence contains a claim, i.e. the thesis, or conclusion, of an argument; in particular we focus on context-dependent claims, where the context (i.e. the topic of the argument) is a determining factor in classifying a sentence. We show how our framework is able to take advantage of contextual information in a straightforward way and we present some preliminary results that indicates how this approach is viable on real world datasets.
- The *third contribution* is a machine learning approach to aortic size normalcy assesment. The definition of normalcy is crucial when dealing with thoracic aortas, as a dilatation of its diameter often precedes serious disease. A number of methods have been developed to obtain the expected aortic diameter starting from demographic and anthropometric characteristics such as height, weight, age and sex. Unfortunately, these methods suffer from several limitations: first, they are

suitable either only for infants, children, and young adults, or only for adults; and second, they make strong assumptions on the relationship between predictors and aortic diameters. In this contribution we overcome these limitations by building a new estimator based on OC-SVM [77] fitted on a cohort of 1024 healthy individuals aging from 5 to 89 years, and we compare its results to those obtained on the same set of subjects by an approach based on linear regression. We also build a second estimator that combines the diameters measured at multiple levels in order to assess the normalcy of the overall shape of the aorta.

Part I

Relational learning

Chapter 2

Background

This chapter gives a brief introduction to essential concepts used in this thesis. We start by introducing some notation on graphs that we will use. Then, we describe the main techniques used in machine learning. Finally, we review some related work on relational learning.

2.1 Graphs

Most relational data can be easily expressed in terms of graphs as they are, in their simplest form, sets of objects connected by some kind of relationship.

Definition 1 (Graph). *A graph is a pair of sets $G = (V, E)$ where V is the set of vertices (or nodes), and $E \subseteq V \times V$ is the set of edges.*

Definition 2 (Directed and undirected graph). *A graph $G = (V, E)$ is directed if E is a set of ordered pairs of vertices. G is undirected if E is a set of unordered pairs of vertices, or equivalently if $(v_i, v_j) \in E \iff (v_j, v_i) \in E$.*

As we shall see, it is often useful to examine the substructures contained in a given graph. Among the most commonly used substructures we find neighborhoods and paths.

Definition 3 (Neighborhood). *Given a graph $G = (V, E)$ and a vertex $v \in V$, the neighborhood of v in G is the set of nodes $\{w : (v, w) \in E\}$.*

Definition 4 (Path, connected graph). *Given a graph $G = (V, E)$, a path $\pi = (\pi_1, \dots, \pi_n)$, is a sequence of vertices $\pi_i \in V$, $i = 1, \dots, n$ such as*

$(\pi_i, \pi_{i+1}) \in E$, $i = 1, \dots, n - 1$. G is said to be connected if for any two nodes $u, v \in V$ there exists a path $\pi = (\pi_1, \dots, \pi_n)$ such as $\pi_1 = u, \pi_n = v$ or $\pi_1 = v, \pi_n = u$.

A useful concept when we need to capture the influence of one individual object inside a graph (e.g. when dealing with graphs that represent social networks) is the ego graph. Intuitively, an ego graph rooted in one object o describe the portion of the original graph that directly interacts with o .

Definition 5 (Induced subgraph). *The subgraph of $G = (V, E)$ induced by $V_g \subset V$ is the graph $g = (V_g, E_g)$ where $E_g = \{(u, v) \in E : u \in V_g, v \in V_g\}$.*

Definition 6 (Ego graph). *The ego graph $g_{v,r}$ of $G = (V, E)$ with root $v \in V$ and radius r is the subgraph of G induced by the set of vertices whose shortest path distance from v is at most r .*

2.2 Machine learning

Machine learning is the subfield of artificial intelligence that studies algorithms to enable computers to learn from data, or more specifically to infer an unknown function from a set of objects (*examples*). In this thesis we will focus on supervised learning, where we want to learn a function $f : X \rightarrow Y$ from a set of labeled examples $T = \{(x_i, y_i)\}_{i=1}^N \subset X \times Y$, $y_i = f(x_i) \forall i$.

We can see supervised learning tasks as optimization problems. If we call $\hat{f} : X \rightarrow Y$ the inferred function, and $L : Y \times Y \rightarrow \mathbb{R}$ a function estimating the error (*loss*) between the expected and the actual outputs, we can express our supervised learning problem as

$$\underset{\hat{f}}{\text{minimize}} L\left(f(x), \hat{f}(x)\right) \quad x \in X. \quad (2.1)$$

As it is impractical to minimize over every possible function, supervised learning algorithms usually restrict the search field to a class of functions depending on some parameter θ . As mentioned earlier we only have a limited set T of labelled examples, so the problem becomes

$$\underset{\theta}{\text{minimize}} L\left(y_i, \hat{f}(x_i; \theta)\right) \quad (x_i, y_i) \in T. \quad (2.2)$$

In this thesis we will focus on *classification* tasks, i.e. supervised learning problems where the unknown function f maps the examples to a finite set of classes $Y = \{c_1, \dots, c_M\}$.

2.2.1 Support Vector Machines

Support Vector Machines [89] are supervised learning models commonly used for classification problems. The basic idea of linear SVM is to find an hyperplane in a high-dimensional feature space capable of separating examples belonging to different classes. Unseen examples will be then classified based on their position relative to the hyperplane found during the training phase. For a given set of data, infinite separating hyperplanes could exist; SVM searches for the maximum-margin one, i.e. the one with maximum distance from the nearest examples.

Let T be a set of training examples \mathbf{x}_i with labels y_i :

$$T = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}. \quad (2.3)$$

Samples belonging to the first class will be labeled with $y = 1$, while those belonging to the second class will be labeled with $y = -1$. The set is said to be linearly separable if there exist a vector \mathbf{w} and a scalar b such that

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 0 \quad \text{if } y_i = 1, \quad (2.4)$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \quad \text{if } y_i = -1, \quad (2.5)$$

i.e. if a hyperplane able to separate the elements belonging to the two classes exists. Note that, without loss of generality, it is possible to scale \mathbf{w} and b to obtain

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \quad \text{if } y_i = 1, \quad (2.6)$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1, \quad (2.7)$$

$$(2.8)$$

that can be rewritten in compact form as

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall \mathbf{x}_i \in T. \quad (2.9)$$

Given a separating hyperplane, unseen examples \mathbf{x} can be classified based on the sign of the decision function

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b. \quad (2.10)$$

For each linearly separable set of examples an infinite number of separating hyperplanes could exist. The SVM training algorithm searches for

the one with the largest *margin* between the separation boundary and the training patterns. Knowing that the distance between an hyperplane and a generic pattern \mathbf{x} is

$$\frac{\mathbf{w} \cdot \mathbf{x} + b}{\|\mathbf{w}\|} \quad (2.11)$$

and assuming a separating hyperplane with margin M , then

$$\frac{y_k (\mathbf{w} \cdot \mathbf{x}_k + b)}{\|\mathbf{w}\|} \geq M \quad (2.12)$$

holds for all $(\mathbf{x}_k, y_k) \in T$. The problem can thus be written as

$$\underset{\mathbf{w}, \|\mathbf{w}\|=1}{\text{maximize}} \quad M \quad (2.13)$$

$$\text{subject to} \quad y_k (\mathbf{w} \cdot \mathbf{x}_k - b) \geq M \quad \forall (\mathbf{x}_k, y_k) \in T, \quad (2.14)$$

where the norm of \mathbf{w} is fixed to remove the ambiguity between solutions that differ only in scaling. The same result can be achieved by fixing the product $M\|\mathbf{w}\| = 1$, which leads to reformulate the problem as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.15)$$

$$\text{subject to} \quad y_k (\mathbf{w} \cdot \mathbf{x}_k - b) \geq 1 \quad \forall (\mathbf{x}_k, y_k) \in T. \quad (2.16)$$

We can now derive the solution of the optimization problem by using Karush-Kuhn-Tucker necessary conditions [41, 45] for optimal solutions. Given a nonlinear optimization problem

$$\underset{x}{\text{minimize}} \quad f(x), \quad (2.17)$$

$$\text{subject to} \quad g_i(x) \leq 0 \quad (2.18)$$

any optimal solutions x^* must satisfy

$$-\nabla f(x^*) = \sum_i \mu_i \nabla g_i(x^*); \quad (2.19)$$

$$\mu_i g_i(x^*) = 0. \quad (2.20)$$

By applying (2.19) and (2.20) to (2.15) and (2.16) we obtain

$$\mathbf{w}^* = \sum_i \mu_i y_i \mathbf{x}_i; \quad (2.21)$$

$$0 = \mu_i (1 - y_i (\mathbf{w}^* \cdot \mathbf{x}_i - b)). \quad (2.22)$$

We notice from (2.22) that $\mu_i \neq 0$ only for \mathbf{x}_i lying on the margin (i.e. patterns satisfying $y_i(\mathbf{w}^* \cdot \mathbf{x}_i - b) = 1$), called *support vector*. Let now be S the set of support vectors. We can rewrite (2.21) as

$$\mathbf{w}^* = \sum_{\mathbf{x}_i \in S} \mu_i y_i \mathbf{x}_i. \quad (2.23)$$

From (2.22) we can also derive, for $\mathbf{x}_i \in S$, the offset b :

$$\mathbf{w}^* \cdot \mathbf{x}_i - b = \frac{1}{y_i} = y_i \quad (2.24)$$

$$b = \mathbf{w}^* \cdot \mathbf{x}_i - y_i. \quad (2.25)$$

Soft margin

The SVM algorithm seen above will not work when the examples are not linearly separable and thus cannot deal with mislabeled patterns. The *soft margin* method [14] handles these cases by finding an hyperplane that splits the examples as cleanly as possible. A set of *slack* variables that allow for a certain degree of misclassification are introduced in the optimization constraint and objective leading to the formulation

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_k \zeta_k \quad (2.26)$$

$$\text{subject to} \quad y_k (\mathbf{w} \cdot \mathbf{x}_k - b) \geq 1 - \zeta_k \quad \forall (\mathbf{x}_k, y_k) \in T \quad (2.27)$$

$$\zeta_k \geq 0 \quad \forall k. \quad (2.28)$$

The slack variables in (2.27) allow for samples on the wrong side of the split, while in (2.26) the same variables are bound to be “small” by the optimization algorithm. The C parameter controls the tradeoff between the number of mislabeled examples and the size of the margin between the hyperplane and the correctly classified examples.

Kernels

While the original SVM algorithm was limited to (almost) linearly separable data, it can be extended to work as a nonlinear classifier by applying the *kernel trick* [10].

It can be proved that the dual form of the optimization problem (which

leads to the same solution of the primal form) described by (2.26)-(2.28) is

$$\underset{\mu}{\text{maximize}} \quad \sum_k \mu_k - \frac{1}{2} \sum_{i,j} \mu_i \mu_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (2.29)$$

$$\text{subject to} \quad 0 \leq \mu_k \leq C \quad \forall k \quad (2.30)$$

$$\sum_k \mu_k y_k = 0. \quad (2.31)$$

Moreover, by combining (2.10) with (2.23) we obtain an equivalent decision function

$$D(\mathbf{x}) = \sum_{\mathbf{x}_i \in S} \mu_i y_i \mathbf{x}_i \cdot \mathbf{x} + b. \quad (2.32)$$

These transformations are important because they let us write both the training algorithm and the subsequent decision function only in terms of inner products between patterns.

Suppose now that there exists a function $\phi : S \rightarrow V$ that maps the patterns from source space S to a space V where the examples are more easily separable and where an inner product $\langle \cdot, \cdot \rangle_V$ is defined. We could then apply the SVM algorithm to the transformed patterns in the destination space by using the objective function

$$\underset{\mu}{\text{maximize}} \quad \sum_k \mu_k - \frac{1}{2} \sum_{i,j} \mu_i \mu_j y_i y_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_V \quad (2.33)$$

during training and the decision function

$$D(\mathbf{x}) = \sum_{\mathbf{x}_k \in S} \mu_k y_k \langle \phi(\mathbf{x}_k), \phi(\mathbf{x}) \rangle_V + b. \quad (2.34)$$

It's often convenient to describe the algorithm in term of a *kernel* function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_V \quad (2.35)$$

instead of explicit space maps and inner products. This let us work with high or even infinite-dimensional feature spaces without actually having to compute pattern projections.

2.2.2 Neural networks

Artificial neural networks use a mathematical model inspired by biological neurons to represent computations. Given an input vector $\mathbf{x} \in \mathbb{R}^n$, the output y of a generic artificial neuron is produced by applying a non-linear function f to a weighted sum of the components of x , i.e. $y = f(\sum_i w_i x_i + b)$. A

common choice for activation function is the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$, which produces outputs between 0 and 1 and is therefore suitable for representing probabilities. A more practical function [46] is the hyperbolic tangent $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, which produces outputs between -1 and 1 , making it more likely to generate outputs that on average are close to zero, and so speeding up the convergence when these outputs are used as inputs of other units. Another common choice for activation function is the *rectified linear activation* [31] $\text{relu}(x) = \max(0, x)$.

Neurons that share the same inputs are called a *layer*, and their transfer functions can be easily expressed as matrix operations. Given an input (row) vector $\mathbf{x} \in \mathbb{R}^n$, the vector $\mathbf{y} \in \mathbb{R}^m$ of the outputs of the neurons in the layer is obtained as $\mathbf{y} = f(\mathbf{x} \cdot \mathbf{W}^T + \mathbf{b})$, where each element $w_{i,j}$ of $\mathbf{W} \in \mathbb{R}^{m \times n}$ is the weight associated to the j -th input of the i -th neuron, each element b_i of $\mathbf{b} \in \mathbb{R}^m$ is the bias term of the i -th neuron and the activation function f is applied elementwise. It is easy to extend this notation to allow for concurrently evaluating multiple inputs by replacing the vector \mathbf{x} with a matrix, each row being a different input vector, and the vector \mathbf{b} with a matrix, each row being the same vector of bias terms.

Neural networks usually consists of multiple stacked layers, where the inputs of each layer is given by the outputs of the previous one, i.e.

$$\begin{aligned} \mathbf{y}^L &= f_L(\mathbf{y}^{L-1} \cdot \mathbf{W}^L + \mathbf{b}^L) \\ &\dots \\ \mathbf{y}^k &= f_k(\mathbf{y}^{k-1} \cdot \mathbf{W}^k + \mathbf{b}^k) \\ &\dots \\ \mathbf{y}^1 &= f_1(\mathbf{x} \cdot \mathbf{W}^1 + \mathbf{b}^1). \end{aligned}$$

Neural networks that only use differentiable activation functions can learn their parameters (weight matrices \mathbf{W}^k and biases vectors \mathbf{b}^k) by gradient descent via backpropagation. Moreover, neural networks can be used directly as classifiers by choosing a suitable activation function for the top layer (e.g. the logistic function for binary classification tasks), making it possible to automatically learn both the function that map the examples in a space suitable for classification and the classifier itself, as opposed to SVM-based methods that require either an handcrafted feature map or an handcrafted kernel function.

2.3 Related works

A large body of literature exists on the subject of relational learning. This section will describe methods based on the two most common approaches: graph kernels and neural networks on graphs.

2.3.1 Graph kernels

As we have seen in Section 2.2.1, SVM depends on a kernel function capable of comparing domain objects. While comparing data expressed as real vectors is relatively straightforward, how to compare structured objects is not obvious. Most graph kernels address this problem by adopting the approach proposed by Haussler [38]: the graphs to be compared G, G' are decomposed in sets of substructures (parts) $\mathcal{R}^{-1}(G), \mathcal{R}^{-1}(G')$, and the graph kernel $k(G, G')$ is expressed in terms of a substructure kernel k_p ; $k(G, G') = \sum_{s, s'} k_p(G, G'), s \in \mathcal{R}^{-1}(G), s' \in \mathcal{R}^{-1}(G')$.

A graph kernel is then defined by both the kind of decomposition applied to the graphs, that defines what kind of patterns will be generated, and by the substructure kernel used, that defines how the sets of parts will be compared.

Among the patterns considered from the graph kernel literature we have paths, shortest paths, walks [42], subtrees [69, 78] and neighborhood subgraphs [15]. The similarity between graphs G and G' is computed by counting the number of matches between their common the substructures (i.e. a kernel on the sets of the substructures). The match between two substructures can be defined by using graph isomorphism or some other weaker graph invariant.

When the number of substructures to enumerate is infinite or exponential with the size of the graph the kernel between the two graphs is computed without generating an explicit feature map. Learning with an implicit feature map is not scalable as it has a space complexity quadratic in the number of training examples (because we need to store in memory the gram matrix).

Other graph kernels such as the Weisfeiler-Lehman subtree kernel [78] and the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [15] deliberately choose a pattern generator that scales polynomially and produces an explicit feature map. These kernels, however, cannot handle graphs where nodes are labeled with continuous attributes.

Orsini et al. [61] proposed a general formulation called Graph Invariant

Kernels that extend existing graph kernels to handle graphs with continuous attributes. Their approach compares the sets of vertices of two attributed graphs by combining a kernel on their attributes and a weighting function dependent on the count of common graph invariants.

2.3.2 Neural networks on graphs

Early attempts to build neural networks capable of learning on graphs were based on recursive neural networks (RNN) [33, 75, 81]. These approaches build a vector representation by applying to each node a neural network that aggregates the representations of the node’s neighbors; these in turn are recursively built by other instances of the same template network. These representations are typically derived from a loss minimization procedure, where gradients are computed by the backpropagation through structure algorithm [33].

Neural networks for graphs (NN4G) [54] are feedforward neural network architectures for \mathbf{l} -attributed graphs that first apply a single layer neural network to the vertex attributes $\mathbf{l}(v)$ to produce the an initial encoding $x_1(v)$ for the vertices v in the graph G and then iteratively find new vector representations $x_i(v)$ for the vertices of the input graph G . During the successive iterations the state encoding $x_i(v)$ of a vertex v is obtained by stacking a single neural network layer with sigmoid activation functions that take as input the continuous attributes $\mathbf{l}(v)$ of v and the state encodings $x_{i'}(u)$ of the neighbors u of v during all the previous iterations $i' < i$. Finally, NN4G can either learn an output representation $y_o(p)$ for the vertices (i.e. $p = v$) or for the whole graph (i.e. $p = G$). While the former is obtained by stacking a single layer neural network over the encoding of the vertices produced across all the iterations, the latter is obtained by aggregating for each iteration i the vertex representations $x_i(v)$ over the vertices v of G , producing a graph representation $X_i(G)$ for each iteration i and then stacking stacking a single layer neural network. Differently from RNNs, NN4G can learn from graph inputs without imposing weight sharing and using feedforward neural networks.

Deep graph kernels (DGK) [95] upgrade existing graph kernels with a feature reweighing schema. DGKs represent input graphs as a corpus of substructures (e.g. graphlets, Weisfeiler-Lehman subtrees, vertex pairs with shortest path distance) and then train vector embeddings of substructures

with CBOW/Skip-gram models.¹ Each graph-kernel feature (i.e. the number of occurrences of a substructure) is reweighed by the 2-norm of the vector embedding of the corresponding substructure. Experimental evidence shows that DGKs alleviate the problem of diagonal dominance in GKs.

PATCHY-SAN [59] casts graphs into a format suitable for learning convolutional neural networks (CNNs). First, a fixed-length sequence of vertices is selected from the graph. Then, a fixed-size neighborhood is assembled for each of them. Finally, the vertices in each neighborhood are ordered according to a normalization algorithm and casted to a sequence on which a 1-dimensional convolutional neural network can be applied.

GRAPHSAGE [36] generates representations for nodes of a graph using an algorithm inspired by the Weisfeiler-Lehman isomorphism test. The initial representation \mathbf{h}_v^0 of each node v is set to the corresponding attribute vector \mathbf{x}_v . Then, for a fixed number of times K , a new representation for v is built by applying a single neural network layer to the concatenation of the node’s previous representation \mathbf{h}_v^{k-1} and an aggregated representation $\mathbf{h}_{\mathcal{N}(v)}^k$ of the neighborhood of v (according to a neighborhood function $\mathcal{N}(v)$).

Finally, Hamilton et al. [37] provide a comprehensive review of methods to embed vertices and graphs, and introduce a unified framework to describe these approaches. More specifically, the reviewed approaches are described in term of an encoder and a decoder functions: the former maps nodes to vector embeddings, while the latter decode a user-specified graph statistic from a pair of embeddings produced by the encoder.

¹The CBOW/Skip-gram models receive as inputs cooccurrences among substructures sampled from the input graphs.

Chapter 3

Shift aggregate extract networks

We introduce an architecture based on deep hierarchical decompositions to learn effective representations of large graphs. Our framework extends classic \mathcal{R} -decompositions used in kernel methods, enabling nested part-of-part relations. We also introduce a lossless compression algorithm that reduces both space and time complexity by exploiting symmetries. We show empirically that our approach outperforms current state-of-the-art graph classification methods on social network datasets.¹

3.1 Introduction

Structured data representations are common in application domains such as chemistry, biology, natural language, and social network analysis. In these domains, one can formulate a supervised learning problem where the input portion of the data is a graph, possibly with attributes on vertices and edges. While learning with graphs of moderate size (tens up to a few hundreds of vertices) can be afforded with many existing techniques, scaling up to large networks poses new significant challenges that still leave room for improvement, both in terms of predictive accuracy and in terms of computational efficiency.

Most graph-kernels- and neural-networks-based approaches have been applied to relatively small graphs, such as those derived from molecules [5, 8,

¹This chapter has been submitted as “Shift aggregate extract networks” to *Frontiers in Robotics and AI* [62].

68], natural language sentences [79] or protein structures [3, 9, 91], and only some recent works Niepert et al. [59], Yanardag and Vishwanathan [95] have been applied successfully to small graphs but also to graphs derived from social networks. Large graphs (especially social networks), in fact, typically exhibit a highly skewed degree distribution that originates a huge vocabulary of distinct subgraphs. This scenario makes finding a suitable representation much harder as kernels based on subgraph matching would suffer diagonal dominance [76], while RNNs would face the problem of composing a highly variable number of substructure representations in the recursive step.

In this chapter, we introduce a novel architecture for machine learning with structured inputs, called shift-aggregate-extract network (SAEN). Structured inputs are first decomposed in a hierarchical fashion. A feedforward neural network is then *unfolded* over the hierarchical decompositions using *shift*, *aggregate* and *extract* operations (see Section 3.4). Finally, gradient descent learning is applied to the resulting network.

Like the flat \mathcal{R} -decompositions commonly used to define kernels on structured data [38], \mathcal{H} -decompositions are based on the *part-of* relation, but allow us to introduce a deep recursive notion of *parts of parts*. At the top level of the hierarchy lies the *whole* data structure. Objects at each intermediate level are decomposed into parts that form the subsequent level of the hierarchy. The bottom level consists of atomic objects, such as individual vertices, edges or small graphlets.

SAEN compensates some limitations of recursive neural networks by adding two synergetic degrees of flexibility. First, it unfolds a neural network over a hierarchy of parts rather than using the edge set of the input graph directly; this makes it easier to deal with very high degree vertices. Second, it imposes weight sharing and fixed size of the learned vector representations on a per level basis instead of globally; in this way, more complex parts may be embedded into higher dimensional vectors, without forcing to use excessively large representations for simpler parts.

A second contribution of this work is a *domain compression* algorithm that can significantly reduce memory usage and runtime. It leverages mathematical results from lifted linear programming [55] in order to exploit symmetries and perform a lossless compression of \mathcal{H} -decompositions.

The chapter is organized as follows. In Section 3.2 we introduce \mathcal{H} -decompositions, a generalization of Haussler’s \mathcal{R} -decomposition relations [38]. In Section 3.4 we describe SAEN, a neural network architecture for learning

vector representations of \mathcal{H} -decompositions. Furthermore, in Section 3.5 we explain how to exploit symmetries in \mathcal{H} -decompositions in order to reduce memory usage and runtime. In Section 3.6 we report experimental results on several number of real-world datasets. Finally, in Section 3.7 we discuss some related works and draw some conclusions in Section 3.8.

3.2 \mathcal{H} -decompositions

In this section, we define a deep hierarchical extension of Haussler’s \mathcal{R} -decomposition relation [38].

An \mathcal{H} -decomposition is formally defined as the triple $(\{S_l\}_{l=0}^L, \{\mathcal{R}_{l,\pi}\}_{l=1}^L, X)$ where:

- $\{S_l\}_{l=0}^L$ are disjoint sets of objects S_l called levels of the hierarchy. The bottom level S_0 contains atomic (i.e. non-decomposable) objects, while the other levels $\{S_l\}_{l=1}^L$ contain compound objects, $s \in S_l$, whose parts $s' \in S_{l-1}$ belong to the preceding level, S_{l-1} .
- $\{\mathcal{R}_{l,\pi}\}_{l=1}^L$ is a set of l, π -parametrized $\mathcal{R}_{l,\pi}$ -convolution relations, where $\pi \in \Pi_l$ is a membership type from a finite alphabet Π_l of size $n(l) = |\Pi_l|$. At the bottom level, $n(0) = 1$. A pair $(s, s') \in S_l \times S_{l-1}$ belongs to $\mathcal{R}_{l,\pi}$ iff s' is part of s with membership type π . For notational convenience, the parts of s are denoted as $\mathcal{R}_{l,\pi}^{-1}(s) = \{s' \mid (s', s) \in \mathcal{R}_{l,\pi}\}$.
- X is a set $\{\mathbf{x}(s)\}_{s \in S_0}$ of p -dimensional vectors of attributes assigned to the elements s the bottom layer S_0 .

The membership type π is used to represent the roles of the parts of an object. For $L > 1$, an \mathcal{H} -decomposition is a multilevel generalization of the classic \mathcal{R} -convolution. It represents structured data as a hierarchy of π -parametrized parts. Some concrete examples of \mathcal{H} -decompositions are given in the following section.

Example 1. *In Figure 3.1 we propose a 4-level \mathcal{H} -decomposition by decomposing graph $Graph \in S_3$ into a set of radius-neighborhood (radius $\in \{1, 2\}$) subgraphs $Ball \in S_2$ (see Figure 3.2 for a pictorial representation of the parts) and employ their radius as membership type. Furthermore, we extract edges $Edge \in S_1$ from the radius-neighborhood subgraphs. Finally, each edge is decomposed in vertices $V \in S_0$. The elements of the $\mathcal{R}_{l,\pi}$ -convolution are*

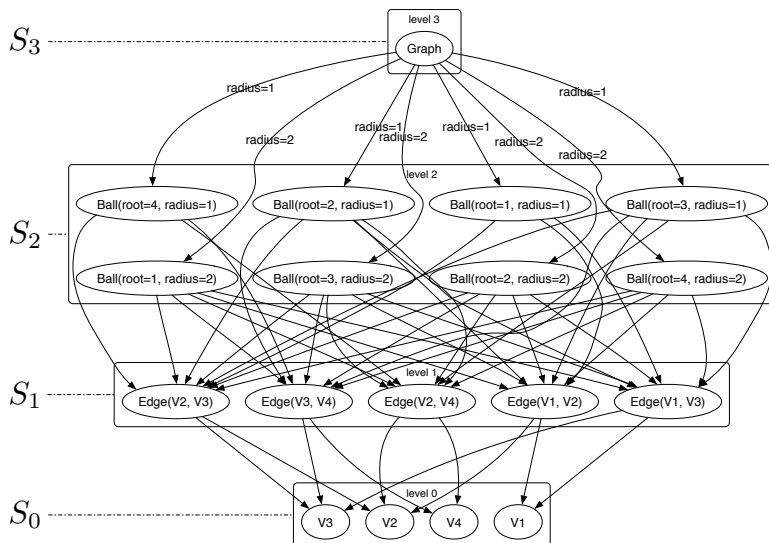


Figure 3.1: Pictorial representation of a sample \mathcal{H} -decomposition. We produce a 4-level \mathcal{H} -decomposition by decomposing graph $Graph \in S_3$ into a set of $radius$ -neighborhood ($radius \in \{1, 2\}$) subgraphs $Ball \in S_2$ and employ their $radius$ as membership type. Furthermore, we extract edges $Edge \in S_1$ from the $radius$ -neighborhood subgraphs. Finally, each edge is decomposed in vertices $V \in S_0$. The elements of the $\mathcal{R}_{l,\pi}$ -convolution are pictorially shown as directed arcs. Since membership types π for edges and vertices would be all identical their label is not represented in the picture.

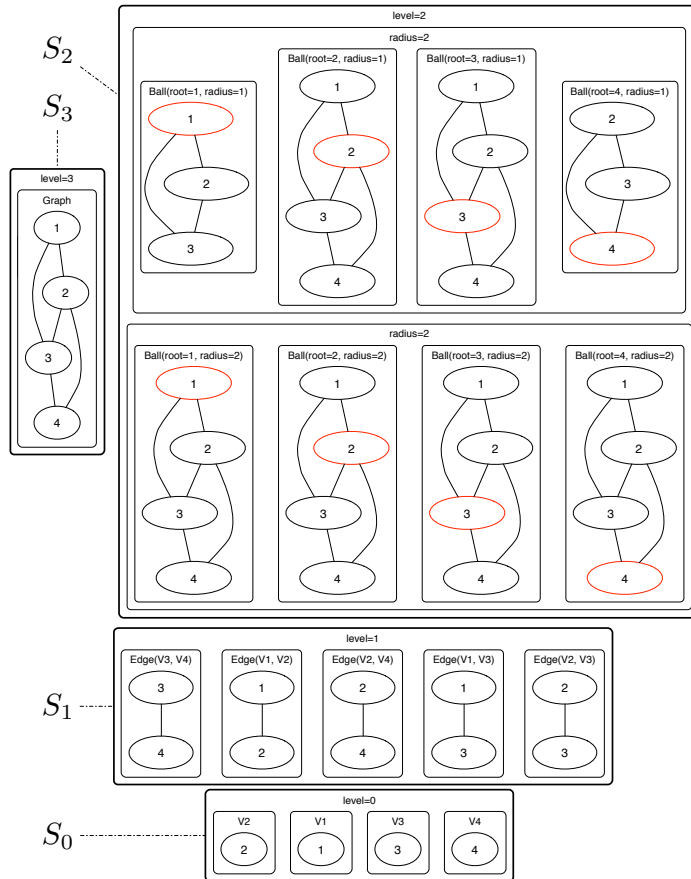


Figure 3.2: Pictorial representation of the substructures that are contained in each node of the \mathcal{H} -decomposition showed in Figure 3.1. The objects of the \mathcal{H} -decomposition are grouped to according their S_l sets ($l = 0, \dots, 3$). For each *radius*-neighborhood subgraph we show the root node in red.

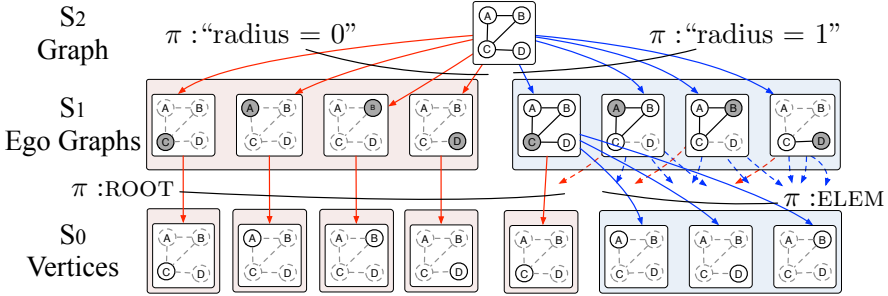


Figure 3.3: The EGD is an \mathcal{H} -decomposition structured in 3 levels. Level 2 contains the input attributed graph $G = (V, E, X)$ where V is the set of vertices and E is the set of edges and X is a set of p -dimensional vectors of attributes assigned to the vertices $v \in V$ of the graph G . The input graph G is then decomposed into ego graphs g of radius $r = 0, \dots, R$ where R is the maximum radius that we allow in the decomposition. The ego graphs g are elements of level 1 and are parts of G with π -type r . Ego graphs g are further decomposed into vertices v . We use the π -types ROOT and ELEM to specify whether a vertex v is the root of the ego graph g or just an element respectively. The vertices v which are the elements of level 0 and are labeled with vectors of vertex attributes.

pictorially shown as directed arcs. Since membership types π for edges and vertices would be all identical their label is not represented in the picture.

3.3 Instances of \mathcal{H} -decompositions

We describe three \mathcal{H} -decompositions based on ego graphs, on pairs of ego graphs, and on shortest paths. They are inspired from closely related graph kernels.

3.3.1 Ego Graph Decomposition

The ego graph \mathcal{H} -decomposition (EGD) has $L = 3$ levels defined as follows (see Figure 3.3):

- Level 2 consists of the whole attributed graph $G = (V, E, \mathbf{x})$ where \mathbf{x} is a labeling function that attaches a p -dimensional vector of attributes $\mathbf{x}(v)$ to each vertex v .

- Level 1 consists of all ego graphs $g_{v,r}$ with roots $v \in V$ and $r \in [0, R]$. The π -type of $g_{v,r}$ is simply r . Note that for $r = 0$, all ego graphs $g_{v,0}$ consist of single vertices.
- Level 0 consists of single vertices with two possible π -types: ROOT and ELEM to specify whether a vertex v is the root $g_{v,r}$ or not.

3.3.2 Nested Ego Graph Decomposition

The nested ego graph \mathcal{H} -decomposition (NEGD) has $L = 3$ levels defined as follows:

- Level 2 (S_2) consists of the whole attributed graph $G = (V, E, f_V, f_E)$ where f_V and f_E are two labeling functions that attach respectively a p -dimensional vector of attributes $f_V(v)$ to each vertex v and a symbol $f_E(u, w)$ from a finite alphabet Π_1 to each edge (u, w) .
- Level 1 (S_1) consists of all ego graphs $g_{v,1} = (V_v, E_v)$ with roots $v \in V$. The π -type of $g_{v,1}$ is the number of vertices $|V_v|$.
- Level 0 (S_0) consists of the ego graphs $g_{w,1}$, $\forall w \in V_v$, with π -type ROOT if $w = v$, or π -type $f_E(v, w)$ otherwise.
- A bijection $\mathbf{x} : S_0 \rightarrow \mathbb{N}$ associates a different identifier to each distinct ego graph in S_0 , i.e. $\mathbf{x}(s_1) = \mathbf{x}(s_2) \iff s_1 = s_2, \forall s_1, s_2 \in S_0$.

3.4 Learning representations with SAEN

A shift-aggregate-extract network (SAEN) is a composite function that maps objects at level l of an \mathcal{H} -decomposition into $d(l)$ -dimensional real vectors. It uses a sequence of parametrized functions $\{f_0, \dots, f_L\}$, for example a sequence of neural networks with parameters $\theta_0, \dots, \theta_L$ that will be trained during the learning. At each level, $l = 0, \dots, L$, each function $f_l : \mathbb{R}^{n(l)d(l)} \rightarrow \mathbb{R}^{d(l+1)}$ operates as follows:

1. It receives as input the *aggregate* vector $\mathbf{a}_l(s)$ defined as:

$$\mathbf{a}_l(s) = \begin{cases} \mathbf{x}(s) & \text{if } l = 0 \\ \sum_{\pi \in \Pi_l} \sum_{s' \in \mathcal{R}_{l,\pi}^{-1}(s)} \mathbf{z}_\pi \otimes \mathbf{h}_{l-1}(s') & \text{if } l > 0 \end{cases} \quad (3.1)$$

where $\mathbf{x}(s)$ is the vector of attributes for object s .

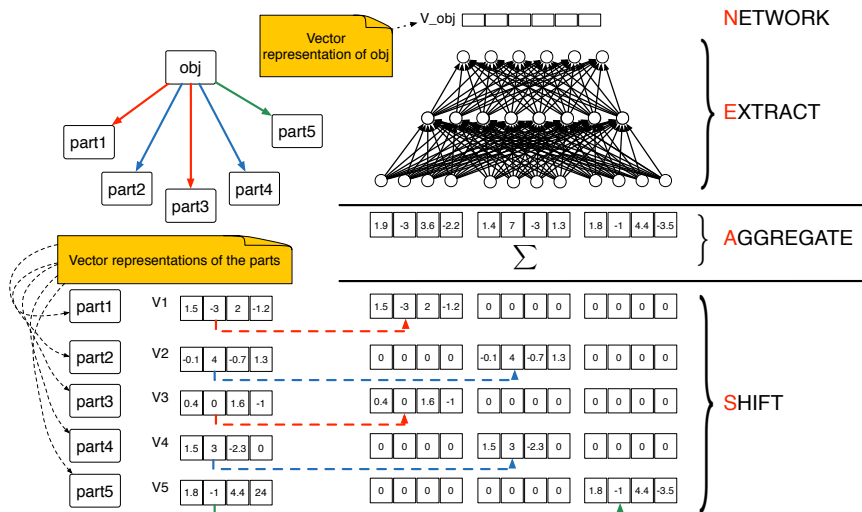


Figure 3.4: Pictorial representation of the SAEN computation explained in Eq. 3.1 and Eq. 3.2. The SAEN computation is unfolded over all the levels of an \mathcal{H} -decomposition. On the top-right part we show an object $obj \in S_l$ decomposed into its parts $\{part_i\}_{i=1}^5 \subseteq S_{l-1}$ from the level below. The parametrized “part of” relation $\mathcal{R}_{l,pi}$ is represented by directed arrows, we use colors (red, blue and green) to distinguish among π -types. In the bottom-left part of the picture we show that each part is associated to a vectorial representation. In the bottom-right part of the picture we show the *shift* step in which the vector representations of the parts are shifted using the Kronecker product in Eq. 3.1. Then the shifted representation are summed in the aggregation step and in the extract step a feedforward neural is applied in order to obtain the vector representation of object obj .

2. It *extracts* the vector representation of s as

$$\mathbf{h}_l(s) = f_l(\mathbf{a}_l(s); \theta_l). \quad (3.2)$$

The vector $\mathbf{a}_l(s)$ is obtained in two steps: first, previous level representations $\mathbf{h}_{l-1}(s')$ are *shifted* via the Kronecker product \otimes using an indicator vector $\mathbf{z}_\pi \in \mathbb{R}^{n(l)}$. This takes into account of the membership types π . Second, shifted representations are *aggregated* with a sum. Note that all representation sizes $d(l)$, $l > 0$ are hyper-parameters that need to be chosen or adjusted.

The shift and aggregate steps are identical to those used in kernel design when computing the explicit feature of a kernel $k(x, z)$ derived from a sum $\sum_{\pi \in \Pi} k_\pi(x, z)$ of base kernels $k_\pi(x, z)$, $\pi \in \Pi$. In principle, it would be indeed possible to turn SAEN into a kernel method by removing the extraction step and define the explicit feature for a kernel on \mathcal{H} -decompositions. Removing the extraction step from Eq. 3.1 results in:

$$\mathbf{a}_l(s) = \begin{cases} \mathbf{x}(s) & \text{if } l = 0 \\ \sum_{\pi \in \Pi_l} \sum_{s' \in \mathcal{R}_{l,\pi}^{-1}(s)} \mathbf{z}_\pi \otimes \mathbf{a}_{l-1}(s') & \text{if } l > 0 \end{cases} \quad (3.3)$$

However, that approach would increase the dimensionality of the feature space by a multiplicative factor $n(l)$ for each level l of the \mathcal{H} -decomposition, thus leading to an exponential number of features. When the number of features is exponential, their explicit enumeration is impractical. A possible solution would be to directly define the kernel similarity and keep the features implicit [13]. However, this solution would have space complexity that is quadratic in the number of graphs in the dataset. Some approaches overcome this limitation by employing hash or hash-like functions to build a compact representation of hierarchical structures [26, 96]; these methods, however, cannot learn how to build that representation as the hash function is fixed.

When using SAEN, the feature space growth is prevented by exploiting a distributed representation (via a multilayered neural network) during the extraction step. As a result, SAEN can easily cope with \mathcal{H} -decompositions consisting of multiple levels.

3.5 Exploiting symmetries for domain compression

In this section we propose a technique, called *domain compression*, which allows us to save memory and speed up the SAEN computation. Domain compression exploits symmetries in \mathcal{H} -decompositions to compress them without information loss. This technique requires that the attributes $\mathbf{x}(s)$ of the elements s in the bottom level S_0 are categorical.

Definition 7. *Two objects a, b in a level S_l are collapsible, denoted $a \sim b$, if they share the same representation, i.e., $\mathbf{h}_l(a) = \mathbf{h}_l(b)$ for all the possible values of the parameters $\theta_0, \dots, \theta_l$.*

According to Definition 7, objects in the bottom level S_0 are collapsible when their attributes are identical, while objects at any level $\{S_l\}_{l=1}^L$ are collapsible if they are made of the same sets of parts for all the membership types π .

A compressed level S_l^{comp} is the quotient set of level S_l with respect to the collapsibility relation \sim .

Before providing a mathematical formulation of domain compression we provide two examples: in Example 2 we explain the intuition beyond domain compression showing in Figure 2 the steps that need to be taken to compress a \mathcal{H} -decomposition, in Example 3 we provide a pictorial representation of the \mathcal{H} -decomposition of a real world graph and its compressed version.

Example 2. *Figure 3.5 a) shows the pictorial representation of an \mathcal{H} -decomposition whose levels are denoted with the letters of the alphabet A, B, C, D. We name each object using consecutive integers prefixed with the name of the level. We use purple and orange circles to denote the categorical attributes of the objects of the bottom stratum. Directed arrows denote the “part of” relations whose membership type is distinguished using the colors blue and red.*

Figure 3.5 b) shows the domain compression of the \mathcal{H} -decomposition in a). When objects are collapsed the directed arcs coming from their parents are also collapsed. Collapsed arcs are labeled with their cardinality.

Figures 3.5 c), d), e) and f) describe the domain compression steps starting from level A until level D.

- *Figure 3.5 c) shows that since A3 and A4 have the same categorical attribute of A1 (i.e. purple) they are grouped and collapsed to A1. Fur-*

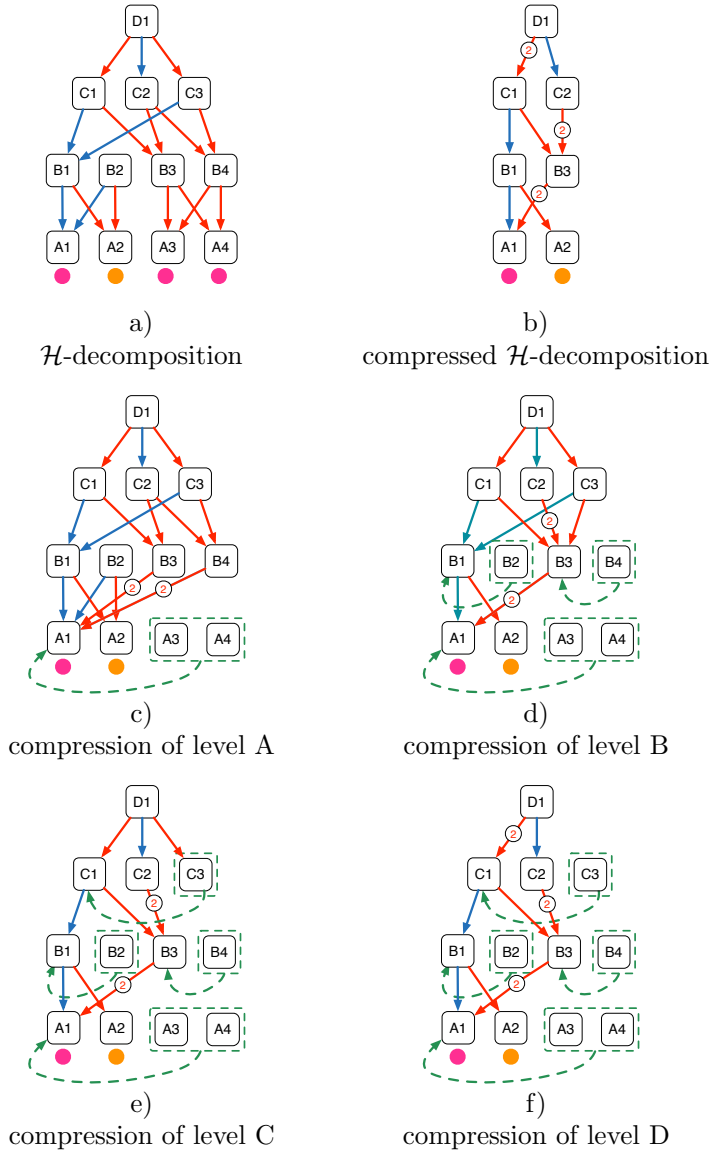


Figure 3.5: Intuition of the domain compression algorithm explained in Example 2.

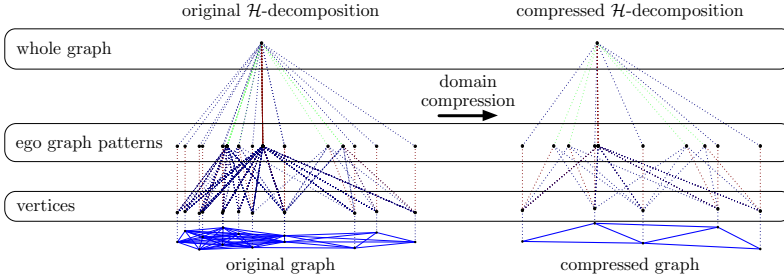


Figure 3.6: Pictorial representation of the \mathcal{H} -decomposition of a graph taken from the IMDB-BINARY dataset (see Section 3.6.1) together with its compressed version.

thermore, the arrows in the fan-in of $A3$ and $A4$ are attached to $A1$ with the consequent cardinality increase of the red arrows that come from $B3$ and $B4$.

- In Figure 3.5 d) we show the second iteration of domain compression in which objects made of the same parts with the same membership types are collapsed. Both $B1$ and $B2$ in Figure 3.5 c) were connected to $A1$ with a blue arrow and to $A2$ with a red arrow and so they are collapsed. In the same way $B3$ and $B4$ are collapsed because in c) they were connected to $A1$ with a red arrow with cardinality 2.
- In Figure 3.5 e) $C1$ and $C3$ are collapsed because in d) they were both connected to $B1$ with a blue arrow and $B3$ with a red arrow.
- Finally in f) since $C1$ and $C3$ were collapsed in the previous step we increase to 2 the cardinality of the red arrow that connects $D1$ and $C1$ and remove the red arrow from $D1$ to $C3$ since $C3$ was collapsed to $C1$ in Figure 3.5 e).

The final result of domain compression is illustrated in Figure 3.5 b).

Example 3. In Figure 3.6 we provide a pictorial representation of the domain compression of an \mathcal{H} -decomposition (EGD, described in Section 3.3.1). On the left we show the \mathcal{H} -decomposition of a graph taken from the IMDB-BINARY dataset (see Section 3.6.1) together with its compressed version on the right.

In order to compress \mathcal{H} -decompositions we adapt the lifted linear programming technique proposed by Mladenov et al. [55] to the SAEN architecture. A

matrix $M \in \mathbb{R}^{n \times p}$ with $m \leq n$ distinct rows can be decomposed as the product DM^{comp} where M^{comp} is a compressed version of M in which the distinct rows of M appear exactly once.

Definition 8. The Boolean decomposition matrix, D , encodes the collapsibility relation among the rows of M so that $D_{ij} = 1$ iff the i^{th} row of M falls in the equivalence class j of \sim , where \sim is the equivalence relation introduced in Definition 7.²

Example 4. (Example 2 continued)

The bottom level of the \mathcal{H} -decomposition in Figure 3.5 a) has 4 objects $A1$, $A2$, $A3$ and $A4$ with categorical attributes indicated with colors.

Objects $A1$, $A2$, $A4$ have a purple categorical attribute while $A3$ has a orange categorical attribute. If we give to purple the encoding $[0, 3]$ and to orange the encoding $[4, 1]$ we obtain an attribute matrix

$$X = \begin{bmatrix} 0 & 3 \\ 0 & 3 \\ 4 & 1 \\ 0 & 3 \end{bmatrix} \quad (3.4)$$

in which each row contains the encoding of the categorical attribute of an object of the bottom stratum and objects were taken with the order $A1$, $A2$, $A3$, $A4$.

Since the rows associated to $A1$, $A3$, $A4$ are identical we can compress matrix X to matrix

$$X^{comp} = \begin{bmatrix} 0 & 3 \\ 4 & 1 \end{bmatrix} \quad (3.5)$$

as we can notice this is the attribute matrix of the compressed \mathcal{H} -decomposition shown in Figure 3.5 b).

Matrix X can be expressed as the matrix product DX^{comp} between the decomposition matrix D and the compressed version of X^{comp} where

$$D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.6)$$

² Mladenov et al. [55] lifts linear programming and defines the equivalence relation induced from the labels obtained by performing color passing on a Gaussian random field. We use an the equivalence relation in Definition 7 because we are working with \mathcal{H} -decompositions.

and was obtained applying Definition 8.

As explained in Mladenov et al. [55] a pseudo-inverse C of D can be computed by dividing the rows of D^\top by their sum (where D^\top is the transpose of D).

However, it is also possible to compute a pseudo-inverse C' of D by transposing D and choosing one representer for each row of D^\top . For each row of D^\top we can simply choose a nonzero element as representer and set all the other to zero.

Example 5. *The computation of the pseudo-inverse C of the D matrix of Example 4 results in the following equation:*

$$C = \begin{bmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.7)$$

the matrix multiplication between the compression matrix C and the X leads to the compressed matrix X^{comp} (i.e. $X^{comp} = CX$).

In the first row of matrix C there are 3 nonzero entries that correspond to the objects **A1**, **A2**, **A4**, while on the second row there is a nonzero entry that corresponds to object **A3**.

As we said above, since we know that the encodings of those objects are identical instead of making the average we could just take a representer.

For example in Figure 3.5 c) we chose **A1** as representer for **A2** and **A4**, obtaining the compression matrix

$$C' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.8)$$

In the first row of matrix C' there is a nonzero entry that correspond to the object **A1** (which is the chosen representer), while on the second row there is a nonzero entry that corresponds to object **A3** (as in C).

While from the compression point of view we still have $X^{comp} = C'X$, choosing a representer instead of averaging equivalent objects is advantageous when using sparse matrices because the number of nonzero elements decreases.

We apply domain compression to SAEN by rewriting Eqs. 3.1 and 3.2 in matrix form.

We rewrite Eq. 3.1 as:

$$A_l = \begin{cases} X & \text{if } l = 0 \\ \mathbf{R}_l \mathbf{H}_{l-1} & \text{if } l > 0 \end{cases} \quad (3.9)$$

where:

- $A_l \in \mathbb{R}^{|S_l| \times n(l-1)d(l)}$ is the matrix that represents the *shift-aggregated* vector representations of the object of level S_{l-1} ;
- $X \in \mathbb{R}^{|S_0| \times p}$ is the matrix that represents the p -dimensional encodings of the vertex attributes in V (i.e. the rows of X are the \mathbf{x}_{v_i} of Eq. 3.1);
- $\mathbf{R}_l \in \mathbb{R}^{|S_l| \times n(l)|S_{l-1}|}$ is the concatenation

$$\mathbf{R}_l = [R_{l,1}, \dots, R_{l,\pi}, \dots, R_{l,n(l)}] \quad (3.10)$$

of the matrices $R_{l,\pi} \in \mathbb{R}^{|S_l| \times |S_{l-1}|} \forall \pi \in \Pi_l$ which represent the $\mathcal{R}_{l,\pi}$ -convolution relations of Eq. 3.1 whose elements are $(R_{l,\pi})_{ij} = 1$ if $(s', s) \in \mathcal{R}_{l,\pi}$ and 0 otherwise.

- $\mathbf{H}_{l-1} \in \mathbb{R}^{n(l)|S_{l-1}| \times n(l)d(l)}$ is a block-diagonal matrix

$$\mathbf{H}_{l-1} = \begin{bmatrix} H_{l-1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & H_{l-1} \end{bmatrix} \quad (3.11)$$

whose blocks are formed by matrix $H_{l-1} \in \mathbb{R}^{|S_{l-1}| \times d(l)}$ repeated $n(l)$ times. The rows of H_{l-1} are the vector representations \mathbf{h}_j in Eq. 3.1.

Eq. 3.2 is simply rewritten to $H_l = f_l(A_l; \theta_l)$ where $f_l(\cdot; \theta_l)$ is unchanged w.r.t. Eq. 3.2 and is applied to its input matrix A_l row-wise.

Domain compression on Eq. 3.9 is performed by the DOMAIN-COMPRESSON procedure (see Algorithm 1). which takes as input the attribute matrix $X \in \mathbb{R}^{|S_0| \times p}$ and the part-of matrices $R_{l,\pi}$ and returns their compressed versions X^{comp} and the $R_{l,\pi}^{comp}$ respectively. The algorithm starts by invoking (line 1) the procedure COMPUTE-CD on X to obtain the compression and decompression matrices C_0 and D_0 respectively. The compression matrix C_0 is used to compress X (line 2) then we start iterating over the levels $l = 0, \dots, L$ of the \mathcal{H} -decomposition (line 4) and compress the $R_{l,\pi}$ matrices. The compression of the $R_{l,\pi}$ matrices is done by right-multiplying them by the decompression matrix D_{l-1} of the previous level $l - 1$ (line 5). In this way we collapse the parts of relation $\mathcal{R}_{l,\pi}$ (i.e. the columns of $R_{l,\pi}$) as these were identified in level S_{l-1} as identical objects (i.e. those objects corresponding to the rows of X or $R_{l-1,\pi}$ collapsed during the previous step).

Algorithm 1 DOMAIN-COMPRESSION

```

DOMAIN-COMPRESSION( $X, R$ )
1   $C_0, D_0 = \text{COMPUTE-CD}(X)$ 
2   $X^{comp} = C_0 X$ 
3   $R^{comp} = \{\}$ 
4  for  $l = 1$  to  $L$ 
5       $R^{col\_comp} = [R_{l,\pi} D_{l-1}, \forall \pi = 1, \dots, n(l)]$ 
6       $C_l, D_l = \text{COMPUTE-CD}(R^{col\_comp})$ 
7      for  $\pi = 1$  to  $n(l)$ 
8           $R_{l,\pi}^{comp} = C_l R_{l,\pi}^{col\_comp}$ 
9  return  $X^{comp}, R^{comp}$ 

```

The result is a list $R^{col_comp} = [R_{l,\pi} D_{l-1}, \forall \pi = 1, \dots, n(l)]$ of column compressed $R_{l,\pi}$ -matrices. We proceed collapsing equivalent objects in level S_l , i.e. those made of identical sets of parts: we find symmetries in R^{col_comp} by invoking COMPUTE-CD (line 6) and obtain a new pair C_l, D_l of compression, and decompression matrices respectively. Finally the compression matrix C_l is applied to the column-compressed matrices in R^{col_comp} in order to obtain the Π_l compressed matrices of level S_l (line 8).

Algorithm 1 allows us to compute the domain compressed version of Eq. 3.9 which can be obtained by replacing: X with $X^{comp} = C_0 X$, $R_{l,\pi}$ with $R_{l,\pi}^{comp} = C_l R_{l,\pi} D_{l-1}$ and H_l with H_l^{comp} . Willing to recover the original encodings H_l we just need to employ the decompression matrix D_l on the compressed encodings H_l^{comp} , indeed $H_l = D_l H_l^{comp}$. This is possible as the domain compression procedure guarantees that, for $l = 1, \dots, L$, $H_l = f_l(A_l; \theta) = f_l(D_l C_l A_l; \theta) = D_l f_l(C_l A_l; \theta)$, as the functions f_l operate row-wise and the compression algorithm only removes duplicated rows, collapsing them to a single representer. Other matrix decomposition approaches such as SVD would not offer such guarantee, making the compression and decompression matrices dependent on the trainable representation-building functions and their parameters. As a consequence of that, any potential advantage of using those matrix decompositions would be negated by the fact that it would be necessary to run the compression algorithm after every training step to take into account the variation of the parameters of the network.

The domain compression strategy of SAEN is similar to the hierarchical compression method described by Aioli et al. [2], where a whole forest is compressed to a single directed acyclic graph by collapsing identical subtrees. In SAEN, however, we preserve the hierarchy by collapsing only objects belonging to the same level, whereas in the forest compression algorithm every set of identical objects is collapsed regardless of their position in the original hierarchical structure.

As we can see by substituting S_l with S_l^{comp} , the more are the symmetries (i.e. when $|S_l^{comp}| \ll |S_l|$) the greater the domain compression will be. This makes the proposed algorithm particularly effective when dealing with matrices resulting from \mathcal{H} -decompositions, as having nested part-of relationships make it likely to have multiple objects composed of identical sets of parts.

3.6 Experimental evaluation

We perform an experimental evaluation of SAEN on graph classification datasets and answer the following questions:

- Q1** How does SAEN compare to the state of the art?
- Q2** Can SAEN exploit symmetries in social networks to reduce the memory usage and the runtime?

3.6.1 Datasets

In order to answer the experimental questions we tested our method on six publicly available datasets first proposed by Yanardag and Vishwanathan [95].

- **COLLAB**
is a dataset where each graph represent the ego-network of a researcher, and the task is to determine the field of study of the researcher between *High Energy Physics*, *Condensed Matter Physics* and *Astro Physics*.
- **IMDB-BINARY, IMDB-MULTI**
are datasets derived from IMDB where in each graph the vertices represent actors/actresses and the edges connect people which have performed in the same movie. Collaboration graphs are generated from

Table 3.1: Statistics of the datasets used in our experiments.

DATASET	SIZE	AVG. VERTICES	AVG. MAX. DEGREE
COLLAB	5000	74.49	73.62
IMDB-BINARY	1000	19.77	18.77
IMDB-MULTI	1500	13.00	12.00
REDDIT-BINARY	2000	429.62	217.35
REDDIT-MULTI5K	5000	508.51	204.08
REDDIT-MULTI12K	11929	391.40	161.70
MUTAG	188	17.93	3.01
PTC	344	25.56	3.73
NCI1	4110	29.87	3.34
PROTEINS	1113	39.06	5.79

movies belonging to genres *Action* and *Romance* for IMDB-BINARY and *Comedy*, *Romance* and *Sci-Fi* for IMDB-MULTI, and for each actor/actress in those genres an ego-graph is extracted. The task is to identify the genre from which the ego-graph has been generated.

- REDDIT-BINARY, REDDIT-MULTI5K, REDDIT-MULTI12K are datasets where each graph is derived from a discussion thread from Reddit. In those datasets each vertex represent a distinct user and two users are connected by an edge if one of them has responded to a post of the other in that discussion. The task in REDDIT-BINARY is to discriminate between threads originating from a discussion-based subreddit (*TrollXChromosomes*, *atheism*) or from a question/answers-based subreddit (*IAmA*, *AskReddit*). The task in REDDIT-MULTI5K and REDDIT-MULTI12K is a multiclass classification problem where each graph is labeled with the subreddit where it has originated (*worldnews*, *videos*, *AdviceAnimals*, *aww*, *mildlyinteresting* for REDDIT-MULTI5K and *AskReddit*, *AdviceAnimals*, *atheism*, *aww*, *IAmA*, *mildlyinteresting*, *Showerthoughts*, *videos*, *todayilearned*, *worldnews*, *TrollXChromosomes* for REDDIT-MULTI12K).

Even if our objective was to build a method suitable for large graphs, for the sake of completeness we also tested our method on some small bioinformatic datasets.

- MUTAG [21] is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds labeled according to whether or not they have a mutagenic effect on the Gramnegative bacterium *Salmonella typhimurium*. PTC [87] is a dataset of 344 chemical compounds that reports the carcinogenicity for male and female rats and it has 19 discrete labels. NCI1 [92] is a dataset of 4100 examples and is a subset of balanced datasets of chemical compounds screened for ability to suppress or inhibit the growth of a panel of human tumor cell lines. PROTEINS [9] is a binary classification dataset made of 1113 proteins. Each protein is represented as a graph where vertices are secondary structure elements (i.e. helices, sheets and turns). Edges connect nodes if they are neighbors in the amino-acid sequence or in the 3D space.

3.6.2 Experiments

E1 We experiment with SAEN applying the EGD \mathcal{H} -decomposition on PROTEINS, COLLAB, IMDB-BINARY, IMDB-MULTI, REDDIT-BINARY, REDDIT-MULTI5K, and REDDIT-MULTI12K, and the NEGD \mathcal{H} -decomposition on MUTAG, PTC, and NCI1. We used the colors resulting from 4 iterations of the Weisfeiler-Lehman algorithm [78] as identifiers for the ego graphs contained in the bottom level of NEGD.

In order to perform classification we add a cross-entropy loss on the extraction step $h_L(s)$ (see Eq. 3.2) of the top level L (i.e. $L = 2$) of the EGNN \mathcal{H} -decomposition. We used Leaky ReLUs [52] as activation function on all the units of the neural networks $\{f_l(\cdot; \Theta_l)\}_{l=0}^2$ of the extraction step (cf. Eq. 3.2).

SAEN was implemented in TensorFlow and in all our experiments we trained the neural network parameters $\{\Theta_l\}_{l=0}^2$ by using the Adam algorithm [44] to minimize a cross-entropy loss.

The classification accuracy of SAEN was measured by 10-times 10-fold cross-validation. With respect to the selection of the hyper-parameters for each social network dataset, we chose the number of layers and units for each level of the part-of decomposition, the size of each layer, and the coefficient for L2 regularization on the network weights by training on 8/9th of the training set of the first split of the 10-times 10-fold cross-validation and using as validation set the remaining 1/9th to evaluate the chosen parameters. For each social network dataset we report the mean and the standard deviation

Table 3.2: Comparison of accuracy results on social network datasets.

DATASET	DGK [95]	PATCHY-SAN [59]	SAEN (our method)
COLLAB	73.09 ± 0.25	72.60 ± 2.16	78.50 ± 0.69
IMDB-BINARY	66.96 ± 0.56	71.00 ± 2.29	71.59 ± 1.20
IMDB-MULTI	44.55 ± 0.52	45.23 ± 2.84	48.53 ± 0.76
REDDIT-BINARY	78.04 ± 0.39	86.30 ± 1.58	87.22 ± 0.80
REDDIT-MULTI5K	41.27 ± 0.18	49.10 ± 0.70	53.63 ± 0.51
REDDIT-MULTI12K	32.22 ± 0.10	41.32 ± 0.42	45.27 ± 0.30

Table 3.3: Comparison of accuracy on bio-informatics datasets.

DATASET	PATCHY-SAN [59]	SAEN (our method)
MUTAG	92.63 ± 4.21	82.48 ± 1.43
PTC	62.29 ± 5.68	56.80 ± 1.40
NCI1	78.59 ± 1.89	78.62 ± 0.40
PROTEINS	75.89 ± 2.76	72.73 ± 0.96

of the 10 accuracy values in Table 3.2 where we compare our results with those by Yanardag and Vishwanathan [95] and by Niepert et al. [59]. In Table 3.3 we compare the results obtained by our method on bioinformatic datasets with those obtained by Niepert et al. [59] reporting mean and the standard deviation obtained with the same statistical protocol.

In Table 3.4 we report for each dataset the radiuses r of the neighborhood subgraphs used in the EGD decomposition and the number of units in the hidden layers for each level.

E2 In Table 3.5 we show the file sizes of the preprocessed datasets before and after the compression together with the data compression ratio.³ We also estimate the benefit of domain compression from a computational time point of view and report the measurement of the runtime for 10 epochs with and without compression together with the speedup factor. We do not report the compression algorithm runtime as it adds up at most one minute to the

³The size of the uncompressed files are shown for the sole purpose of computing the data compression ratio. Indeed the last version of our code compresses the files on the fly.

Table 3.4: Parameters used for the EGD decompositions for each datasets.

DATASET	DECOMPOSITION	HIDDEN UNITS		
		S_0	S_1	S_2
COLLAB	EGD, $r = 1$	15 – 5	5 – 2	5 – 3
IMDB-BINARY	EGD, $r = 2$	2	5 – 2	5 – 3 – 1
IMDB-MULTI	EGD, $r = 2$	2	5 – 2	5 – 3
REDDIT-BINARY	EGD, $r = 1$	10 – 5	5 – 2	5 – 3 – 1
REDDIT-MULTI5K	EGD, $r = 1$	10	10	6 – 5
REDDIT-MULTI12K	EGD, $r = 1$	10	10	20 – 11
MUTAG	NEGD	20	40 – 20	40 – 20 – 1
PTC	NEGD	50	100 – 50	100 – 50 – 1
NCI1	NEGD	50	100 – 50	100 – 50 – 1
PROTEINS	EGD, $r = 3$	3	3	9 – 6 – 1

Table 3.5: Comparison of sizes and runtimes (for 10 epochs) of the datasets before and after the compression.

DATASET	SIZE (MB)			RUNTIME		
	ORIGINAL	COMP.	RATIO	ORIGINAL	COMP.	SPEEDUP
COLLAB	337	119	0.35	2' 27"	1' 06"	2.23
IMDB-BINARY	24	18	0.75	8"	6"	1.33
IMDB-MULTI	31	25	0.81	19"	17"	1.12
REDDIT-BINARY	129	47	0.36	47"	16"	2.94
REDDIT-MULTI5K	368	132	0.36	2' 10"	55"	2.36
REDDIT-MULTI12K	712	287	0.40	4' 25"	2' 02"	2.17

decomposition time, which is negligible compared to the training time.

For the purpose of this experiment, all tests were run on a computer with two 8-cores Intel Xeon E5-2665 processors and 94 GB RAM. SAEN was implemented in Python with the TensorFlow library.

3.6.3 Discussion

A1 As shown in Table 3.2, EGD performs consistently better than the other two methods on all the social network datasets. This confirms that the chosen \mathcal{H} -decomposition is effective on this kind of problems. Table 3.1 shows that the average maximum node degree (AMND)⁴ of the social network datasets is in the order of 10^2 . SAEN can easily cope with highly skewed node degree distributions by aggregating distributed representation of patterns while this is not the case for DGK and PATCHY-SAN. DGK uses the same patterns of the corresponding non-deep graph kernel used to match common substructures. If the pattern distribution is affected by the degree distribution most of those patterns will not match, making it unlikely for DGK to work well on social network data. PATCHY-SAN employs as patterns neighborhood subgraphs truncated or padded to a size k in order to fit the size of the receptive field of a CNN. However, since Niepert et al. [59] experiment with $k = 10$, it is not surprising that they perform worst than SAEN on COLLAB, IMDB-MULTI, REDDIT-MULTI5K and REDDIT-MULTI12K since a small k causes the algorithm to throw away most of the subgraph; a more sensible choice for k would have been the AMND of each graph (i.e. 74, 12, 204 and 162 respectively, cf. Tables 3.1 and 3.2).

Table 3.3 compares the results of SAEN with the best PATCHY-SAN instance on chemoinformatics and bioinformatics datasets. Results obtained by SAEN are comparable with the ones obtained by Niepert et al. [59] on NCI1 and PROTEINS, confirming that SAEN is best suited for large graphs. Moreover, SAEN does not perform well on MUTAG and PTC, as these datasets are too small to afford the highly expressive representations that SAEN can learn and in spite of regularization with L2 we consistently observed significant overfitting.

A2 The compression algorithm has proven to be effective in improving the computational cost of our method. Most of the datasets halved their run-

⁴The AMND for a given dataset is obtained by computing the maximum node degree of each graph and then averaging over all graphs.

times while maintaining the same expressive power. Moreover, we reduced the memory usage on the largest datasets to less than 40% of what would have been necessary without compression.

3.7 Related works

In this section we compare SAEN with other relational learning approaches described in literature.

Many graph kernels compute the similarity between graphs without an explicit feature map. These approaches, however, are not applicable to large datasets as they have a space complexity quadratic in the number of training examples. One advantage of graph kernels such as the Weisfeiler-Lehman subtree kernel (WLST) [78] and the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [15] is the possibility to efficiently compute explicit feature vectors, thus avoiding to solve the optimization problem in the dual. As we explained in Section 3.4, we could in principle turn SAEN into a graph kernel by removing the extraction step; this approach however would be impractical because of the exponential growth of the number of features. Additionally, the corresponding feature map would be fixed before observing data, as it happens with all graph kernels.

Micheli [54] proposed neural networks for graphs (NN4G) as an iterative algorithm that refine node representations in multiple steps. This approach, however, propagates the representations using the connectivity of the input graph, while SAEN enables the user to choose how to propagate representations by choosing a suitable \mathcal{H} -decomposition. Moreover, the SAEN user can specify how the vector encoding should be shifted before the aggregation by using the π -membership types of the \mathcal{H} -decompositions. Furthermore, SAEN can be trained end-to-end with backpropagation while NN4G cannot, as at each iteration of the computation of a state encoding NN4G *freezes* the weights of the previous iterations.

Deep graph kernels (DGK) [95] upgrade existing graph kernels with a feature reweighing schema that exploits embedding derived from CBOW/Skip-gram models. This approach, however, inherits from graph kernels a flat decomposition (i.e. just one layer of depth) for the input graphs. Moreover, the vector representations of the substructures are not trained end-to-end as SAEN would do.

PATCHY-SAN [59] casts graphs into a format suitable for learning convo-

lutional neural networks. This algorithm, however, select a fixed number of neighborhood subgraphs that are then casted to a fixed-size receptive field. These operation involve either padding or truncation operations, that can be detrimental for the statistical performance of the downstream CNN since it throws away part of the input graph. On the other hand SAEN is able to handle structured inputs of variable sizes without throwing away part of the them.

A related neural network architecture was recently introduced by [86] to extend the multi-instance learning framework to data represented as bags of bags of instances. That network can be seen as a special case of SAEN using maximum as the aggregation operator and no π -types (i.e. no shifts).

GRAPHSAGE [36] generates representations for vertices of a graph using an algorithm inspired by the Weisfeiler-Lehman isomorphism test. The approach used by GRAPHSAGE to propagate representations is similar to the application of SAEN’s shift-aggregate operators between level 0 and 1 of ego graph decompositions; unlike SAEN, however, the new node descriptor is built via a single neural network layer instead of a generic extract operation. Furthermore, the algorithm in GRAPHSAGE is forced to use a fixed neighborhood function for all the propagation steps, whereas SAEN is explicitly designed to be able to handle different “part of” relationships at different levels of the hierarchy. Finally, while the special handling of the neighborhood’s center is hardcoded in GRAPHSAGE, in SAEN the more generic π -types mechanism is used to describe the role of each node in the ego graphs, and of each ego graph in the whole graph.

Hamilton et al. [37] proposed a comprehensive review of methods to embed vertices and graphs. While SAEN could, in principle, be adapted for node classification tasks by building a suitable decomposition, in our work we focused on classifying whole graphs. Sum-based approaches such as the ones proposed by [16] and [24] build graph representations by summing node embeddings or edge embeddings; these approaches however cannot represent more complex decompositions and cannot distinguish between vertices with different roles. According to Hamilton et al. [37], other approaches based on graph-coarsening [11, 22] have cubic complexity on number of vertices, and are thus unsuitable for large social networks classification.

3.8 Conclusions

Hierarchical decompositions introduce a novel notion of depth in the context of learning with structured data, leveraging the nested part-of-parts relation. In this work, we defined a simple architecture based on neural networks for learning representations of these hierarchies. We showed experimentally that the approach is particularly well-suited for dealing with graphs that are large and have high degree, such as those that naturally occur in social network data. Our approach is also effective for learning with smaller graphs, such as those occurring in chemoinformatics and bioinformatics, although in these cases the performance of SAEN does not exceed the state-of-the-art established by other methods. A second contribution of this work is the domain compression algorithm, which greatly reduces memory usage and allowed us to halve the training time on the largest datasets.

Chapter 4

Context-dependent claim detection using SAEN

In this chapter we introduce an application of Shift Aggregate Extract Networks to context-dependent claim detection task. Claim detection is the assessment of whether a sentence contains a claim, i.e. the thesis, or conclusion, of an argument; in particular we focus on context-dependent claims, where the context (i.e. the topic of the argument) is a determining factor in classifying a sentence. We show how Shift Aggregate Extract Networks are able to take advantage of contextual information in a straightforward way and some preliminary results that indicates how this approach is viable on real world datasets.¹

4.1 Introduction

Argumentation studies the way humans debate and reason. Having its roots in logic, rhetoric, and philosophy, this ancient discipline has recently become a hot topic also for computer science, giving birth to the research field of computational argumentation. The problem of automatically extracting arguments from unstructured text has defined a novel area of interest for artificial intelligence and computational linguistics, called argumentation

¹A manuscript based on this chapter is under preparation as “SAEN for Argumentation Mining” [4].

mining [51]. Although several, different formal definitions of an argument have been given throughout the years, in this chapter we will mainly consider the general framework described by Walton [93] where an argument is defined by a claim, a set of premises, and an inference from the premises to the claim.

A wide range of artificial intelligence and machine learning techniques have been recently employed to address this challenging task. In particular, they have been successfully applied to *specific domains* like legal documents [56], microtexts [64], Wikipedia articles [47, 48] or student essays [83]. On the other hand, the problem of cross-domain generalization is still open [20]. Nevertheless, some initial study on cross-domain claim identification has shown that there are properties, at least on the lexical level (e.g., the occurrence of keywords such as “should”), that some diverse domains seem to share [20], which further motivates the ambitious challenge of detecting arguments and their relations in texts of any kind and genre.

There is a wide range of applications that have been triggered by advancements in argumentation mining. Recent works have been proposed in the context of claim retrieval [72], argument synthesis [6], claim classification [34], analysis of political debates [12, 49], automatic essay scoring [60].

A crucial task in argumentation mining is claim detection. The goal there is to assess whether a sentence contains a claim, which is the thesis, or conclusion, of an argument. Typically, the machine learning classifiers that have been employed in claim detection consider the sentence to be classified as argumentative or not independently of the rest of the document [47, 48, 82]. Only a few recent approaches have tried to exploit contextual information² to describe a sentence for this task.

The majority of such approaches explicitly encodes contextual information in specific features, for example by considering the presence of known indicators or discourse markers in preceding and subsequent sentences [57, 65, 82, 83]. Most often, methods are proposed for domains where documents follow predefined structures, like in the case of student essays, legal texts, or scientific articles. There, some early works in argument mining considered argumentative zoning [85] to identify and classify sections on scientific

²The word *context* in argumentation mining has sometimes been used as a synonym of *topic* [47, 48]: context-dependent claim detection has been defined as the extraction of claims from a text, when a topic is given in advance. In this work, by indicating the *context of a sentence* we mean the information regarding the document in which the sentence appears.

documents using features such as the location of a sentence within a document and within subsections and paragraphs, sentence length, whether the sentence contains a word from the title, etc. [63]. Recently, knowledge of the document structure has been used by Persing and Ng [65], Stab and Gurevych [83], and Wei et al. [94] in domains such as student essays, where global constraints are enforced to tie the output of the system to the document structure. All these works heavily rely on handcrafted features. These would have to be manually adapted to and designed for each new sample of data, which is hardly sustainable [25].

In [35], claim detection is performed at sentence level with structured output support vector machines [88], hence by taking into account the sequential correlation among sentences. Even in this case, several handcrafted features are employed, such as the relative position of the sentence within the paragraph and the document, the output of a sentiment classifier, a semantic role labeler, and a semantic coreference chain resolver. In order to take into account contextual information, to describe each sentence, the feature vectors of a few preceding and subsequent sentences are also used.

In other cases, the topic has been used as an external, additional piece of information to be exploited when looking for claims [47]. While this approach is certainly relevant in tasks where the topic to be debated is given in advance, yet it presents two limitations: first, the outcome of the argumentation mining system would be affected by the way the topic is phrased, which might not be desirable; second, in many cases the assumption of knowing the topic in advance simply does not hold.

In this chapter, we show that the context in which sentences are embedded provides effective features for argumentation mining, even when the document structure is unknown and the topic is unknown. We also show that contextual information can be exploited by applying the same method across different domains. In particular, we propose a novel strategy for claim detection in which we learn two separate representations, one for the sentence to be classified, and one for the rest of the document where that sentence occurs. The second representation is meant to provide topical contextual information so that the classifier may detect whether the claim of an argumentative sentence does actually fit the topic being debated.

We learn sentence and context embeddings by exploiting an architecture based on shift aggregate extract networks (see Chapter 3). We exploit the hierarchical approach of SAEN to handle context in a very natural way: the

target sentence and the remaining portion of the document serve as the top level of the hierarchy, while fragments of dependency graphs and individual words are used at the lower levels.

4.2 Argumentation Mining

Mining arguments from text is a challenging problem that encompasses several tasks, or steps. The way these tasks are formulated and addressed strongly depends on a variety of factors, including the underlying argument model, the document genre, the application domain. Generally speaking, argumentation mining systems have to identify argumentative entities (e.g., premises and claims) and the relations between them [51]. Many approaches address these tasks following a pipeline scheme, while recently there has been some effort in jointly addressing multiple tasks [35, 65]. In this work we focus on the detection of sentences containing claims, which is still considered the main issue in argumentation mining [20]. The task is typically formulated as a sentence classification task, and many different machine learning and natural language processing approaches have been proposed to address it, including logistic regression or support vector machines with rich feature sets [47, 82], tree kernels [48], convolutional and recurrent neural networks with word embeddings [20].

Very often, the chosen methods have been adapted and calibrated on the specific application that was targeted. There have been mainly two reasons behind this rationale. The first reason is that applications are typically linked to a specific document genre, such as legal documents for argument detection and retrieval in law [56, 63], or user-generated content in web resources [35, 47]. This in turn implies that specific features are commonly designed for each task (and thus, for each method), such as the presence of discourse markers, or the syntactical structure of sentences and paragraphs. The second reason has to do with corpora. Datasets, in fact, are hard to build in argumentation mining, since they require a significant effort in defining guidelines and performing annotations. This activity is made particularly difficult by the inherent ambiguity and subjectivity in the definition and identification of an argument [35, 51].

In this chapter we aim at exploiting contextual information in order to detect sentences containing claims. Context is a very general term, and it has been used from different perspectives in the literature. Levy et al. [47]

first introduced the term *context* with the meaning of *topic*, with the aim of identifying context-dependent (i.e., topic-dependent) claims. Their approach proposes to directly use a definition of the topic of interest within the claim detector, so as to retrieve only those claims that refer to that topic. Yet, this approach suffers from the limitations mentioned in Section 4.1, that is that the phrasing of the topic would affect the classifier, and moreover knowing the topic in advance is not always a reasonable assumption. Habernal and Gurevych [35] exploit contextual information in a different way, that is by employing a structured support vector machine (namely, SVM-HMM) to tag sentences. In this way, the classification of a sentence will depend also on the classification of neighbor sentences. In addition, they describe each sentence with features also coming from the preceding and subsequent sentences. More recently, Eger et al. [25], Niculae et al. [58], Persing and Ng [65], have presented different approaches that share the common idea that some tasks in argumentation mining should be jointly addressed in a multi-task setting. Structured support vector machines and recurrent neural networks are proposed within this context. In particular, Eger et al. [25] remark the limitations of exploiting hand-crafted features in computational argumentation, highlighting how deep neural architectures have the potential to learn rich and effective representations.

4.3 Context-aware hierarchical decomposition

We propose an approach to automatically detect context-dependent claims by exploiting the contextual information given by the documents in which each sentence appears. More specifically, we take advantage of the flexibility of \mathcal{H} -decompositions (described in Section 3.2) to build *context-aware hierarchical decompositions* that make use of contextual information in a straightforward way. To do so, we introduce the concept of *contextualized sentences*.

Definition 9 (Contextualized sentence). *Given $D = \{s_i\}_{i=1}^N$ a document represented as a set of sentences, we define the set of contextualized sentences of D as the set of pairs $\{(s_i, D \setminus \{s_i\})\}_{i=1}^N$.*

If we represent sentences as graphs (perhaps by using a dependency parser), we can easily build context-aware variants of existing \mathcal{H} -decomposition by stacking an additional level on top of the hierarchy. This level will consist of contextualized sentences, and each of them will be decomposed in the

graph representing the sentence itself (with π -type “self”) and the graphs representing each other sentence in the document (with π -type “context”). The computation defined in Section 3.4 will then be able to combine the representation for each sentence with an aggregated descriptor of the rest of the document.

For this work we focused on the context-aware extension of the ego graph decomposition described in Section 3.3.1, and more specifically:

- Level 3 consists of all the contextualized sentences contained in a document;
- Level 2 consists of all the sentences contained in a document, represented as graphs obtained from a dependency parser;
- Level 1 and Level 0 are defined in the same way of EGD;
- A function $\mathbf{x} : S_0 \rightarrow \mathbb{R}^p$ associates to each vertex (word) in level 0 a corresponding p -dimensional word vector.

Context-aware hierarchical decompositions have several advantages. First, sentence and context representations are learned at the same time, without requiring a separate preprocessing step to extract contextual information. Second, the domain compression algorithm (described in Section 3.5) greatly reduces the temporal complexity of the algorithm when classifying all the sentences in a document. In this case, in fact, the decomposition of all the contextualized sentences would generate N^2 objects. However, most of them would be duplicates, as each original sentence would appear one time with π -type “self” and $N - 1$ times with π -type “context”. Therefore, the domain compression algorithm would be able to collapse all these identical structures and reduce the number of them back to N . Figure 4.1 exemplifies this process with a simple context-aware hierarchical decomposition.

4.4 Experimental Evaluation

We performed an experimental evaluation of our approach on three different datasets: IBM Wikipedia Corpus, Consumer Debt Collection Practices (CDCP), and Persuasive Essays.

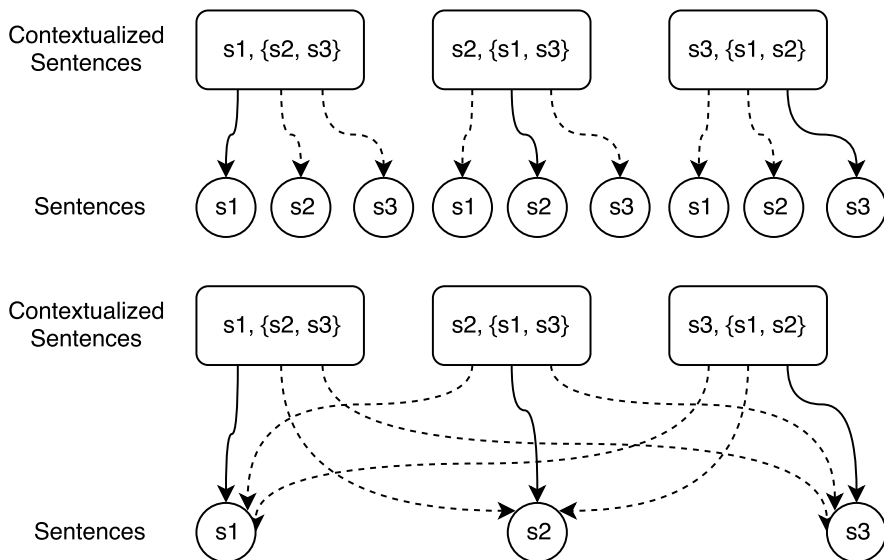


Figure 4.1: Pictorial representation of top two levels of a context-aware hierarchical decomposition before and after the application of the domain compression algorithm. Solid lines indicate part-of relationships with π -type “self”, dashed lines indicate part-of relationships with π -type “context”.

Table 4.1: Statistics of the datasets used in our experiments.

Dataset	Documents	Sentences	Avg. sentences for doc.
CDCP	731	4932	6.75
ESSAYS	402	7124	17.72
IBM	522	76869	150.56

4.4.1 Datasets

IBM Wikipedia Corpus

The first dataset that we used is the IBM Wikipedia Corpus [1, 70].³ This dataset consists of 522 Wikipedia articles, each one of them annotated for claims and evidences. Overall, the corpus contains 2080 claims out of a total of 76869 sentences. Following the conventions described in [70], the articles are associated with 58 topics, of which 39 are used for training and testing via a leave-one-topic-out cross validation procedure while the other 19 are used as validation set.

As many articles are associated to multiple topics, the same article could appear both in the training and in the test sets while performing the cross validation. To address this problem, we adopted the following strategy for each of the 39 runs:

- when a document appear both in the test set and in the training and/or validation sets, we remove it from the training/validation sets;
- when a document appear both in the validation set and in the training set, we remove it from the validation set;
- we remove duplicate documents from the training and the validation sets.

Consumer Debt Collection Practices (CDCP)

As a second benchmark, we use the recently released CDCP [58] corpus⁴ which consists of a collection of user comments from the eRulemaking website, about rule proposals regarding Consumer Debt Collection Practices by the Consumer Financial Protection Bureau. The dataset presents phrases already segmented and labeled according to five different classes, namely POLICY, VALUE, TESTIMONY, FACT, and REFERENCE. According to the dataset guidelines, the first two classes (POLICY and VALUE) represent subjective judgements and interpretations, while the remaining three categories typically support such conclusions. We thus formulated a binary classification task, where POLICY and VALUE make up the positive class. Overall, the corpus contains 2,997 positive sentences out of 4,932. A single train/test split is provided, and segmented sentences are already given as well.

³Available at https://www.research.ibm.com/haifa/dept/vst/mlta_data.shtml

⁴Available at <http://joonsuk.org>.

Persuasive Essays

Finally, as a third dataset we employed the Persuasive Essays corpus developed by Stab and Gurevych [83]. This corpus has been widely used in many argumentation mining papers, as it is one of the few datasets with a complete annotation that comprises the detection of argumentative entities, their relations, and the stance of such relations. The corpus consists of 402 persuasive essays, split into a training set of 322 documents, and a test set of 80 documents. Argumentative entities are annotated as claims, major claims, and premises. For our purpose, we considered claims and major claims as positive class (as in [20]), which resulted in a total of 2,117 positive sentences out of 7124 examples.⁵

4.4.2 Experiments and results

We applied the context-aware ego graph decomposition described in Section 4.3 with radius 1 to each dataset, and we trained a shift aggregate extract network on each of them. For each dataset we followed the experimental setting described by the conventions described by their authors: leave-one-topic out for IBM Wikipedia Corpus and a single train/test split for CDCP and Persuasive Essays.

We compare the results obtained by our method (SAEN+C) with those obtained by the method described by Lippi and Torroni [50] (SSTK) and with those obtained by SAEN with a context-unaware ego graph decomposition (SAEN). For the experiment on IBM Wikipedia Corpus we considered for SAEN and SAEN+C the average output of four identical SAENS initialized with different random weights, as we noticed that these architectures were very sensible to the choice of the initial state.

In Table 4.2 we report the results obtained on IBM Wikipedia Corpus in terms of AUROC and $F1@200$ ⁶, in Table 4.3 we report the results obtained on CDCP in terms of AUROC and F1, and in Table 4.4 we report the results obtained on Persuasive Essays.

Our context-aware approach outperforms both SSTK and SAEN with a context-unaware ego graph decomposition. On CDCP and Persuasive Es-

⁵There is a tiny difference with respect to the number of sentences reported in [83], although we employed the same software (DKPro) for segmentation.

⁶Please note that the results reported here are not directly comparable to those published in other works, due to the removal of duplicate articles with the procedure described in Section 4.4.1

Table 4.2: Comparison of results on the IBM Wikipedia corpus.

Method	AUROC	$F_1@200$
SSTK	0.784	0.163
SAEN	0.812	0.149
SAEN+C	0.821	0.176

Table 4.3: Comparison of results on CDCP.

Method	AUROC	F1
SSTK	0.837	0.808
SAEN	0.856	0.846
SAEN+C	0.825	0.829

says, however, our approach does not perform well. This is probably caused by the fact that documents contained in these datasets do not provide enough contextual information, as the average number of sentences per document (shown in Table 4.1) is an order of magnitude smaller in these datasets with respect to IBM Wikipedia Corpus.

4.5 Conclusions

Cross-domain claim identification is still an open challenge. In this chapter we proposed a machine learning approach that leverages the contextual information embedded in the document in which a sentence appears. We describe how to build context-aware extensions to existing \mathcal{H} -decompositions, an approach that allows shift aggregate extract networks to exploit the context in a straightforward way. Moreover, decompositions built with this approach can be easily compressed via SAEN’s domain compression algorithm, greatly reducing the cost of classifying all the sentences contained in a document.

Table 4.4: Comparison of results on Persuasive Essays.

Method	AUROC	F1
SSTK	0.794	0.605
SAEN	0.767	0.556
SAEN+C	0.766	0.525

Preliminary results show that our approach is able to outperform existing methods when applied to documents that contains sufficient contextual information. Our method, however, has still two issues that need to be addressed. First, the instability of our approach required us to run our algorithm multiple times. To deal with this problem we would need to investigate the dynamics of our algorithm when dealing with complex hierarchical structures. Our conjecture, in fact, is that the computation of SAEN requires special care when choosing how to initialize weights and how to regularize the network, and that the existing approaches for these problems have to be tuned to work with our kind of hierarchical networks. Second, results show that SAEN performs worse than existing method on IBM Wikipedia Corpus when contextual information is not taken into account. This problem could be addressed by building a claim-detection-specific \mathcal{H} -decomposition: in our experiments we only used the context-aware extension of the ego graph decomposition, which is very general and thus is not suited to take advantage of additional information deriving from dependency graphs. A possible extension of our method could explore how to embed this additional information, perhaps by embedding part-of-speech and dependency tags in a custom decomposition.

Part II

Novelty detection in cardiography

Chapter 5

A machine learning approach to assess normalcy of aortic size

This chapter describe a machine learning approach to aortic size normalcy assesment. The definition of normalcy is crucial when dealing with thoracic aortas, as a dilatation of its diameter often precedes serious disease. In this contribution we build a normalcy estimator based on OC-SVM and fitted on a cohort of 1024 healthy individuals aging from 5 to 89 years, and we compare its results to those obtained on the same set of subjects by an approach based on linear regression. We also build a second estimator that combines the diameters measured at multiple levels in order to assess the normalcy of the overall shape of the aorta.¹

5.1 Introduction

Thoracic aorta (TA) is a geometrically complex structure which is routinely assessed by standard two-dimensional (2D) transthoracic echocardiography (TTE) [32, 67]. Aortic dilatation is an important predictor of outcome [18, 19, 43], and its detection prompts the need for accurate clinical and imaging follow-up [7, 40, 67], in order to prevent catastrophic events

¹This chapter will be submitted as “Two-dimensional echocardiographic aortic size in 1024 healthy individuals aged 5 to 89 years. Development of a machine learning approach to assess normalcy” to *Journal of American College of Cardiology: Cardiovascular Imaging* [27].

(such as dissection) by appropriate therapeutic choices including drugs and prophylactic surgery. Thus, an accurate definition of normalcy and the availability of tools for assessing normalcy are crucial for diagnosis and follow-up strategies.

In order to be usable in a practical general context, a normalcy calculator should fulfil the following desiderata. First, it should assess the aortic size at several levels (at least sinuses of Valsalva, sinotubular junction, and proximal ascending aorta). Second, it should predict normalcy on a single patient basis taking into account the influence of demographics and anthropometrics on TA size. Third, it should also be uniformly applicable to the whole range of patients, thus avoiding the introduction of age or body-size groups that make decisions difficult for subjects situated near the range extrema.

Although a large body of literature exists on the description of normal limits of TA size, no standard normalcy calculator meeting the three above desiderata is currently available. There are several reasons why TA normalcy assessment is still an unsettled matter. To begin with, the findings in previous studies are not immediately comparable because of different demographic and anthropometric characteristics of the study populations, different TTE modes, and different strategies for measurements including interfaces (leading-to-leading vs. inner- to-inner), and timing (end-systole vs. end-diastole). Moreover, the usability of the results of these studies for practical purposes may be limited since some of them (e.g. [73]) assessed the aortic size at only one level of the TA (usually sinuses of Valsalva). Furthermore, most studies provide ranges of normalcy or a graphical approach to assess normalcy, while relatively few provided algorithms to predict normalcy on a single patient basis taking into account the influence of demographics and anthropometrics on TA size. Overall, the available algorithms are only applicable to specific cases: those assessing the aortic size at each of the 4 TA sites are confined to neonates, infants and young adults, or provide graphic nomograms separated for gender, body size and age groups; while those predicting only the aortic root are based on different echocardiographic conventions for childhood and adulthood.

Many previous approaches to establish nomograms of aortic diameters are based on linear regression modeling via ordinary least squares. Diameters are predicted from demographic and/or anthropomorphic explanatory variables such as height, weight, age, and sex, or derived attributes such as body surface area (BSA) and body mass index (BMI). Regression models

(typically one separate model per diameter and sometimes separate models for different genders) assume that the noise is additive and normally distributed with zero mean and constant variance (homoscedasticity). Once fitted, these models produce normalcy calculators based on Z -scores. More precisely, Z is calculated as the difference between the measured and the predicted diameters, divided by the standard deviation estimated from the mean squared error on the training data. The calculators are eventually used to infer the abnormality of diameters (e.g. using the criterion $Z > 2$).

Several limitations of these approaches to normalcy have been discussed [53]. First, the relationship between predictors and a certain aortic diameter is not necessarily linear. To correct this, some methods assume that linearity holds in the log-log space [29], while others have used polynomial models [17]. For example, in [29] it is proposed to predict $\log d = \beta_1 \log \text{BSA} + \beta_0$. A second problem is that the homoscedasticity assumption is often violated and not accounted for [53]. While methods that operate in the log-log space somewhat do take heteroscedasticity into account (when mapping back to the original spaces, variance increases with the independent variable), the way variance varies with predictors remains constrained by the model assumptions. A third issue is that data might not satisfy the model assumptions of normally distributed additive noise [53].

Paradoxically, success and effectiveness of these methods are linked to their inaccuracy in predicting aortic diameters. Indeed, if a diameter of a new subject was predicted with a small error (by a very accurate model), then the resulting Z -score would be small. While this is desirable for healthy subjects, it may be problematic for pathological subjects, potentially yielding a low sensitivity when using the calculator as a diagnostic tool. Indeed, few existing normalcy studies have extensively tested the diagnostic power of the proposed calculators in terms of sensitivity and specificity, measured on an independent test set of healthy and pathological subjects.

In this chapter, we aim at developing a general tool for TA normalcy measured at 4 levels by 2D-TTE, applicable to a large group of healthy individuals ranging from pediatric to elderly. For this purpose, we considered two alternative approaches: a conventional approach based on linear regression models, and a novelty detection approach based on one-class support vector machine (OC-SVM) [77]. Novelty detection techniques are common in the context of industrial applications (see, e.g., [66] for a review) and they have also found a few successful applications to some medical con-

texts [28, 80, 84]. However, to the best of our knowledge, they have never been applied before in the context of TA normalcy. As an advantage over conventional approaches, OC-SVM is able to delimit regions of high density without making any assumption about the underlying probability distribution, thus potentially overcoming some of the above mentioned limitations of regression models. As a second advantage, OC-SVM allows normalcy to be assessed for the whole aorta morphology, taking all level measurements into account simultaneously, while conventional linear regression typically assesses normalcy for individual levels separately. We validated and compared the two approaches using an independent cohort of subjects deemed at risk of having aortic dilatation.

5.2 Materials and methods

5.2.1 Study population

Healthy individuals aged 5 years or older were prospectively and consecutively identified and enrolled in 3 independent echocardiographic laboratories if they had: normal 12-lead ECG, left ventricular (LV) ejection fraction $\geq 55\%$ and normal wall motion score index. Subjects were excluded if they were first-degree relatives either of patients with bicuspid aortic valve or TA aneurysm/dissection, or Marfan syndromes or related disorders (also if the subject him/herself was free from that disease) and/or had:

- a. arterial systemic hypertension and/or were on active anti-hypertensive treatment;
- b. overt coronary artery disease (defined as previous acute coronary syndrome and/or revascularization procedures, or positive stress tests of inducible ischemia);
- c. primary cardiomyopathy and/or genetic cardiovascular disease;
- d. congenital heart disease;
- e. mitral valve prolapse, mitral or aortic valvular insufficiency of higher degree than trivial, valvular stenosis of any degree, or any previous cardiac or vascular surgery or interventional procedure (including ablation of accessory pathways);

- f. previous chemotherapy and/or chest radiotherapy;
- g. documented episodes of atrial fibrillation or flutter (even if paroxysmal and remote), either complex or frequent (i.e. > 10 ectopic beats/hour at Holter monitoring) supra-ventricular or ventricular arrhythmias;
- h. any kind of cardiovascular therapy;
- i. previous cardioembolic stroke, including transient ischemic attacks;
- j. diabetes mellitus or any kind of endocrinologic disorder.

As an independent cohort of subjects deemed at risk of having TA dilatation, we enrolled patients with either clinical diagnosis of Marfan syndrome or related disorders, or bicuspid aortic valve or consecutive outpatients with aortic diameter > 40 mm at any TA levels studied in the same laboratories by the same echocardiographers.

5.2.2 Echocardiography

Comprehensive echocardiographic examinations were performed using commercially available systems equipped with a multifrequency phased-array transducer according to a predefined protocol for the performance of the echocardiographic exam, its storage, review and measurement, by 3 board-certified cardiologists with more than 10 years of clinical experience in performing and interpreting echocardiographic examinations, during ECG – monitoring for proper timing of all the measures. Left ventricular (LV) end-diastolic and end-systolic volumes and ejection fraction (EF) were calculated using the modified biplane Simpson’s rule; LV mass was calculated using linear measurements of 2D recordings of the left ventricle [7]. Left atrial volume was assessed by the biplane area-length. Pulsed Doppler of LV mitral inflow was recorded in the apical 4-chamber view at the tips of the mitral valve: early (E) and atrial (A) peak velocities (m/s) were measured and E/A ratio was calculated. Early diastolic e' velocities were measured at the septal and the lateral site of the mitral annulus by pulsed Tissue Doppler averaged. The ratio between transmitral E and average e' (E/e' ratio) was calculated.

The aortic diameters were measured in 2D-TTE mode at 4 levels [i.e.: aortic annulus (AAn), sinuses of Valsalva (SoV), sinotubular junction (SJ),

and proximal ascending aorta (PAA)]. Care was used to adjust the parasternal long-axis view to align the echocardiographic plane with each of the 4 aortic levels in order to obtain the largest aortic diameters; magnified views were used for greater precision. All measurements were made at end-diastole, perpendicular to the long axis of the aorta, using the leading-edge to leading-edge technique. Specific measurements were made by the average of 3 to 5 cardiac cycles. Height (in m) and weight (in Kg) were measured at the time of the TTE; body mass index (BMI) was computed as weight/height squared and body surface area (BSA) calculated by the Du Bois and Du Bois formula. Systolic and diastolic blood pressure were measured using a cuff sphygmomanometer at the end of the examination.

To assess reproducibility, the main investigator repeated the analysis after a period of 2 weeks. A second independent observer, blinded to principal observer's results, performed the measurements in a randomly chosen subgroup of 50 subjects from each laboratory. Interobserver variability were studied as intraclass correlation coefficients (ICCs). Reproducibility of the entire set of aortic measurements was good to excellent at each aortic level, without any difference between echo-labs. In particular, ICCs was 0.92 for the AAn, 0.98 for the SoV, 0.95 for the STJ, and 0.98 for the ascending aorta.

5.2.3 Regression analysis model

Classic multivariate regression analysis was employed to predict aortic diameters from age, sex, and BSA. As in previous approaches, the mean squared error on training data was used to estimate (homoscedastic) variance and to derive Z-scores. Unlike previous approaches (in particular [29, 73] that focused on specific age groups, and trained separate gender-specific models) a single model was trained on all the available data ($n = 1024$), yielding a single normalcy calculator that is applicable to any individual for the 4 different aortic levels. We also defined a global Z score as the maximum Z score among the 4 levels.

5.2.4 One-class support vector machine model

The one-class support vector machine (OC-SVM) is a machine learning method that estimates the support (i.e. the high-density region) of the unknown joint probability p over a given set of variables [77]. Given a data

set of instances drawn from p , the learning algorithm infers a real valued function f such that for any realization x of the variables, $f(x) > 0$ if x belongs to the support of p . In the following, $f(x)$ will be referred to as the O-score. In our context, x is a vector representing normalized (by dividing for the maximum observed value) age, sex, BSA, and the single or all 4 aortic diameters as our goal is to estimate normalcy of the combination of these features. When x contains age, sex, BSA, and one individual aortic diameter, $f(x)$ will be referred to as the local O-score (relative to the particular chosen diameter). When x includes all four diameters, $f(x)$ will be referred to as the global O-score. To the best of our knowledge, this is the first normalcy indicator that takes into account the whole aorta morphology. The decision function is computed as

$$f(x) = \sum_{i=1}^n \alpha_i k(x_i, x) - \rho \quad (5.1)$$

where the index i ranges over the subjects in the training set, k is a kernel function that measures the similarity between two vectors of variables, and α_i and ρ are coefficients determined by the learning algorithm. Unlike linear regression (that results in one coefficient for each variable and one intercept) the nature of this model makes it difficult to report the results of the training algorithm in the paper.² The OC-SVM approach offers several advantages:

- it does not rely on any assumption on the density of interest and it is therefore not affected by problems such as heteroscedasticity and residues that are not normally distributed;
- it can exploit a kernel function to measure the similarity between a new test subject and the subjects in the study population; kernel functions implicitly map a realization x to a point in an infinite dimensional feature space, thus allowing to model complex nonlinear relationships among variables;
- as mentioned above, the global O-score can take into account the interplay among aortic diameters (thus incorporating global morphological information of the whole aorta), by including them in x together with all other demographic and anthropomorphic variables.

²For this reason, we make available a calculator implementing the above formula at <http://aorta-normalcy.dinfo.unifi.it>.

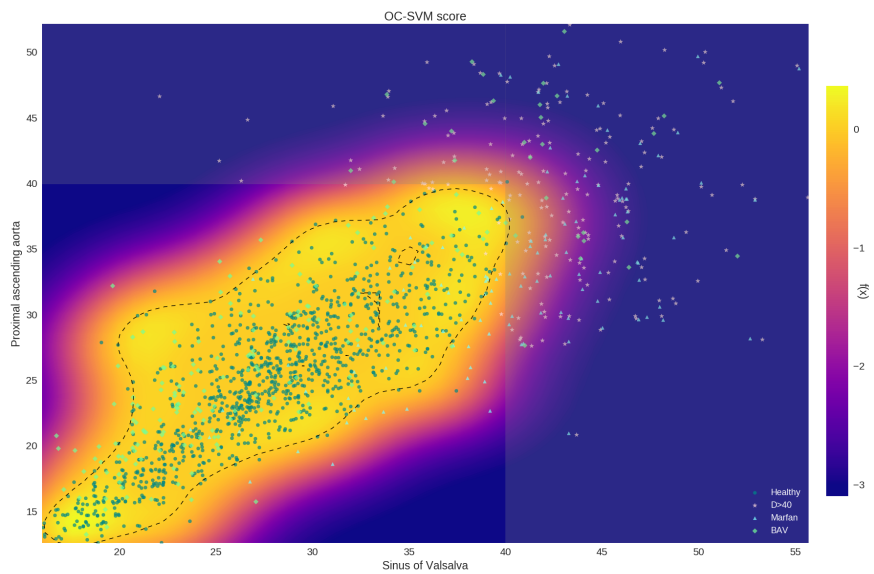


Figure 5.1: Example of heatmap of O-score obtained by a OC-SVM model that predicts normalcy by using only PAA and SOV diameters. The solid line is the contour associated with O-score = 0. Note that this contour delimits a closed region thanks to the use of a radial basis function (RBF) kernel: inside the yellow region, scores are negative thus indicating normalcy.

In the impossibility to represent the seven dimensions feature space of our OC-SVM model (4 aortic levels, BSA age and sex), as an illustrative example, Figure 5.1 shows a heatmap of O-score obtained by a OC-SVM model that predicts normalcy by using only PAA and SOV diameters. As a reference, the TA dilatation region defined by the current guidelines (> 40 for at least one aortic segment) has been slightly darkened. Inside the yellow region, O-scores are positive, indicating normalcy. The solid line is the contour associated with O-score = 0. Note that this contour delimits a closed region thanks to the use of a radial basis function (RBF) kernel. By contrast, linear regression models are unable to delimit closed regions: If we used linear regression on the data of Figure 5.1, the normalcy region would have been delimited by two parallel lines, resulting in many clearly abnormal cases (e.g. patients with PAA and SoV approximately equal but both larger than 40 mm) being predicted as normal. It should be remarked a second important difference between Z-scores and O-scores: the former has a direct probabilistic semantics (under the assumption of Gaussian residuals); by contrast, the absolute value of the latter is not directly interpretable as a population percentile but still it represents an indication of the model's confidence about normalcy (if positive) or abnormality (if negative). OC-SVM has two parameters to be tuned: ν (an upper bound to the number of outliers) and γ (the RBF kernel width). We set $\nu = 0.04$ since this is roughly the proportion of population that would have a Z-score > 2 when using the traditional linear regression approach; γ was set to 2 in order to obtain a high sensitivity in detecting pathological subjects (one diameter > 40 mm) without deteriorating specificity above the 4% level.

5.2.5 ROC analysis

Receiver operating characteristic (ROC) curves were used to assess and compare the discriminatory power of linear regression and OC-SVM in distinguishing between pathological and normal individuals. Significance in the pairwise comparisons between the areas under the curve (AUC) was assessed by a DeLong test [23] using the pROC package [71]. For each model, the best cutoff that maximized the sensitivity-specificity sum was determined and gives an indication of the optimal model's sensitivity and specificity. A value of $p < 0.05$ was chosen as the cut-off level for statistical significance.

Table 5.1: Demographic and echocardiographic characteristics of the 1024 healthy subjects.

Characteristics	Median (IQR)
Age (years)	30 (16-48)
Sex F n (%)	553 (54)
Height (cm)	165 (156-173)
Weight (Kg)	61 (50-73)
Body mass index (Kg/m ²)	21.99 (19.53-25)
Body surface area (m ²)	1.66 (1.50-1.86)
Systolic blood pressure (mm Hg)	120 (110-130)
Diastolic blood pressure (mm Hg)	70 (65-80)
Heart rate (beats per minute)	74 (66-85)
Left atrial volume index (ml/m ²)	21.62 (16.8-27)
Left ventricular end-diastolic volume index (ml/m ²)	55 (46.8-64)
Left ventricular ejection fraction (%)	64 (61-68)
Left ventricular mass index (g/m ²)	70 (60.1-82)
E/A	1.5 (1.15-1.98)
E/e'	5.9 (5.1-6.9)

5.3 Results

5.3.1 Characteristics of normalcy and deemed-at-risk cohorts

We prospectively enrolled 1024 healthy subjects, aging from 5 to 89 years, mostly female ($n = 553$, 54%). Two-hundred fifty (24.4%) were ≤ 15 years, $n = 317$ (30.9%) were from 16 to 35 years, $n = 308$ (30.1%) were from 36 to 55 years, and $n = 149$ (14.6%) were ≥ 56 years. In Table 5.1 demographic and echocardiographic characteristics of the study group are reported. Women were significantly older than males [43 (21-55) years vs. 18 (12-36) years, respectively, $p < 0.0001$] and had smaller body size, left atrial and LV indexed volumes, and smaller LV mass index (data not shown). We identified (A) 82 patients with unequivocal increased aortic size (LG), i.e. > 40 mm in at least one aortic level, and 404 patients deemed at risk of aortic dilatation due to either, (B) Marfan syndrome ($n = 115$), or (C) patients with isolated congenital bicuspid aortic valve ($n = 289$). Normal aortic size, raw and indexed for BSA, are reported in Table 5.2 as median and

Table 5.2: Aortic size at aortic annulus, sinuses of Valsalva, sinotubular junction and proximal ascending aorta level (raw diameter and diameter indexed for BSA) of the 1024 healthy subjects.

Aortic levels	Raw diameter Median (IQR)	BSA-indexed diameter Median (IQR)
Aortic annulus (mm)	19.0 (17.0-21.0)	11.75 (10.80-12.93)
Sinuses of Valsalva (mm)	28.0 (24.0-31.2)	17.34 (15.62-19.33)
Sinotubular junction (mm)	24.2 (20.5-27.0)	14.71 (13.38-16.24)
Proximal ascending aorta (mm)	25.6 (21.0-29.0)	15.57 (13.87-17.59)

Table 5.3: Association of age, gender and BSA with aortic diameter at the 4 investigated levels (*: $p < 0.001$, §: $p < 0.005$).

	Aortic annulus	Sinuses of Valsalva	Sinotubular junction	Proximal ascending aorta
Intercept	9.885	13.092	8.116	8.865
Age (yrs)	0.015*	0.117*	0.113*	0.164*
BSA (m2)	5.942*	7.533*	7.708*	7.017*
Female Sex	-1.700*	-2.295*	-1.066*	-0.607§

IQR. All the investigated parameters, namely age, gender, height, weight, and BSA significantly affected aortic size at each of the 4 aortic level. For consistency with the practical aims of our study in clinical echocardiography and for introducing a lower number of variables in the model without losing information, we utilized age, gender, and BSA for subsequent analyses (see Table 5.3).

The effect of age was predominant at each aortic level before 20 years, and was negligible afterward only for AAn (Figures 5.2a, 5.2c, 5.2e, 5.2g). BSA was a significant determinant of each of the 4 aortic level, with a particularly large scatter of distribution for SoV, SJ and PAA (Figures 5.2b, 5.2d, 5.2f, 5.2h).

Based on this multivariable regression analysis a Z-score calculator for each of the 4 TA levels is provided. Taking into consideration the same variables, a O-score calculator for every single TA region, and a global O-score, considering the interplay among different aortic diameters is also provided. In Figure 5.3 outputs of both methods are represented.

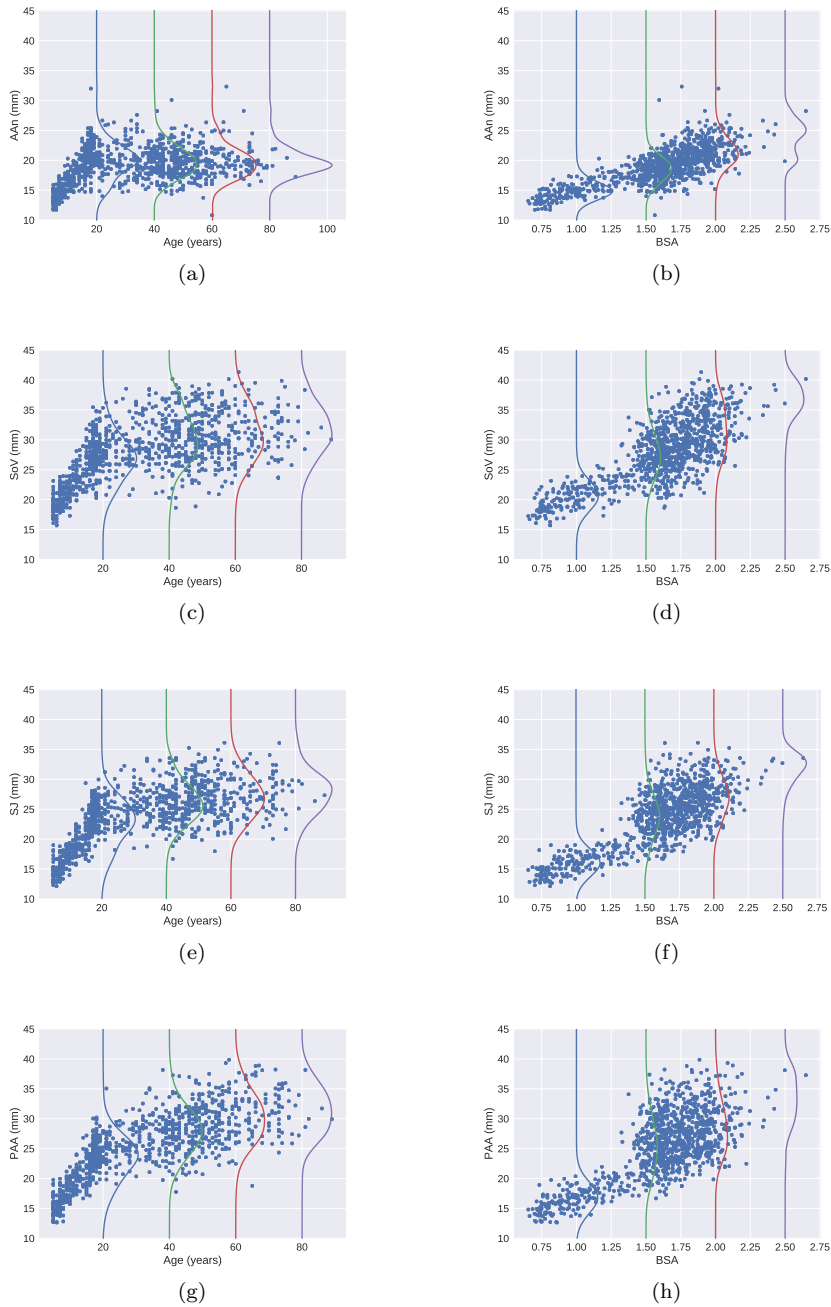
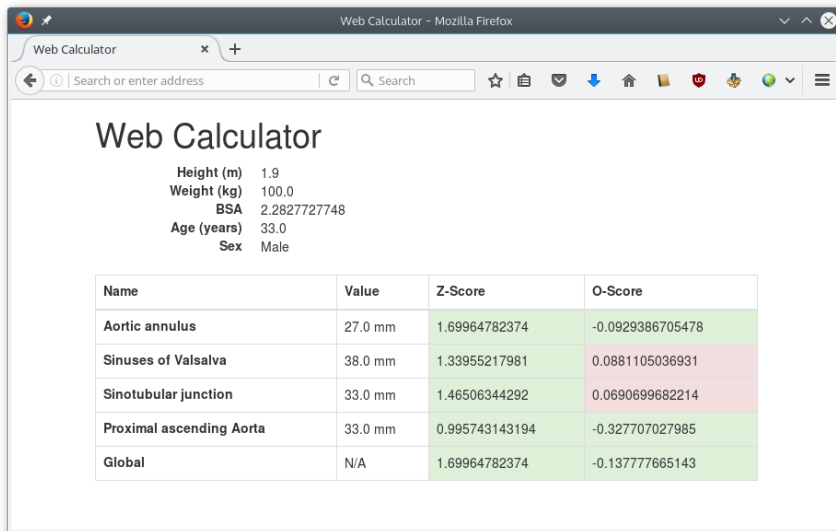


Figure 5.2: Representation of effect of age and BSA at each aortic level [Aortic annulus (AA), Sinuses of Valsalva (SoV), Sinotubular junction (SJ), and Proximal ascending aorta (PAA)].



Web Calculator

Height (m) 1.9
Weight (kg) 100.0
BSA 2.2827727748
Age (years) 33.0
Sex Male

Name	Value	Z-Score	O-Score
Aortic annulus	27.0 mm	1.69964782374	-0.0929386705478
Sinuses of Valsalva	38.0 mm	1.33955217981	0.0881105036931
Sinotubular junction	33.0 mm	1.46506344292	0.0690699682214
Proximal ascending Aorta	33.0 mm	0.995743143194	-0.327707027985
Global	N/A	1.69964782374	-0.137777665143

Figure 5.3: Representation of the outputs of the multivariable regression model Z-score calculator and OC-SVM model O-score calculator.

Table 5.4: Differences in the attribution of normalcy or dilatation of aorta by the application of linear regression and OC-SVM models in deemed-at-risk individuals ($n = 404$).

Total $N = 404$	Z-score	O-score
Normalcy (Z-score<2, O-score<0 in all aortic levels)	111 (27.5%)	177 (43.8%)
Pathologic (Z-score>2, O-score>0 in at least 1 aortic level)	293 (72.5%)	227 (56.2%)

5.3.2 ROC analysis by linear regression model and OC-SVM model

As a first step, we performed a ROC analysis for the 82 individuals considered to have TA dilatation according to current guidelines (i.e.: > 40 mm for at least one aortic segment). The AUC of the OC-SVM global model including sex, age and BSA (0.995, sensitivity 97.56, specificity 95.51) was significantly higher than AUC obtained of the linear regression model, including the same variables (0.958, sensitivity 85.36, specificity 9.75; $p = 0.0012$).

5.3.3 Performance of linear regression and OC-SVM models in deemed-at-risk individuals

We separately run the two different models in the cohort of individuals deemed at risk of having aortic dilatation due either to the presence of MFS or BAV. Based on the linear regression analysis, the prevalence of TA dilatation (as Z-score > 2 in at least one segment), was 293/404 (72.5%) while it was 227/404 (56.2%) as O-score > 0 in at least one aortic region ($p < 0.00001$) (see Table 5.4). Seventy-five patients had Z-score > 2 in at least one TA region but O-score < 0 in all segments (18.6%), while O-score was > 0 in at least one TA level in 9 patients (2.2%) whose Z-score was < 2 in all aortic segments for a total of 84/404 (20.8%) discordant results. In Table 5.5, demographic and clinical characteristics of 84 discordant subjects were reported.

Table 5.5: Demographic and clinical characteristics of 84/404 discordant deemed-at-risk subjects.

	Patients with Z-score >2, but O-score <0 <i>n</i> = 75	Patients with Z-score <2, but O-score >0 <i>n</i> = 9
Age, years	17.33 (12.92 – 41.33)	5.5 (5 – 33)
Sex Female, n (%)	17 (22.67 %)	3 (33.33 %)
Body surface area, m²	1.62 (1.32 – 1.81)	1.38 (0.68 – 2.10)
MFS, n (%)	62 (82.67 %)	3 (33.33 %)
BAV, n (%)	13 (17.33 %)	6 (66.67 %)
Diameter indexed for BSA		
Aortic annulus,	13.92 (12.59 – 15.58)	13.40 (11.27 – 16.41)
Sinuses of Valsalva	20.41 (18.16 – 22.08)	18.47 (17.38 – 23.79)
Sinotubular junction	16.47 (14.52 – 18.33)	15.97 (15.21 – 17.61)
Proximal ascending aorta	18.59 (17.38 – 21.42)	18.32 (16.66 – 22.62)
Maximal Z-score	3.8	1.95
Maximal O-score	-0.01	0.56

5.4 Discussion

A number of previous studies explored determinants of normal aortic size measured by TTE, describing reference ranges in healthy individuals. Beyond echocardiographic methodological discrepancies between different studies, available nomograms and algorithms provided to predict normal size of the TA are of limited usability in a busy clinical echocardiographic laboratory, due to the need of implementing multiple algorithms for different age-groups and gender, also provided that not all of them assess normalcy at each and every aortic levels. Moreover, in order to be usable in a practical general context, a normalcy calculator should ideally fulfill the following desiderata. First, it should assess the aortic size at several levels (at least sinuses of Valsalva, sinotubular junction, and proximal ascending aorta). Second, it should predict normalcy on a single patient basis taking into account the influence of demographics and anthropometrics on TA size. Third, it should also be uniformly applicable to the whole range of patients, thus avoiding the introduction of age or body-size groups that make decisions intricate for subjects situated near the range extrema.

We have thus developed a general and comprehensive tool for normalcy of each of the 4 levels of TA, based on a large cohort of healthy individuals with a wide age range, free from cardiovascular diseases and, as a novelty, also free from family history either of TA aneurysm/dissection, or MFS or BAV.

Our tool offers two alternative normalcy assessment methods, one based on the traditional linear regression model and one based on a novel strategy that employs the machine learning algorithm OC-SVM. Both models were trained the same data derived from demographic and anthropometric characteristics of our study group of 1024 normal individuals. Each method provide a distinctively different score, namely a Z-score and an O-score, allowing prediction of normalcy for each aortic level. Additionally, the OC-SVM is able to provide a global O-score, taking into account the morphology of the whole aorta. As a further novelty, we also verified the effectiveness of the two methods on a cohort of individuals classified as abnormal based on current guidelines, and on a cohort of individuals deemed at risk of having aortic dilatation.

Noteworthy, both scores had an excellent performance in detecting abnormality as defined by guidelines. Nonetheless, estimated prevalence of TA dilatation in deemed-at-risk individuals was different between the 2 methods, with peculiar discordant patterns (Table 5.1). In particular, O-score was abnormal despite normal Z-score in about 2% of deemed-at-risk patients, mostly pediatric subjects more frequently affected by BAV. On the other hand, Z-score was abnormal differently from O-score in 18% of deemed-at-risk patients, most frequently affected by MFS, with a higher age and wider age span. Thus, this observational comparison shows, for the first time, that the 2 different approaches used to assess normalcy result into different prevalence of abnormality although based on the same anthropometric and demographic variables, supporting the potential role of the novelty detection techniques over the conventional approach to predict normalcy. In particular, the OC-SVM approach offers several advantages:

- it does not rely on any assumption on the density of interest and it is therefore not affected by problems such as heteroscedasticity and residues that are not normally distributed;
- it can exploit a kernel function to measure the similarity between a new test subject and the subjects in the study population; kernel functions implicitly map a realization x to a point in an infinite dimensional feature space, thus allowing to model complex nonlinear relationships among variables;
- as mentioned above, the global O-score can take into account the interplay among aortic diameters (thus incorporating global morphological

information of the whole aorta), by including them in x together with all other demographic and anthropomorphic variables.

In conclusion, we provide a new tool to assess TA normalcy and detect aortic dilatation based on a novel strategy that employs the machine learning algorithm OC-SVM (i.e.: O-score). When compared with an approach based on the traditional linear regression model trained on the same data set (i.e.: Z-score), the OC-SVM was slightly although significantly more effective in detecting abnormality in individuals with guidelines-defined aortic dilatation, and provided a distinctively smaller prevalence of abnormal aortic size in patients at risk of TA dilatation. Future studies should explore the potential role of novelty detection techniques in the reappraisal of prevalence of TA dilatation and to assess the natural history of O-score in the general population as well as in particular groups of patients at risk of aortic events.

Chapter 6

Conclusions

This chapter summarizes the contribution of this thesis and discusses avenues for future research.

6.1 Summary of contribution

This thesis makes two contributions to relational learning by proposing a novel method to apply neural networks to structured data and a possible application to a natural language processing task. A third contribution involves an application of novelty detection in cardiology.

- In Chapter 3 we introduced a framework to approach supervised learning problems on structured data by extending the \mathcal{R} -convolution concept used in most graph kernels [38]. We described how to represent a relational structure as a hierarchy of objects and we defined how to unroll a template neural network on it. This approach has shown to be able to outperform state-of-the-art methods on large social networks datasets, while at the same time being competitive on small chemobiological datasets. We also introduced a lossless compression algorithm for the hierarchical decompositions that improves the temporal and spatial complexities of our approach by exploiting symmetries in the input data.
- In Chapter 4 we described an application of the aforementioned method to context-dependent claim detection task. Claim detection is the assessment of whether a sentence contains a claim, i.e. the thesis, or con-

clusion, of an argument; in particular we focused on context-dependent claims, where the context (i.e. the topic of the argument) is a determining factor in classifying a sentence. We showed how Shift Aggregate Extract Networks are able to take advantage of contextual information in a straightforward way and we presented some preliminary results that indicates how this approach is viable on real world datasets.

- In Chapter 5 we described a machine learning approach to aortic size normalcy assesment. The definition of normalcy is crucial when dealing with thoracic aortas, as a dilatation of its diameter often precedes serious disease. A number of methods have been developed to obtain the expected aortic diameter starting from demographic and anthropometric characteristics such as height, weight, age and sex. Unfortunately, these methods suffer from several limitations: first, they are suitable either only for infants, children, and young adults, or only for adults; and second, they make strong assumptions on the relationship between predictors and aortic diameters. We overcame these limitations by building a new estimator based on OC-SVM [77] fitted on a cohort of 1024 healty individuals aging from 5 to 89 years, and we compared its results to those obtained on the same set of subjects by an approach based on linear regression. We also built a second estimator that combines the diameters measured at multiple levels in order to assess the normalcy of the overall shape of the aorta.

6.2 Directions for future work

As new relational data is produced every day, relational learning tasks are set to become more and more important. Labeling large datasets to train supervised learning algorithms, however, is an operation that in many cases must still be performed manually. A possible direction for future work would be to extend the architecture proposed in Chapter 3 for semi-supervised learning problems, where we can obtain large number of examples, only a small percentage of which are labeled. More specifically, it would be feasible to implement an unsupervised learning method using stacked autoencoders [90] for the extract step, leaving the shift-aggregate steps intact. While it is obvious how to reconstruct representations inside a single hierarchical level, however, it would be challenging to develop a suitable pseudo-inverse for shift-aggregation steps.

A second direction for future work would be to carry out a theoretical study on the dynamics of learning in the neural networks induced by SAEN. While training our architecture on large social graphs, we did notice that the complexity of these networks led to some instability where a “bad” weights initialization could lead to suboptimal results. The problem of choosing a good starting point when learning by gradient descent algorithms has been extensively researched [30, 39, 46, 74], however the existing results cannot be applied directly to our method due to the differences between our architecture and standard (deep) feedforward networks. A theoretical study would both help the choice of good initial weights (and therefore improve the stability of our method), and improve our understanding of how representations are built by the learning algorithm.

Appendix A

Publications

International Journals

Submitted

1. F. Orsini, **D. Baracchi**, P. Frasconi. “Shift Aggregate Extract Networks”, *Frontiers in Robotics and AI*, 2017.

To be submitted

1. P. Frasconi, **D. Baracchi**, B. Giusti, A. Kura, G. Spazian, S. Favilli, A. Cherubini, A. Di Lenarda, G. Pepe, R. Bonow, S. Nistri. “Two-dimensional echocardiographic aortic size in 1024 healthy individuals aged 5 to 89 years. Development of a machine learning approach to assess normalcy”, to be submitted to *Journal of American College of Cardiology: Cardiovascular Imaging*.

Manuscript in preparation

1. **D. Baracchi**, P. Frasconi, M. Lippi, F. Orsini, P. Torroni. “SAEN for Argumentation Mining”.

Bibliography

- [1] Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *ArgMining@ ACL*, pages 64–68, 2014.
- [2] Fabio Aioli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti. Fast on-line kernel learning for trees. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 787–791. IEEE, 2006.
- [3] P Baldi and G Pollastri. The principled design of large-scale recursive neural network architectures—dag-rnns and the protein structure prediction problem. *J Mach Learn Res*, 4(Sep):575–602, 2003.
- [4] Daniele Baracchi, Paolo Frasconi, Marco Lippi, Francesco Orsini, and Paolo Torrioni. Saen for argumentation mining. Manuscript in preparation.
- [5] Anna Maria Bianucci, Alessio Micheli, Alessandro Sperduti, and Antonina Starita. Application of cascade correlation networks for structures to chemistry. *Applied Intelligence*, 12(1-2):117–147, 2000.
- [6] Yonatan Bilu and Noam Slonim. Claim synthesis via predicate recycling. In *Proceedings of The 54th Annual meeting of the Association for Computational Linguistics*, 2016.
- [7] RO Bonow, B Carabello, AC De Leon, LH Edmunds, BJ Fedderly, MD Freed, WH Gaasch, CR McKay, RA Nishimura, PT O’Gara, et al. Acc/aha guidelines for the management of patients with valvular heart disease. executive summary. a report of the american college of cardiology/american heart association task force on practice guidelines (committee on management of patients with valvular heart disease). *The Journal of heart valve disease*, 7(6): 672–707, 1998.

-
- [8] K M Borgwardt and H-P Kriegel. Shortest-path kernels on graphs. In *Proc. of the ICDM-05*, pages 8–pp. IEEE, 2005.
- [9] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):i47–i56, 2005.
- [10] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [11] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.
- [12] Amparo Elizabeth Cano-Basave and Yulan He. A study of the impact of persuasive argumentation in political debates. In *Proceedings of NAACL-HLT*, pages 1405–1413, 2016.
- [13] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in neural information processing systems*, pages 625–632, 2002.
- [14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [15] F Costa and K De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proc. of the ICML-10*, pages 255–262. Omnipress, 2010.
- [16] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International Conference on Machine Learning*, pages 2702–2711, 2016.
- [17] Frederic Dallaire, Jean-Luc Bigras, Milan Prsa, and Nagib Dahdah. Bias related to body mass index in pediatric echocardiographic z scores. *Pediatric cardiology*, 36(3):667–676, 2015.
- [18] Ryan R Davies, Lee J Goldstein, Michael A Coady, Shawn L Tittle, John A Rizzo, Gary S Kopf, and John A Elefteriades. Yearly rupture or dissection rates for thoracic aortic aneurysms: simple prediction based on size. *The Annals of thoracic surgery*, 73(1):17–28, 2002.
- [19] Ryan R Davies, Ryan K Kaple, Divakar Mandapati, Amy Gallo, Donald M Botta, John A Elefteriades, and Michael A Coady. Natural history of ascending aortic aneurysms in the setting of an unreplaced bicuspid aortic valve. *The Annals of Thoracic Surgery*, 83(4):1338–1344, 2007.

- [20] Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. What is the essence of a claim? cross-domain claim identification. *arXiv preprint arXiv:1704.07203*, 2017.
- [21] Asim Kumar Debnath, de Compadre RL Lopez, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.*, 34(2):786–797, 1991.
- [22] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.
- [23] Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845, 1988.
- [24] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.
- [25] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. Neural end-to-end learning for computational argumentation mining. *arXiv preprint arXiv:1704.06104*, 2017.
- [26] Lorenzo Ferrone and Fabio Massimo Zanzotto. Towards syntax-aware compositional distributional semantic models. In *COLING*, pages 721–730, 2014.
- [27] Paolo Frasconi, Daniele Baracchi, Betti Giusti, Ada Kura, Gaia Spazian, Silvia Favilli, Antonella Cherubini, Di Lenarda Andrea, Guglielmina Pepe, Robert Bonow, and Stefano Nistri. Two-dimensional echocardiographic aortic size in 1024 healthy individuals aged 5 to 89 years. development of a machine learning approach to assess normalcy. To be submitted to *Journal of American College of Cardiology: Cardiovascular Imaging*, 2017.
- [28] Andrew B Gardner, Abba M Krieger, George Vachtsevanos, and Brian Litt. One-class novelty detection for seizure analysis from intracranial eeg. *Journal of Machine Learning Research*, 7(Jun):1025–1044, 2006.
- [29] Mathieu Gautier, Delphine Detaint, Christophe Fermanian, Philippe Aegerter, Gabriel Delorme, Florence Arnoult, Olivier Milleron, François Raoux, Chantal Stheneur, Catherine Boileau, et al. Nomograms for aortic root diameters in children using two-dimensional echocardiography. *The American journal of cardiology*, 105(6):888–894, 2010.

- [30] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.
- [31] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
- [32] Steven A Goldstein, Arturo Evangelista, Suhny Abbara, Andrew Arai, Federico M Asch, Luigi P Badano, Michael A Bolen, Heidi M Connolly, Hug Cuéllar-Calàbria, Martin Czerny, et al. Multimodality imaging of diseases of the thoracic aorta in adults: from the american society of echocardiography and the european association of cardiovascular imaging: endorsed by the society of cardiovascular computed tomography and society for cardiovascular magnetic resonance. *Journal of the American Society of Echocardiography*, 28(2):119–182, 2015.
- [33] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE, 1996.
- [34] Chinnappa Guggilla, Tristan Miller, and Iryna Gurevych. CNN- and lstm-based claim classification in online user comments. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2740–2751. ACL, 2016.
- [35] Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 2016.
- [36] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.
- [37] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.
- [38] D Haussler. Convolution kernels on discrete structures. Technical report, Citeseer, 1999.
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In

- Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [40] Guillaume Jondeau, Pierre Boutouyrie, Patrick Lacolley, Brigitte Laloux, Olivier Dubourg, Jean-Pierre Bourdarias, and Stéphane Laurent. Central pulse pressure is a major determinant of ascending aorta dilation in marfan syndrome. *Circulation*, 99(20):2677–2681, 1999.
- [41] William Karush. Minima of functions of several variables with inequalities as side conditions. *Master thesis, University of Chicago*, 1939.
- [42] H Kashima, K Tsuda, and A Inokuchi. Marginalized kernels between labeled graphs. In *n Proc. of the ICML-03*, volume 3, pages 321–328, 2003.
- [43] Seo Young Kim, Neil Martin, Elizabeth C Hsia, Reed E Pyeritz, and Daniel A Albert. Management of aortic disease in marfan syndrome: a decision analysis. *Archives of Internal Medicine*, 165(7):749–755, 2005.
- [44] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [45] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492. University of California Press, Berkeley and Los Angeles, 1951.
- [46] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.
- [47] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. Context dependent claim detection. In Jan Hajic and Junichi Tsujii, editors, *COLING 2014, Dublin, Ireland*, pages 1489–1500. ACL, 2014.
- [48] Marco Lippi and Paolo Torroni. Context-independent claim detection for argument mining. In *IJCAI*, volume 15, pages 185–191, 2015.
- [49] Marco Lippi and Paolo Torroni. Argument mining from speech: Detecting claims in political debates. In *AAAI*, pages 2979–2985, 2016.
- [50] Marco Lippi and Paolo Torroni. Margot: A web server for argumentation mining. *Expert Systems with Applications*, 65:292–303, 2016.
- [51] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2): 10, 2016.

- [52] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.
- [53] Wadi Mawad, Christian Drolet, Nagib Dahdah, and Frederic Dallaire. A review and critique of the statistical methods used to generate reference values in pediatric echocardiography. *Journal of the American Society of Echocardiography*, 26(1):29–37, 2013.
- [54] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.
- [55] M Mladenov, B Ahmadi, and K Kersting. Lifted linear programming. In *AISTATS-12*, pages 788–797, 2012.
- [56] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011. doi: 10.1007/s10506-010-9104-x. URL <http://dx.doi.org/10.1007/s10506-010-9104-x>.
- [57] Huy V Nguyen and Diane J Litman. Extracting argument and domain words for identifying argument components in texts. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 22–28, 2015.
- [58] Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument mining with structured svms and rnns. In *The 55th Annual Meeting of the Association for Computational Linguistics*, 2017.
- [59] M Niepert, M Ahmed, and K Kutzkov. Learning convolutional neural networks for graphs. *arXiv preprint arXiv:1605.05273*, 2016.
- [60] Nathan Ong, Diane Litman, and Alexandra Brusilovsky. Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28, 2014.
- [61] Francesco Orsini, Paolo Frasconi, and Luc De Raedt. Graph invariant kernels. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI’15)*, 2015.
- [62] Francesco Orsini, Daniele Baracchi, and Paolo Frasconi. Shift aggregate extract networks. Submitted to *Frontiers in Robotics and AI*, 2017.
- [63] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM, 2009.

- [64] Andreas Peldszus and Manfred Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *EMNLP*, volume 2015, pages 938–948, 2015.
- [65] Isaac Persing and Vincent Ng. End-to-end argumentation mining in student essays. In *Proceedings of NAACL-HLT*, pages 1384–1394, 2016.
- [66] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.
- [67] E Raimund, V Aboyans, C Boileau, E Bossone, R Di Bartolomeo, et al. Esc guidelines on the diagnosis and treatment of aortic diseases. *Eur Heart J*, 35(41):2873–2926, 2014.
- [68] Liva Ralaivola, Sanjay J. Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005. 00251.
- [69] J Ramon and T Gärtner. Expressivity versus efficiency of graph kernels. In *First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74. Citeseer, 2003.
- [70] Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. Show me your evidence-an automatic method for context dependent evidence detection. In *EMNLP*, pages 440–450, 2015.
- [71] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12:77, 2011.
- [72] Haggai Roitman, Shay Hummel, Ella Rabinovich, Benjamin Sznajder, Noam Slonim, and Ehud Aharoni. On the retrieval of wikipedia articles containing claims on controversial topics. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 991–996. International World Wide Web Conferences Steering Committee, 2016.
- [73] Mary J Roman, Richard B Devereux, Randi Kramer-Fox, and John O’Loughlin. Two-dimensional echocardiographic aortic root dimensions in normal children and adults. *The American journal of cardiology*, 64(8):507–512, 1989.
- [74] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

- [75] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *Neural Networks, IEEE Transactions on*, 20(1):61–80, 2009. 00073.
- [76] Bernhard Schoelkopf, Jason Weston, Eleazar Eskin, Christina Leslie, and William Stafford Noble. A kernel approach for learning from almost orthogonal patterns. In *European Conference on Machine Learning*, pages 511–528. Springer, 2002.
- [77] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- [78] N Shervashidze, P Schweitzer, E J van Leeuwen, K Mehlhorn, and K M Borgwardt. Weisfeiler-lehman graph kernels. *J Mach Learn Res*, 12(Sep):2539–2561, 2011.
- [79] R Socher, C C Lin, C Manning, and A Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proc. of the ICML-11*, pages 129–136, 2011.
- [80] Helge Erik Solberg and Ari Lahti. Detection of outliers in reference distributions: performance of horn’s algorithm. *Clinical chemistry*, 51(12):2326–2332, 2005.
- [81] A Sperduti and A Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.
- [82] Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *EMNLP*, pages 46–56, 2014.
- [83] Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 2017.
- [84] L Tarassenko, A Hann, and D Young. Integrated monitoring and analysis for early warning of patient deterioration. *BJA: British Journal of Anaesthesia*, 97(1):64–68, 2006.
- [85] Simone Teufel. Argumentative zoning. *PhD Thesis, University of Edinburgh*, 1999.
- [86] Alessandro Tibo, Paolo Frasconi, and Manfred Jaeger. A network architecture for multi-multi-instance learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017*, Skopje, 2017. Springer.

- [87] Hannu Toivonen, Ashwin Srinivasan, Ross D King, Stefan Kramer, and Christoph Helma. Statistical evaluation of the predictive toxicology challenge 2000–2001. *Bioinformatics*, 19(10):1183–1193, 2003.
- [88] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484, 2005.
- [89] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.
- [90] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.
- [91] A Vullo and P Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20(5):653–659, 2004.
- [92] Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.
- [93] Douglas Walton. Argumentation theory: A very short introduction. In Guillermo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 1–22. Springer US, 2009. ISBN 978-0-387-98196-3. doi: 10.1007/978-0-387-98197-0_1.
- [94] Zhongyu Wei, Chen Li, and Yang Liu. A joint framework for argumentative text analysis incorporating domain knowledge. *arXiv preprint arXiv:1701.05343*, 2017.
- [95] P Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proc. of KDD-15*, pages 1365–1374, 2015.
- [96] Fabio Massimo Zanzotto and Lorenzo Dell’Arciprete. Distributed tree kernels. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, pages 115–122. Omnipress, 2012.