Ghent University
Faculty of Sciences
Department of Applied Mathematics, Computer Science and Statistics

Similarity relations for multi-instance data

Ruben Vijverman

GHENT UNIVERSITY

Academic year 2021–2022

Supervisor: prof. Chris Cornelis

Master's dissertation submitted in order to obtain the academic degree of master of mathematics

Acknowledgements

This master's thesis was made possible thanks to the support and cooperation of many people. Therefore, I would like to sincerely thank everyone who offered me support and advice.

First of all, I would like to thank my supervisor Prof. Chris Cornelis for providing me with the subject matter and the excellent support during the writing of this thesis. Thanks to his expert advice and critical remarks, I learned a lot about this subject. He was always willing to help and answer my questions. I am also very grateful to him for repeatedly proofreading my master's thesis.

Secondly, I would like to thank Febe Vagenende for the mental and emotional support. In addition, I would like to thank her for proofreading and spellchecking my thesis.

The people attending my presentation during one of the CWI seminars also deserve a word of thanks for the advice and questions I received from them regarding my master dissertation.

Finally, I thank my parents, my sister and friends for the unconditional support and interest they showed in this master's thesis.

Thank you!

Permission of use on loan

The author gives permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In all cases of other use, the copyright terms have to be respected, in particular with regard to the obligation to state explicitly the source when quoting results from this master dissertation.

Signed on 31 May 2022 Ruben Vijverman

Contents

1 Introduction

This master's thesis is mostly about using similarity relations for multi-instance classification. Multi-instance classification, like any other classification problem, tries to classify objects to its correct class. However, what makes multi-instance classification unique is that these objects are a collection of instances. These collections of instances will be called bags. Normally speaking classification problems only have to deal with a single instance at a time, and therefore there are several approaches that try to solve the multi-instance classification problem by predicting the labels of the instances separately in order to then aggregate these predictions to a single prediction for the whole bag. This can be challenging because the training data often only gives the labels on the bag-level which means making accurate models to classify single instances can become virtually impossible. Extrapolating the bag label to the instances may, after all, lead to a lot of miss-classified training samples. The aim of this thesis is to circumvent these problems by directly labelling the bags. However, this approach leads to other challenging problems, many of which will be discussed in further chapters.

More specifically, similarity between bags will be studied using concepts from fuzzy set theory and graph theory. This thesis aims to provide some theoretical results regarding these similarity relations between bags, as well as some experimental results. The theoretical discussion will mainly regard distances between bags and the connection between distance and similarity between bags. The experimental results come from putting several of these similarity relations to the test in a fuzzy multi-instance classification algorithm. I have chosen this topic for several reasons. First of all, I have a keen interest in machine learning and graph algorithms. Both of which are used to some extent in this master's thesis.

Secondly, this topic offered a good balance between theoretical and practical work. This is also largely how this master's thesis is structured. First, Chapter 2 introduces some basic concepts that will be used throughout this master's thesis. In Chapter 3 the theoretical properties of similarity relations and distances between bags are discussed. We take a look at some distances between bags that are common in the literature and some bag distances that are new to the context of multi-instance classification. Secondly, we discuss the conversion of bag distance to bag similarity. This conversion can lead to some interesting additional properties of the similarity relation which are also studied. Lastly, this chapter handles some theoretical properties of the newly introduced bag distances using concepts from graph theory. These theoretical results are vital for the implementation of these bag distances in any practical application, because they significantly decrease the computational complexity of these distances.

Chapter 4 introduces the concept of multi-instance learning and aims to give an overview of the approaches taken for solving the multi-instance classification problem in the literature. Furthermore, we introduce the BFMIC family of multi-instance classifiers that will be used in the experimental study.

In Chapter 5 we conduct the experimental study and compare the performance of different similarity relations on both balanced datasets and imbalanced datasets. We also consider the use of some pre-processing steps and study their effect on the performance of different similarity relations.

Finally, in Chapter 6 we summarize the master's thesis and the obtained results. Furthermore, we provide some potential possibilities for further research.

2 Preliminaries

2.1 Fuzzy set theory

Fuzzy set theory is an extension to classical set theory where objects can belong to a set to a given degree. In classical set theory objects either belong to the set or they do not. In fuzzy set theory we model these fuzzy sets as an $X \to [0, 1]$ mapping that signifies to which degree the object belongs to the fuzzy set.

Analogous to classical set theory we can define a fuzzy relation R in X as a fuzzy set in $X \times X$. If the fuzzy relation R is reflexive and symmetric, the relation R is called a fuzzy tolerance relation. That is:

$$\begin{aligned} R(x, x) &= 1 \quad \forall x \in X \\ R(x, y) &= R(y, x) \quad \forall x, y \in X \end{aligned}$$

These fuzzy tolerance relations can be used to express how similar two objects are to each other.

The classical logical connectives can also be extended to fuzzy logical connectives. The Boolean conjunction is extended to a conjunctor C, which is a mapping $C : [0, 1] \times [0, 1] \to [0, 1]$ that is increasing in both arguments and satisfies the following border conditions:

- $C(0, 0) = 0$
- $C(1, 0) = 0$
- $C(0, 1) = 0$
- $C(1, 1) = 1$

A conjunctor C for which $\forall x, y, z \in [0, 1]$

- $C(1, x) = x$
- $C(x, y) = C(y, x)$
- $C(C(x, y), z) = C(x, C(y, z))$

is called a t-norm. Some well-known examples of t-norms are the Łukasiewicz t-norm $T_L(x, y) = \max(0, x + y - 1)$, the product t-norm $T_p(x, y) = xy$, and the minimum $T_{\min}(x, y) = \min(x, y)$.

2.2 Ordered weighted average aggregation

Ordered weighted average (OWA) aggregation [1] is a versatile way to aggregate a vector of values to a single value. The aggregation follows the following procedure: first, the values of the vector are sorted in decreasing order. Secondly, the weighted sum is calculated using a weight vector W by assigning each weight of W according to the ordered position.

Definition 2.2.1. The OWA-aggregation of a vector $V = \langle v_1, v_2, \ldots, v_n \rangle$ is given by:

$$\mathrm{OW\,A_W}(V) = \sum_{i=1}^{n} w_i v_{\rho(i)} = W \cdot \tilde{V},$$

where $\tilde{V} = \langle v_{\rho(1)}, v_{\rho(2)}, \ldots, v_{\rho(n)} \rangle$ with $v_{\rho(i)}$ being the i-largest element of V and weight vector $W = \langle w_1, w_2, \ldots, w_n \rangle$ with $\forall i : w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$.

Many common aggregation strategies can be modelled using these weight vectors. Take for instance:

1. $W^{max} = \langle 1, 0, \ldots, 0 \rangle$
2. $W^{min} = \langle 0, \ldots, 0, 1 \rangle$
3. $W^{avg} = \langle w_1, w_2, \ldots, w_n \rangle$ with $\forall i : w_i = \frac{1}{n}$

These weight vectors can be used to respectively determine the maximum, minimum and average value of a vector V. It should however be clear that there are many more possible OWA-operators. A property that characterizes an OWA-operator is its *orness*-value. This value represents how close the aggregation is to a regular maximum. The orness of an OWA-operator $W = \langle w_1, w_2, \ldots, w_n \rangle$ is calculated as:

$$orness(W) = \frac{1}{n-1} \sum_{i=1}^{n} (n - i) w_i$$

It can be easily verified that $orness(W_{max}) = 1$, $orness(W_{min}) = 0$, $orness(W_{avg}) = \frac{1}{2}$

It is common to also define the *andness* of an OWA-operator. This value expresses how close the aggregation is to a normal minimum and is defined as $1 - orness(W)$. If $orness(W) > \frac{1}{2}$ the OWA-aggregation using W as weight vector corresponds to a softened maximum. Similarly $andness(W) > \frac{1}{2}$ indicates a softened minimum. We will now see what the effect of reversing such a weight vector is on the $orness$ and $andness$.

Lemma 2.2.2. *A weight vector* $W = \langle w_1, w_2, \ldots, w_n \rangle$ *and its reverse* $W' = \langle w'_1, w'_2, \ldots, w'_n \rangle = \langle w_n, w_{n-1}, \ldots, w_1 \rangle$ *satisfy* $orness(W) + orness(W') = 1$

Proof. We can write $orness(W')$ as

$$\begin{aligned} orness(W') &= \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w'_i \\ &= \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_{n+1-i} \end{aligned}$$

If we now carry out the substitution $j = n + 1 - i$, we find

$$orness(W') = \frac{1}{n-1} \sum_{j=1}^{n} (j-1) w_j$$

This indicates that

$$\begin{aligned} orness(W') + orness(W) &= \frac{1}{n-1} \sum_{j=1}^{n} (j-1) w_j + \frac{1}{n-1} \sum_{j=1}^{n} (n-j) w_j \\ &= \frac{1}{n-1} \sum_{j=1}^{n} (n-j+j-1) w_j \\ &= \frac{n-1}{n-1} \sum_{j=1}^{n} w_j \\ &= \sum_{j=1}^{n} w_j = 1. \end{aligned}$$

□

Lemma 2.2.2 provides a natural way to couple OWA-operators. If $orness(W) > \frac{1}{2}$ then we know that the orness of its reverse weight vector W' will be smaller than $\frac{1}{2}$ and vice versa. This allows each OWA-operator to define a substitute for the normal maximum and minimum. From now on we will write a weight vector that has an orness greater than or equal to $\frac{1}{2}$ as W_u and its reverse as W_l. This allows us to define:

$$\begin{aligned} W_u^{strict} &= \langle 1, 0, \ldots, 0 \rangle = W^{max} \\ W_l^{strict} &= \langle 0, \ldots, 0, 1 \rangle = W^{min} \end{aligned}$$

and moreover allows us to only define one of the two in a pair. The most common pairs of OWA-operators are:

- **Strict weights (strict)**:

$$\begin{aligned} W_u^{strict} &= \left\langle 1, 0, \ldots, 0 \right\rangle \\ W_l^{strict} &= \left\langle 0, \ldots, 0, 1 \right\rangle \end{aligned}$$

- **Additive weights (add)**:

$$W_u^{add} = \left\langle \frac{2}{n+1}, \frac{2(n-1)}{n(n+1)}, \ldots, \frac{4}{n(n+1)}, \frac{2}{n(n+1)} \right\rangle$$

$$W_l^{add} = \left\langle \frac{2}{n(n+1)}, \frac{4}{n(n+1)}, \ldots, \frac{2(n-1)}{n(n+1)}, \frac{2}{n+1} \right\rangle$$

- **Exponential weights (exp)**:

$$W_u^{exp} = \left\langle \frac{2^{n-1}}{2^n-1}, \frac{2^{n-2}}{2^n-1}, \ldots, \frac{2}{2^n-1}, \frac{1}{2^n-1} \right\rangle$$

$$W_l^{exp} = \left\langle \frac{1}{2^n-1}, \frac{2}{2^n-1}, \ldots, \frac{2^{n-2}}{2^n-1}, \frac{2^{n-1}}{2^n-1} \right\rangle$$

- **Inverse additive weights (invadd)**:

$$W_u^{add} = \left\langle \frac{1}{D_p}, \frac{1}{2D_p}, \ldots, \frac{1}{(p-1)D_p}, \frac{1}{pD_p} \right\rangle$$

$$W_l^{add} = \left\langle \frac{1}{pD_p}, \frac{1}{(p-1)D_p}, \ldots, \frac{1}{2D_p}, \frac{1}{D_p} \right\rangle$$

  with $D_p = \sum_{i=1}^{n} \frac{1}{i}$

- **Average weights (avg)**:

$$W_u^{avg} = \left\langle \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}, \frac{1}{n} \right\rangle$$

$$W_l^{avg} = \left\langle \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}, \frac{1}{n} \right\rangle$$

2.3 Similarity measures for fuzzy sets

To express the degree to which two fuzzy sets overlap or resemble each other, we define the concept of a similarity measure. This concept plays a role in many practical applications, and will be of notable importance for the classification of multi-instance data.

Definition 2.3.1. Let $\mathcal{F}(X)$ be the class of all fuzzy sets on a universe X. A similarity measure (or similarity relation) is a mapping $S : \mathcal{F}(X) \times \mathcal{F}(X) \to [0, 1]$ that satisfies the following conditions:

$$(\forall A, B \in \mathcal{F}(X) \times \mathcal{F}(X)) \quad S(A, B) = S(B, A)$$

$$(\forall A, B \in \mathcal{F}(X) \times \mathcal{F}(X)) \quad A = B \implies S(A, B) = 1$$

Similarity measures can be categorized using several properties they might possess. Some common properties are summed up in the following definition.

Definition 2.3.2. Let S be a similarity measure and T a t-norm. We define:

- S is T-transitive if and only if

$$(\forall(A,B,C) \in \mathcal{F}(X) \times \mathcal{F}(X) \times \mathcal{F}(X))(T(S(A,B),S(B,C)) \leq S(A,C))$$

- S satisfies the separation property if and only if

$$(\forall(A,B) \in \mathcal{F}(X) \times \mathcal{F}(X))(S(A,B) = 1 \iff A = B)$$

- S is exclusive if and only if

$$(\forall(A,B) \in (\mathcal{F}(X) \times \mathcal{F}(X) \backslash (\varnothing,\varnothing)))(A \cap B = \varnothing \implies S(A,B) = 0)$$

- S is co-exclusive if and only if

$$(\forall(A,B) \in (\mathcal{F}(X) \times \mathcal{F}(X)))(S(A,B) = 0 \implies A \cap B = \varnothing)$$

2.4 Graph Theory

In the following section several concepts from graph theory, that will be used throughout the following chapters, will be introduced.

Definition 2.4.1. A (undirected) graph is an ordered pair $G = (V, E)$, where V is a set whose elements are called vertices and $E \subseteq \{\{x, y\} \mid x, y \in V\}$ is a set whose elements are called edges. Two vertices x and y are said to be adjacent when $\{x, y\} \in E$. Similarly two distinct edges are said to be adjacent when they share a common vertex as one of their endpoints. A graph is said to be complete whenever every two distinct vertices are adjacent.

Definition 2.4.2. A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E$.

Certain types of (sub)graphs are given a name to communicate the composition of the (sub)graph more easily.

Definition 2.4.3. A pair is a graph consisting of two adjacent vertices. A star is a connected graph where all but one vertices have degree 1. The vertex that does not have degree 1 is called the center and is connected to all the other vertices of the star.

Definition 2.4.4. A directed graph, or digraph for short, is an ordered pair $G = (V, A)$, where V is once again a set of vertices. However, $A \subseteq V \times V$ is now a set of ordered pairs, whose elements are now called directed edges.

Definition 2.4.5. A bipartite graph is a graph where the set of vertices is split up in two disjoint bipartition classes X and Y. We denote a bipartite graph with bipartition classes X, Y as $(X \sqcup Y, E)$. We call a bipartite graph $G = (X \sqcup Y, E)$ balanced if $|X| = |Y|$. A bipartite graph is called complete when

$$\forall x \in X, \forall y \in Y : (x, y) \in E.$$

Definition 2.4.6. A weighted graph is a graph were every edge is assigned a certain weight. We define these weights to be non-negative. Formally we can describe a weighted graph G as an ordered triple (V, E, ω), where V and E are once again sets of vertices, and edges respectively. ω is a mapping from E to $\mathbb{R}^+$ that assigns a non-negative weight to each edge.

2.4.1 Matching

Graphs are often used to model connections between objects. A heavily studied problem in graph theory is that of finding pairs of vertices that best fit together, like in Hall's famous marriage theorem [2]. These pairings lead to the concept of a matching.

Definition 2.4.7. Given a graph $G = (V, E)$, a matching M is a subset of edges from E such that no two edges are adjacent. A vertex is said to be saturated if it is an endpoint of one of the edges in the matching M. A matching is called perfect whenever every vertex is saturated. In Figure 2.1 several examples of matchings are shown.

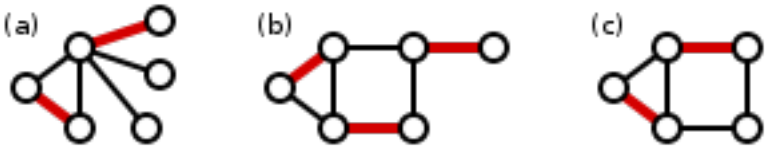

Figure 2.1: Three examples of matchings. Edges in the matching are colored red. Only example (b) is a perfect matching.

As stated before, we are often interested in a matching that is in a sense optimal. In our case, it will make sense to pair up vertices that are close to each other. We can define a complete weighted graph that has the distance between two vertices as the weight of the edge between these two vertices. Then we can define the following concept:

Definition 2.4.8. Given a weighted graph $G = (V, E, \omega)$, the weight of a matching M is defined as

$$w(M) := \sum_{e \in M} \omega(e).$$

A minimum weight perfect matching M is a perfect matching of minimal weight. In Figure 2.2 the weight of two matchings is shown, including a minimum weight perfect matching.

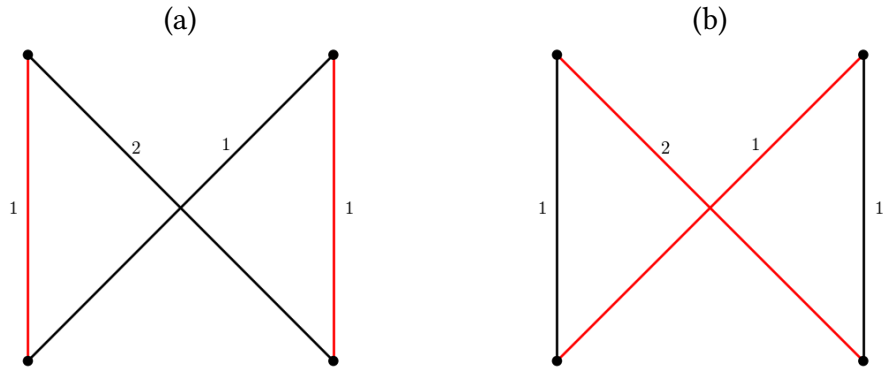

Figure 2.2: The red edges are the edges of the considered matching. Both matchings are perfect matchings because they both saturate all vertices. The matching shown in (a) is a minimum weight perfect matching.

Lemma 2.4.9. *A complete, balanced bipartite graph $G = (X \sqcup Y, E)$ has a perfect matching.*

Proof. Since G is balanced, and thus $|X| = |Y|$, there exists a bijection $\alpha : X \to Y$. Now we construct the matching M:

$$M = \{(x, \alpha(x)) \mid x \in X\}.$$

By construction and because α is a bijection every vertex of X is contained in a unique edge of M. Similarly we can see that every vertex of Y must be contained in a unique edge of M. Therefore M is a matching that saturates all vertices of $X \sqcup Y$, and thus a perfect matching. □

2.4.2 Flow Network

Another common application of graphs is models of networks through which a certain resource travels, e.g. road networks, electrical circuits, and data networks [3].

Definition 2.4.10. A network is a digraph $G = (V, E)$ together with a non-negative mapping $cap : E \to \mathbb{R}^+$, called the capacity function. A flow network (G, cap, s, t) is a network where s and t are vertices from G. s and t are respectively called the source and sink of the flow network.

The source and sink can be seen as the start and end point of the network respectively. The capacity of an edge signifies the maximal amount of a certain resource that can go through that specific edge. This is similar to the concept of bandwidth in data networks. The source provides the resource while the sink takes the resource out of the network. It makes sense to require that all vertices, with the exception of the source and the sink, have an equal amount of resources coming in as going out. In electrical circuits this is called Kirchhoff's first law. We often say that the resource flows through the network. To express how many resources a network can accommodate we introduce the following definition.

Definition 2.4.11. A flow on a network flow is a function $f : E \to R^+$ satisfying the following conditions

- For all edges $(u, v) \in E$: $f((u, v)) \leq c((u, v))$
- For every vertex $v \in V \setminus \{s, t\}$:

$$\sum_{u \in \{x|(x,v) \in E\}} f((u, v)) - \sum_{u \in \{x|(v,x) \in E\}} f((v, u)) = 0$$

- The source s has no incoming flow:

$$\forall u \in \{x \mid (x, s) \in E\} : f((u, s)) = 0$$

- The sink t has no outgoing flow:

$$\forall v \in \{v \mid (t, v) \in E\} : f((t, v)) = 0$$

The total flow $f_{s \to t}$ on the flow network is defined as the total flow going from the source to the sink:

$$f_{s \to t} = \sum_{w \in \{x|(s,x) \in E\}} f((s, w)).$$

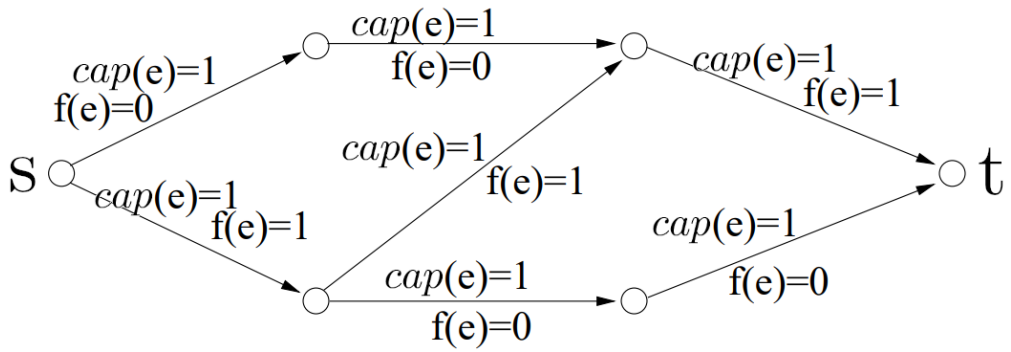

Figure 2.3: Example of a non-optimal flow defined on a flow network.

A common problem defined on flow networks is determining a flow that maximizes the total flow on the network [4]. In Figure 2.3 an example of a non-optimal flow f is given. Indeed, one can easily verify that the flow on this network is 1. However, by defining the flow of the edge going from the lower branch to the upper branch as 0 and defining the flow of all other edges as 1, a total flow of 2 can be achieved. A list of algorithms to solve the maximum-flow problem can be found in [5].

Definition 2.4.12. A weighted flow network is a flow network along with a non-negative mapping $c : E \to \mathbb{R}^+$ called the weight function. We denote the weighted flow network as (G, cap, c, s, t), where G is a digraph, cap is the capacity, c is the weight function and s and t are once again the source and the sink respectively.

Weighted flow networks come from a field where it often makes more sense to talk about a cost function instead of a weight function. A common problem is to then find a maximum flow f running through the network and minimizing the cost to achieve this maximal flow. The cost of running a flow is dependent on how much flow is running through an edge and the cost of the edge. This gives rise to the following definition:

Definition 2.4.13. Let f be a flow on a weighted flow network N with cost function c . We define the cost of the flow f as

$$c(f) = \sum_{e \in E} c(e)f(e)$$

The problem of finding the minimal-cost maximum-flow on a network has been studied in several papers [6][7].

3 Similarity relations between bags

In this chapter we will study similarity relations between bags. Section 3.1 will first focus on distance functions between bags. Here we also introduce some distances between bags which are novel in the context of multi-instance learning. This will then be followed up by Section 3.2, in which the relation between distance functions and similarity relations is explored. We conclude with Section 3.3, where some theoretical properties of the new bag distances are proven.

3.1 Metrics for bags

The concept of a bag has already been mentioned a lot. But we can formally define a bag as follows:

Definition 3.1.1. A bag is a multiset that contains elements of a metric space (M, d).

To determine the similarity between bags we can consider the distance between the bags. Typically distance is modelled using metrics.

Definition 3.1.2. A metric on a non-empty set X is a function $d : X \times X \to \mathbb{R}^+$ that satisfies $\forall x, y, z \in X$

$$\begin{array}{cc} d(x,y) \geq 0 & \text{(non-negative)} \\ d(x,y) = 0 \iff x = y & \text{(identity of indiscernibles)} \\ d(x,y) = d(y,x) & \text{(symmetry)} \\ d(x,y) + d(y,z) \leq d(x,z) & \text{(triangle inequality)} \end{array}$$

We call (X,d) a metric space. Functions that satisfy the first three conditions are called semi-metrics and functions that satisfy conditions 1,3 and 4 are called pseudo-metrics.

The relation between metrics and similarity will be discussed in Section 3.2. This connection implies that algorithms that use a certain metric can be transformed in an algorithm that uses similarity measures. However, we have not yet defined a way to determine the distance between two bags.

Single instances can be trivially embedded in a metric space (M, d) by using the feature vector as coordinates, the bags can thus be seen as a multiset of points in that space and therefore we must extend the metric d to a metric δ on all the non-empty (finite) subsets of M. This extension should be natural in a sense that $\forall x, y \in M : d(x, y) = \delta(\{x\}, \{y\})$ should be satisfied. The best-known metric between subsets of a metric space is the Hausdorff metric.

Definition 3.1.3. Let X, Y be two non-empty subsets of a metric space (M, d). We define the Hausdorff distance $\delta_H(X, Y)$ between X and Y as

$$\delta_H(X,Y) = \max\left\{ \sup_{x\in X} d(x,Y), \sup_{y\in Y} d(X,y) \right\}$$

with

$$d(x,Y) = \inf_{y\in Y} d(x,y)$$

Because the bags will always be finite, we can replace the use of the suprema and infima in the definition by the maximum and the minimum respectively.

Theorem 3.1.4. *Let (M, d) be a metric space, then the Hausdorff distance δ_H that extends d is a pseudo-metric.*

Proof. We will prove the 3 requirements of a pseudo-metric.

1. $\forall X, Y \in \mathcal{P}(M) : \delta_H(X, Y) \geq 0$
   Since $\forall x, y \in M : d(x, y) \geq 0$, we have that $\sup_{x\in X} d(x, Y) \geq 0$ and $\sup_{x\in X} d(X, y) \geq 0$. Therefore, we have that $\forall X, Y \in \mathcal{P}(M) : \delta_H(X, Y) \geq 0$.

2. $\forall X, Y \in \mathcal{P}(M) : \delta_h(X, Y) = \delta_h(Y, X)$
   This follows immediately from the symmetry of d.

3. $\forall X, Y, Z \in \mathcal{P}(M) : \delta_H(X, Y) + \delta_H(Y, Z) \geq \delta_H(X, Z)$
   First of all, the triangle inequality of d gives us:

   $$\forall x \in X, \forall y \in Y : \inf_{z\in Z} d(x,z) \leq \inf_{z\in Z}(d(x,y) + d(y,z)),$$

   and thus

   $$\forall x \in X, \forall y \in Y : d(x,Z) \leq d(x,y) + d(y,Z).$$

   Since

   $$d(y,Z) \leq \sup_{y\in Y} d(y,Z) \leq \delta_H(Y,Z)$$

   holds for every $y \in Y$, we find that

   $$d(x,Z) \leq d(x,y) + \delta_H(Y,Z),$$

   and thus

   $$d(x,Z) \leq \inf_{y\in Y}(d(x,y) + \delta_H(Y,Z))$$
   $$d(x,Z) \leq d(x,Y) + \delta_H(Y,Z).$$

   And once again since

   $$d(x,Y) \leq \sup_{x\in X} d(x,Y) \leq \delta_H(X,Y)$$

we find that

$$\begin{aligned} d(x, Z) &\leq \delta_H(X, Y) + \delta_H(Y, Z) \\ \sup_{x \in X} d(x, Z) &\leq \delta_H(X, Y) + \delta_H(Y, Z). \end{aligned}$$

Similarly, we can find that

$$\sup_{z \in Z} d(X, z) \leq \delta_H(X, Y) + \delta_H(Y, Z)$$

and thus

$$\delta_H(X, Z) = \max \left\{ \sup_{x \in X} d(x, Z), \sup_{z \in Z} d(X, z) \right\} \leq \delta_H(X, Y) + \delta_H(Y, Z)$$

□

Remark 3.1.5. The Hausdorff distance satisfies the second requirement of a metric ($\forall X, Y \in \mathcal{P}(M) : \delta_H(X, Y) = 0 \iff X = Y$) if all $X, Y \in \mathcal{P}(M)$ are finite.
Suppose that $X = Y$, then it is easy to verify that

$$\delta_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\} = 0$$

Now suppose that $\delta_H(X, Y) = 0$. Since X and Y are finite, we find that

$$\max_{x \in X} d(x, Y) = 0 \wedge \max_{y \in Y} d(y, X) = 0$$

Since $d(x, Y) = \min_{y \in Y} d(x, y)$, we find that

$$\begin{aligned} \max_{x \in X} d(x, Y) = 0 &\Rightarrow \forall x \in X : \min_{y \in Y} d(x, y) = 0 \\ &\Rightarrow \forall x \in X : \exists y \in Y : d(x, y) = 0 \\ &\Rightarrow \forall x \in X : \exists y \in Y : x = y \\ &\Rightarrow X \subseteq Y \end{aligned}$$

Analogously, we find that $Y \subseteq X$ and thus $X = Y$.

It should be noted that for the purpose of MIL the bags are finite. Therefore, the suprema and infima in the definition can be replaced with maxima and minima respectively. From now on we will assume that bags are finite. This way, there exist other metrics that are an extension of metrics on M to subsets of M, for example:

Definition 3.1.6. Let X, Y be two finite non-empty subsets of a metric space (M, d). We define the average Hausdorff distance $d_{avgH}(X, Y)$ between X and Y as

$$\delta_{avgH}(X, Y) = \frac{\sum_{x \in X} \min_{y \in Y} d(x, y) + \sum_{y \in Y} \min_{x \in X} d(x, y)}{|X| + |Y|}$$

This is sometimes also used without the normalisation for the number of instances in each bag, giving the so-called sum of minimal distances:

Definition 3.1.7. Let X, Y be two finite non-empty subsets of a metric space (M, d). We define the sum of minimal distances $d_{SumMin}(X, Y)$ between X and Y as

$$\delta_{H_{\text{SumMin}}}(X, Y) = \sum_{x \in X} \min_{y \in Y} d(x, y) + \sum_{y \in Y} \min_{x \in X} d(x, y)$$

Lastly, we can also consider the following easy to compute distance between bags.

Definition 3.1.8. Let X, Y be two finite non-empty subsets of a metric space (M, d). We define the minimal Hausdorff distance $d_{min}(X, Y)$ between X and Y as

$$\delta_{H_{\text{min}}}(X, Y) = \min_{y \in Y} \min_{x \in X} d(x, y)$$

Furthermore, we can use the OWA-operators by replacing the minima (respectively maxima) in the definitions of these distances by soft minima (respectively soft maxima). This means that for each pair of OWA-operators (OWA_{W_U}, OWA_{W_L}) we can define 4 additional distance measures between bags. For example the Hausdorff distance and average Hausdorff distance between X, Y (two finite, non-empty subsets of a metric space (M, d)) would respectively become:

$$\delta_{H,W}(X, Y) = \max \left\{ \begin{array}{l} OWA_{W_U}\big(\{OWA_{W_L}(\{d(x, y)|y \in Y\})|x \in X\}\big), \\ OWA_{W_U}\big(\{OWA_{W_L}(\{d(x, y)|x \in X\})|y \in Y\}\big) \end{array} \right\}$$

$$\delta_{avgH,W}(X, Y) = \frac{\sum_{x \in X} OWA_{W_L}(\{d(x, y)|y \in Y\}) + \sum_{y \in Y} OWA_{W_l}(\{d(x, y)|x \in X\})}{|X| + |Y|}$$

However, there are more distance measures that are not related to the Hausdorff distance measures. We will now discuss three distance measures that rely on relations between the two (multi)sets. These bag distances originally come from the field of philosophy of science to measure the distance between theories in a logical language $\mathcal{L}$. Nevertheless, as suggested in [8], these distance measures can be useful in other fields. Note that these distances are only defined for finite bags.

Surjection distance (δ_s)
This distance measure considers surjections that map the larger multiset onto the smaller. This gives rise to the following distance measure between two non-empty subsets X, Y of a metric space (M, d):

$$\delta_s(X, Y) = \min_{\eta} \sum_{(x,y) \in \eta} d(x, y),$$

where the minimum is taken over all possible surjections η from the larger multiset to the smaller multiset. Note that when both multisets have the same cardinality, the surjections become bijections. In that case it doesn't matter which multiset is the domain and which

multiset is the image because when the inverse functions are considered the minimum will stay the same.

Fair surjection distance (δ_{fs})
Once again surjections that map the larger multiset onto the smaller are considered. The surjection distance can be susceptible to noise and outliers, so to mitigate this we consider 'fair' surjections. The points in the domain should be equally distributed to the pre-images of the surjection, i.e., $\forall x, y \in im(\eta) : ||\eta^{-1}(x)| - |\eta^{-1}(y)|| \leq 1$. We define the following distance measure between two non-empty subsets X, Y of a metric space (M, d):

$$\delta_{fs}(X, Y) = \min_{\eta'} \sum_{(x,y)\in\eta'} d(x, y),$$

where the minimum is taken over all possible fair surjections η' from the larger multiset to the smaller multiset. Once again it doesn't matter which multiset is considered the largest when the cardinalities of the multisets are the same.

The surjection distance and fair surjection distance are very similar. The difference between these two distances becomes clear when one bag has significantly more instances than the other bag, such as in Figure 3.1. The surjection distance and the fair surjection between the bags in this example are illustrated in Figure 3.2.

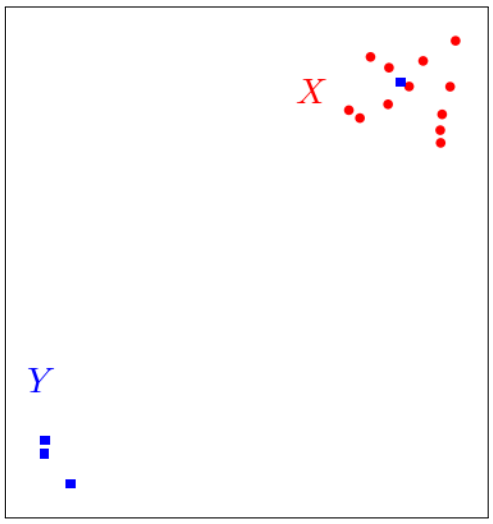

Figure 3.1: Two bags with different sizes. X is represented by the red dots, while Y is represented by the blue squares ($|X| = 12, |Y| = 4$).

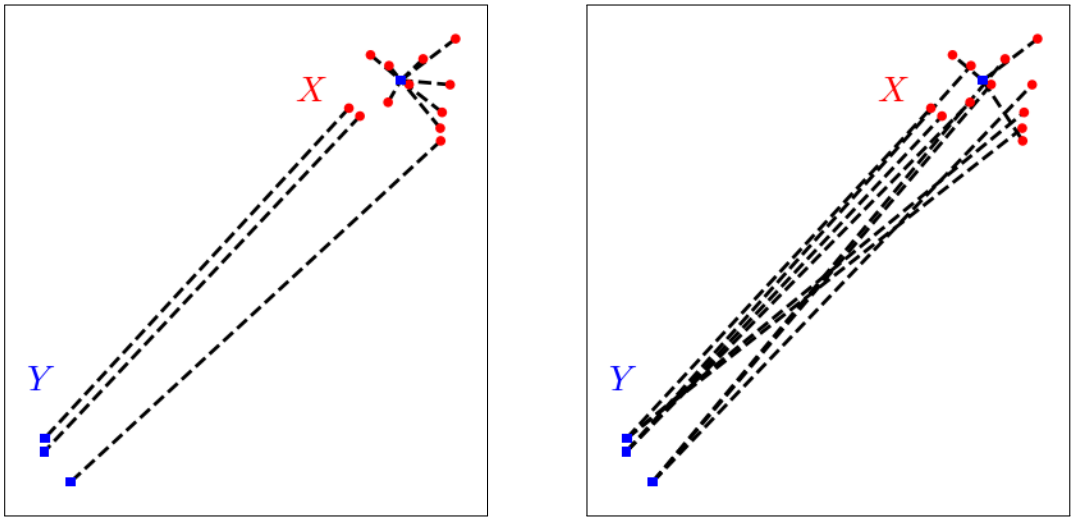

Figure 3.2: The left picture illustrates the surjection distance and the right picture illustrates the fair surjection distance. The distances are defined to be the sum of the lengths of the dotted lines.

As can be seen in Figure 3.2, the fair surjection distributes the instances of X uniformly over the instances of Y. The surjection distance maps almost all instances of X to a single instance of Y. However, this point of Y is clearly an outlier, and this unrepresentative point heavily influences the surjection distance. The influence of this outlier instance is diminished in the fair surjection distance by distributing the instances. We will illustrate this effect by removing the outlier. The effect on the distances can be seen in Figure 3.3. Clearly the computation of the surjection distance changed drastically while the change is more subtle for the fair surjection distance. This illustrates the distinction between the two and shows that the fair surjection distance is less susceptible to outliers.

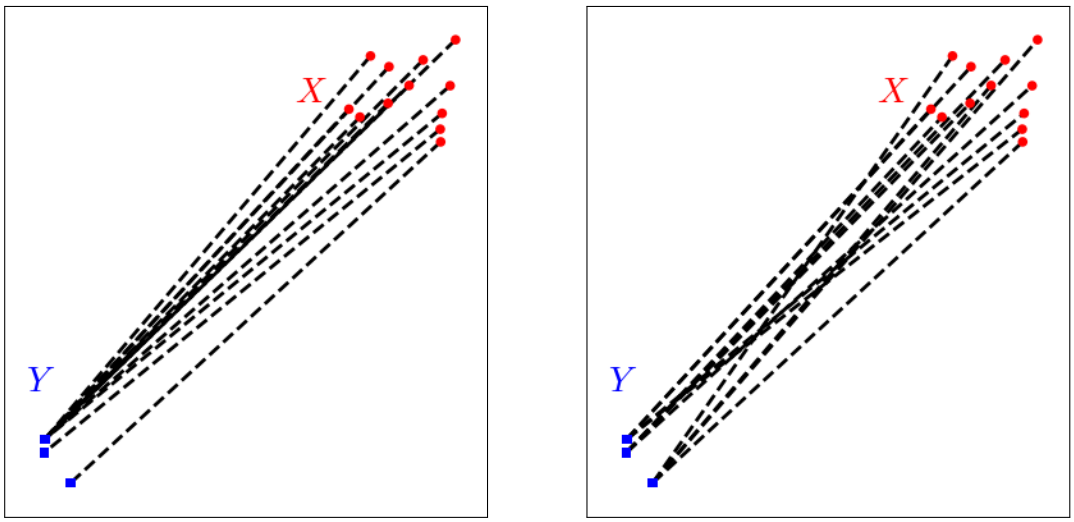

Figure 3.3: The outlier point from Y is removed such that now $|X| = 12$ and $|Y| = 3$. The left picture represents the surjection distance, while the right picture represents the fair surjection distance.

Link distance (δ_l)
Let X, Y be two non-empty subsets of a metric space (M, d). For the last distance measure based on relations between multisets we consider compatibility relations. A compatibility relation is a relation $R \subseteq X \times Y$ satisfying

1. R is serial:
$$\forall x \in X : \exists y \in Y : (x, y) \in R$$
2. R is inverse serial:
$$\forall y \in Y : \exists x \in X : (x, y) \in R$$

We define the following distance measure:

$$\delta_l(X, Y) = \min_R \sum_{(x,y)\in R} d(x, y),$$

where the minimum is taken over all possible compatibility relations R.

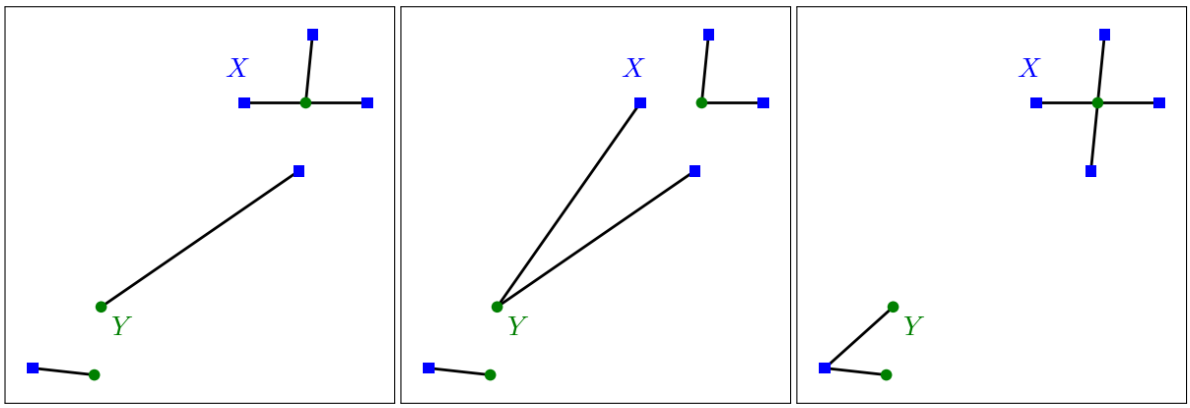

Figure 3.4: An illustration of the surjection distance, the fair surjection distance, and the link distance respectively between two bags. The instances of bag X are represented by blue squares, while the instances of Y are represented by green dots. The distances are defined as the sum of the length of the black lines between the instances.

In Figure 3.4 all three distances are illustrated using the same pair of bags. It should be noted that every (fair) surjection is also a compatibility relation. Therefore, there are cases where the fair surjection distance, surjection distance and link distance are all computed using the same relation. An example of such a case can be seen in Figure 3.5.

Theorem 3.1.9. *The surjection distance, fair surjection distance and link distance are all semi-metrics.*

Proof. Semi-metrics have to satisfy the following conditions:

$$\begin{array}{cl} \delta(x,y) \geq 0 & \text{(non-negative)} \\ \delta(x,y) = 0 \iff x = y & \text{(identity of indiscernibles)} \\ \delta(x,y) = \delta(y,x) & \text{(symmetry)} \end{array}$$

We will start by proving these conditions for all three distances.

- **Non-negative**
  All three distances are defined as a sum of distances between points, and therefore the non-negativity is inherited from the metric between points.

- **Identity of indiscernibles**
  We will demonstrate this proof for the surjection distance. This condition can be proven for the fair surjection distance and link distance in a similar way. Suppose there exist bags X, Y such that $\delta(X, Y) = 0$. It follows from the definition that each element of X is connected to at least one element of Y and vice versa. Since the surjection distance is defined as a sum of distances between points, it follows that every distance between points in the summation must be equal to 0. This proves that $X = Y$. Now suppose $X = Y$. It is clear that the trivial surjection is the surjection that minimizes the sum of distances. Since the distance between every pair of instances is 0, the total sum will equal 0. It now easily follows $\delta_s(X, Y) = 0$.

- **Symmetry**
  The definition of the link distance is completely symmetric, and therefore symmetry is trivial. The surjection distance and fair surjection distance however are defined as a surjection from the largest bag to the smallest bag. Symmetry is once again trivial when the size of the two bags are different. When there are two bags X, Y that contain an equal amount of instances, these surjections become bijections and thus symmetry will still hold.

□

Remark 3.1.10. The triangle inequality does not hold in general for the surjection distance, fair surjection distance and link distance. An example that the triangle inequality does not hold for all bags X, Y, Z can be seen in Figure 3.5. This means that these three distances are not metrics in general.

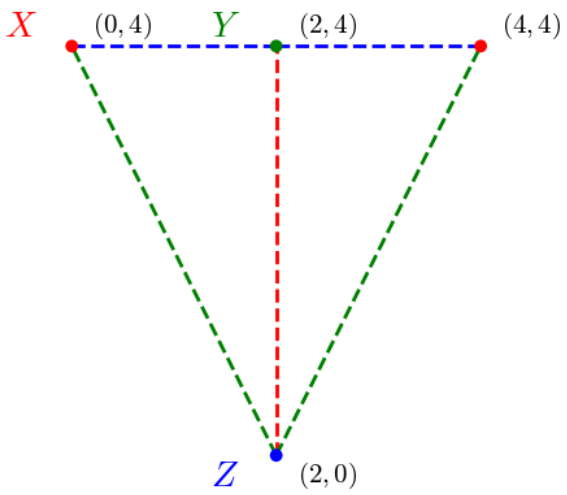

Figure 3.5: X, Y and Z are represented by the red, green and blue dots respectively. These bags are chosen such that the surjection distance, fair surjection distance and link distance are computed using the same relation, resulting in the same distance. Therefore, we will write δ to denote any of the three relation-based distances. Note that we consider the euclidean distance to determine the distance between instances. $\delta(X, Y)$ equals the sum of the lengths of the blue dotted lines, $\delta(Y, Z)$ equals the length of the red dotted line, and $\delta(X, Z)$ equals the sum of the lengths of the green dotted lines. This results in $\delta(X, Y) = 4$, $\delta(Y, Z) = 4$, and $\delta(X, Z) = 4\sqrt{5}$.

Remark 3.1.11. When all bags contain the same amount of instances, the triangle inequality does hold for the surjection distance and the fair surjection distance.

Proof. First of all it should be noted that every surjection between bags of equal size is automatically a fair surjection, more precisely every surjection will be a bijection. Therefore, the fair surjection distance and surjection distance are completely the same when the bags have the same amount of instances. Suppose we have three bags X, Y, Z with $|X| = |Y| = |Z|$. Recall that the surjection distance δ_s between two bags A, B is defined using a surjection $\eta : A \to B$ that minimizes

$$\sum_{(a,b)\in\eta} d(a, b),$$

and that in this particular case the surjection is a bijection because the size of the domain and the image is equal. Therefore, we can conclude that η minimizes

$$c_{AB}(\eta) := \sum_{a\in A} d(a, \eta(a)).$$

We will write η_1, η_2, η_3 to denote the bijections used in $\delta_s(X, Y), \delta_s(Y, Z)$ and $\delta_s(X, Z)$

respectively. In other words:

$$\begin{aligned} c_{XY}(\eta_1) &= \delta_s(X, Y) \\ c_{YZ}(\eta_2) &= \delta_s(Y, Z) \\ c_{XZ}(\eta_3) &= \delta_s(X, Z). \end{aligned}$$

We will now construct a bijection η by considering the composition of η_1 and η_2:

$$\eta : X \to Z : x \to \eta_2(\eta_1(x)).$$

It now follows from the triangle inequality of the metric d that

$$\begin{aligned} c_{XZ}(\eta) &= \sum_{x \in X} d(x, \eta(x)) \\ &\leq \sum_{x \in X} d(x, \eta_1(x)) + d(\eta_1(x), \eta(x)) \\ &= \sum_{x \in X} d(x, \eta_1(x)) + \sum_{x \in X} d(\eta_1(x), \eta_2(\eta_1(x))) \\ &= \sum_{x \in X} d(x, \eta_1(x)) + \sum_{y \in Y} d(y, \eta_2(y)) \\ &= c_{XY}(\eta_1) + c_{YZ}(\eta_2) \\ &= \delta_s(X, Y) + \delta_s(Y, Z). \end{aligned}$$

Recall that η_3 is by definition the bijection that minimizes c_{XZ}, and therefore we get

$$\begin{aligned} \delta_s(X, Z) &= c_{XZ}(\eta_3) \\ &\leq c_{XZ}(\eta) \\ &\leq \delta_s(X, Y) + \delta_s(Y, Z). \end{aligned}$$

Thus, the triangle inequality holds.

□

3.2 Relation between distance and similarity

Intuitively the concepts of distance and similarity are related. When the distance in the feature space between two objects is small, they must be similar.

Given a certain metric we can always define a similarity measure by transforming it with a function $f : \mathbb{R}^+ \to [0, 1]$. As we have seen, a similarity measure $S(x, y) = f(d(x, y))$ has only 2 requirements: the relation should be symmetric and reflexive. Symmetry is trivially inherited from the symmetry of a distance function. However, to satisfy the reflexive property, it is required that $f(0) = 1$.

These conditions are not difficult to meet, meaning that there are many potential candidates for the function f. However, we can add some reasonable assumptions for f. The function f should also be strictly decreasing. This reflects the intuition that objects that lie further apart are less similar. Possible functions that satisfy these conditions are:

- $f(x) = \frac{1}{1+ax}$
- $f(x) = e^{-ax}$
- $f(x) = \frac{2}{\pi} \arctan(\frac{1}{ax})$

with $a \in \mathbb{R}_0^+$

Defining a metric given a similarity measure is not as trivial because only demanding that a transformation g is strictly decreasing does not necessarily result in a relation that satisfies the triangle inequality or the identity of indiscernibles. However, when the similarity measure S satisfies the separation property, a distance measure d defined as $d(x, y) = g(S(x, y))$ with $g : [0, 1] \to \mathbb{R}_0^+$ strictly decreasing, and $g(1) = 0$, does result in a semi-metric. The symmetry and non-negativity are inherited from the similarity measure. If $x = y$ then $d(x, y) = g(S(x, y)) = g(1) = 0$. Since g is strictly increasing, $d(x, y) = g(S(x, y)) = 0$ implies $S(x, y) = 1$. The separation property now implies $x = y$, thus d is a semi-metric.

We will now examine if there are functions f such that the resulting similarity measures satisfy any of the additional properties in Definition 2.3.2.

Theorem 3.2.1. *Consider a convex metric space (M,d). There is no function f and metric d such that $S(x, y) = f(d(x, y))$ is a $T_{\min}$-transitive similarity measure.*

Proof. Suppose S were $T_{\min}$-transitive, i.e.:

$$\forall x, y, z : \min(S(x, y), S(y, z)) \leq S(x, z).$$

This implies that

$$f(d(x, y)) \leq f(d(x, z)) \vee f(d(y, z)) \leq f(d(x, z))$$

and because f is assumed to be strictly decreasing we have that

$$d(x, y) \geq d(x, z) \vee d(y, z) \geq d(x, z).$$

This would mean that there are no points x, y, z such that

$$d(x, y) < d(x, z) \wedge d(y, z) < d(x, z).$$

However, because (M, d) is a convex metric space, for every $x, z \in M$ there exists a y different from x, z such that

$$d(x, y) + d(y, z) = d(x, z)$$

Because $x \neq y \neq z$, the non-negativity and identity of indiscernibles of d imply that

$$d(x, y) > 0$$
$$d(y, z) > 0$$

Thus, we have found points x, y, z for which

$$d(x, y) < d(x, z) \wedge d(y, z) < d(x, z),$$

and therefore S is not $T_{\min} - transitive$. □

Theorem 3.2.2. *If f is a convex, strictly decreasing, continuous function with $dom(f) \subseteq \mathbb{R}^+$, $0 \in dom(f)$ and $f(0) = 1$, and d is a metric, then $S(x, y) = f(d(x, y))$ is a T_L-transitive similarity measure.*

Proof. Suppose f is a convex, strictly decreasing, continuous function with $dom(f) \subseteq \mathbb{R}^+$, $0 \in dom(f)$ and $f(0) = 1$. f is convex if and only if $\forall t \in [0, 1]$ and $x_1, x_2 \in dom(f)$:

$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$$

Suppose we have $a, b \in dom(f) : a + b \in dom(f) \setminus \{0\}$. If we choose $x_1 = a + b$ and $x_2 = 0$, then we can choose $t_1 = \frac{a}{a+b}$, such that $t_1 x_1 = a$. We get

$$f(a) = f(t_1 x_1 + (1 - t_1)x_2) \leq t_1 f(a + b) + (1 - t_1)f(0) = t_1 f(a + b) + (1 - t_1)$$

and because $b = a + b - a = x_1 - a = x_1 - t_1 x_1 = (1 - t_1)x_1$,

$$f(b) = f((1 - t_1)x_1 + t_1 x_2) \leq (1 - t_1)f(a + b) + t_1 f(0) = (1 - t_1)f(a + b) + t_1$$

Together these give

$$f(a) + f(b) \leq f(a + b) + 1 = f(a + b) + f(0)$$

Thus for all $a, b \in dom(f)$ for which $a + b \in dom(f) \setminus \{0\}$ it holds that

$$f(a) + f(b) \leq f(a + b) + f(0) \tag{3.1}$$

The similarity measure $S(x, y) = f(d(x, y))$ is T_L-transitive iff for all points x, y, z of the metric space

$$\max(0, f(d(x, y)) + f(d(y, z)) - 1) \leq f(d(x, z))$$

We will denote the distances $d(x, y), d(y, z), d(x, z)$ as a, b, c respectively. This substitution gives

$$\max(0, f(a) + f(b) - 1) \leq f(c)$$

which is equivalent with

$$f(a) + f(b) \geq 1 \Rightarrow f(a) + f(b) \leq f(c) + 1. \tag{3.2}$$

Suppose $f(a) + f(b) \geq 1$. Since f is decreasing, this condition is more difficult to satisfy when c is higher. Note that the triangle inequality implies that $a + b \geq c$. The largest

feasible c is thus $c = \min(a + b, \sup(dom(f)))$.
If $c = a + b$ then $a + b \in dom(f)$ and therefore the condition follows from Equation (3.1).

The case where $c = \sup(dom(f))$ however implies that $f(c) = \inf_{x \in dom(f)} f(x)$ because f is strictly decreasing, and the condition becomes $\forall a, b \in dom(f)$

$$f(a) + f(b) \geq 1 \Rightarrow f(a) + f(b) \leq 1 + \inf_{x \in dom(f)} f(x).$$

Suppose that $f(a) + f(b) > 1 + \inf(f)$. We already found that $c < a + b$ and since f is strictly decreasing we have $f(a) < f(c - b)$. Combined with $f(a) + f(b) > 1 + \inf(f)$ this becomes:

$$1 + \inf(f) < f(a) + f(b) < f(c - b) + f(b)$$

And since $1 + \inf(f) = f(0) + f(c)$, we find

$$f(0) + f(c) < f(c - b) + f(b).$$

This is however a contradiction with the convexity of f, which states $\forall t \in [0, 1]$:

$$\begin{aligned} f(tc + (1 - t)0) &\leq tf(c) + (1 - t)f(0) = 1 - t + t \inf(f) \\ f(t0 + (1 - t)c) &\leq tf(0) + (1 - t)f(c) = t + (1 - t) \inf(f). \end{aligned}$$

Together this becomes $f(tc) + f((1 - t)c) \leq 1 + \inf(f)$, which for $t = \frac{c-b}{c}$ becomes

$$f(c - b) + f(b) \leq 1 + \inf(f) = f(0) + f(c).$$

The contradiction implies that the condition of T-transitivity still holds. □

Remark 3.2.3. The convexity of f is a sufficient condition for $S(x, y) = f(d(x, y))$ being a T_L-transitive similarity measure, however it is not necessary. An example of a non-convex function f that leads to a T_L-transitive similarity measure is:

$$f(x) = \begin{cases} \frac{1}{10x+1} & \text{if } 0 \leq x \leq 1 \\ \frac{-14616}{1331}x^2 + \frac{29122}{1331}x + \frac{-14385}{1331} & \text{if } 1 < x \leq \frac{137}{126} \end{cases}$$

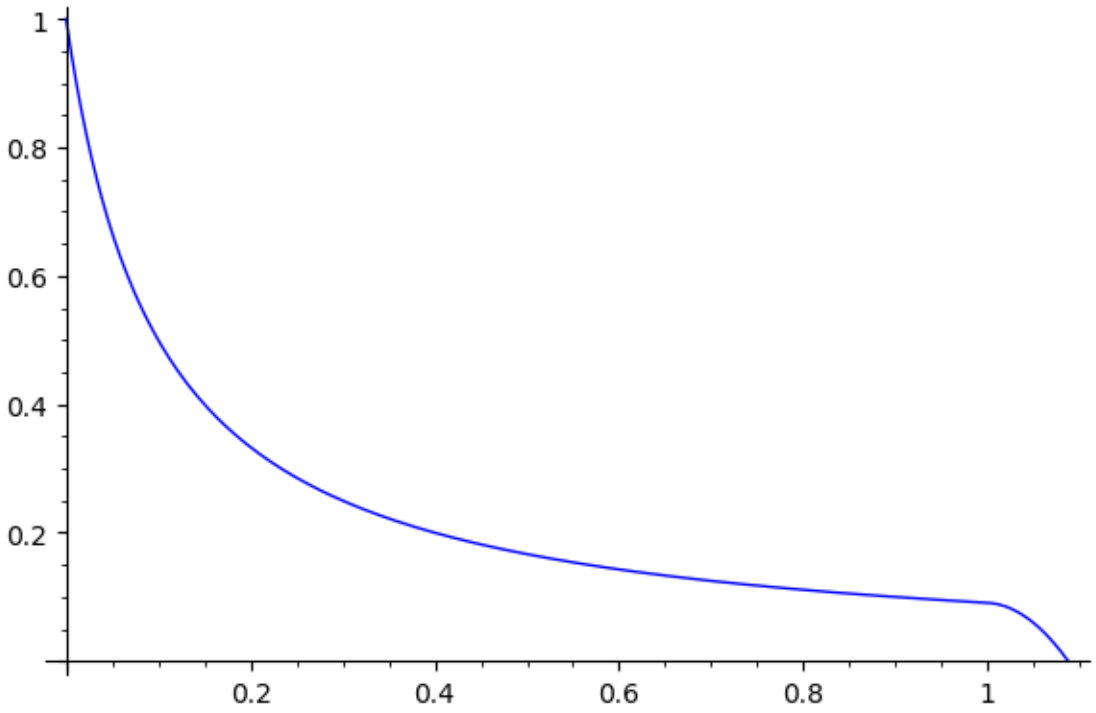

Figure 3.6: Example of a non-convex function that leads to T_L-transitive similarity measure.

This function (shown in Figure 3.6) contains part of a concave parabola and is therefore not convex. This function has been constructed in such a way that f is differentiable in 1. From Equation 3.2 and $c \geq a + b$, it follows that this function leads to a T_L-transitive similarity measure iff for all $a, b \in dom(f)$ for which $a + b \in dom(f)$, the following condition holds:

$$1 \leq f(a) + f(b) \Rightarrow f(a) + f(b) \leq 1 + f(a + b)$$

or equivalently,

$$1 \leq f(a) + f(b) \Rightarrow f(b) - f(a + b) \leq f(0) - f(a).$$

Without loss of generality we can assume that $a \leq b$. This leads to the condition $f(a) \geq f(b)$ and together with $f(a) + f(b) \geq 1$ this leads to $a \leq \frac{1}{10}$. Using the derivative we can derive bounds for the differences $f(b) - f(a + b)$ and $f(0) - f(a)$:

$$f(0) - f(a) \geq -a \max(f'(a), f'(0)) \geq -af'(a),$$
$$f(b) - f(a + b) \leq -a \min(f'(b), f'(a + b)) \leq -af'(a).$$

Where the second inequality is found by noticing that both a and $a + b$ lie between a and $\frac{137}{126}$ and the derivative in this interval will be smallest in a. Therefore f will lead to a T_L-transitive similarity measure.

Theorem 3.2.4. *If d is a (semi-)metric and f is a strictly decreasing function with $f(0) = 1$, then S(x,y)=f(d(x,y)) will satisfy the separation property.*

Proof. The separation property states that for all x, y

$$S(x, y) = 1 \iff x = y,$$
$$f(d(x, y)) = 1 \iff x = y.$$

Because f is strictly decreasing and $f(0) = 1$, this is equivalent to

$$d(x, y) = 0 \iff x = y.$$

and this is true because d is assumed to be a (semi-)metric. □

3.3 Computing the relation-based distances

The three novel relation-based distances all come down to solving an optimization problem. However, the parameters in these optimization problems are relations that satisfy specific conditions. In most cases there is a large range of relations that satisfy the necessary conditions. This implies that the search space can be very large, and thus it is not feasible to consider all the possible relations between two bags. However, these optimization problems can be translated to optimization problems on graphs and networks [8]. More specifically the optimization problems that we have discussed in Section 2.4 for which many known algorithms already exist. In this section we will prove that these relation optimization problems are indeed equivalent to graph optimization problems. In our experimental study these bag distances are computed using these equivalent problems in order to lower the computational complexity.

However, it should be mentioned that even the best algorithms for solving the equivalent graph theory problems have a computational complexity that is considerably larger than the computational complexity of the Hausdorff distances. The best known complexity for solving the minimum weight matching problem [9] is $O(nm + n^2 \log(n))$, where n is the number of vertices and m the number of edges. All graphs that we will consider will be complete graphs, which means that this can be simplified to $O(n^3)$. The best known complexity for the minimal-cost maximum-flow problem for weighted bipartite graph [10] can also be simplified to $O(n^3)$ because our problems will only consider complete bipartite graphs.

3.3.1 Computing the surjection distance δ_s

To determine $\delta_s(X, Y)$ for $X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_m\}$ (assume $n \geq m$) multisets in a metric space (M, d), a surjection η must be found that minimizes

$$\sum_{(x,y)\in\eta} d(x, y).$$

We will show that this is equivalent to finding a minimum weight perfect matching in a certain weighted graph G. From X, Y the complete bipartite graph $G = (X \cup Z, E, w)$ is constructed as follows: first of all, additional vertices $\{z_1, \ldots, z_k\}$ are added to the smaller bipartition class such that both classes have an equal amount of vertices.

$$Z = Y \cup \{z_1, \ldots z_k\} \text{ with } k = n - m$$
$$E = \{(x, z) | x \in X, z \in Z\}$$
$$\forall e = (x, z) \in E : w(e) = \begin{cases} d(x, z) & \text{if } z \in Y \\ d(x, Y) & \text{if } z \notin Y \end{cases}$$

An example of this construction is shown in Figure 3.7. Since G is a complete, balanced bipartite graph, G is guaranteed to have a perfect matching M by Lemma 2.4.9. Remember that we denote the weight of the matching as $w(M)$.

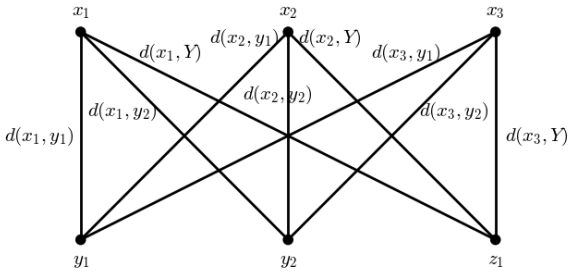

Figure 3.7: The graph G is constructed from the bags X and Y.

Theorem 3.3.1. *Let M be an arbitrary minimum weight perfect matching of a graph G defined as above from two multisets X, Y. Then $\delta_s(X, Y) = w(M)$.*

Proof. $\delta_s(X, Y)$ makes use of an optimal surjection η. We will show that this surjection gives rise to a perfect matching M' of G with $w(M') \leq \delta_s(X, Y)$. We construct M' by taking all edges of G of the form $(x_{\sigma(j)}, y_j)$ with $y_j \in Y$ and $\sigma(j) = min(\{i | \eta(x_i) = y_j\})$. The remaining vertices of Z are $Z \setminus Y = \{z_1, \ldots, z_k\}$. These vertices can be arbitrarily paired up with the remaining vertices of X because $\forall x \in X, z \in Z \setminus Y : w((x, y)) = d(x, Y)$. M' is clearly a perfect matching of G because it saturates all vertices of $X \cup Z$. We have the following edges in M':

1. Every vertex $y \in Y$ is connected to a vertex $x \in \eta^{-1}(y)$ and for these edges we thus have that $w((x, y)) = d(x, y)$.
2. Every vertex $z \in Z \setminus Y$ is connected to a vertex x for which $w((x, z)) = \min_{y \in Y} d(x, y) \leq d(x, \eta(x))$.

Therefore, we know that $w(M')$ is smaller than $\delta_s(X, Y)$. M is assumed to be a perfect matching of minimal weight, so we find $w(M) \leq w(M') \leq \delta_s(X, y)$.

Conversely, we can define a surjection η' from the matching M for which

$$c(\eta') := \sum_{(x,y) \in \eta'} d(x, y) \leq w(M).$$

Since M is a perfect matching every vertex $x_i \in X$ is saturated and thus connected to a single vertex of Z. We will now use these edges to construct the surjection η'. For every edge $e = (x, z) \in M$ we define:

$$\eta'(x) = \begin{cases} z & \text{if } z \in Y \\ \arg\min_{y \in Y}(d(x, y)) & \text{if } z \notin Y \end{cases}$$

η' is clearly a surjection because for each $y_j \in Y$ we have an edge (x_i, y_j) in M, since M was a perfect matching. Furthermore $\eta'(x_i)$ is defined such that for every edge (x_i, z) of M, $d(x_i, \eta(x_i)) = w((x_i, z))$. Therefore $w(M) \geq c(\eta') \geq \delta_s(X, Y)$, where the second inequality comes from the fact that δ_s uses a surjection η for which $c(\eta)$ is minimal. The result now follows from the two inequalities. □

3.3.2 Computing the fair surjection distance δ_{fs}

The fair surjection distance $\delta_{fs}(X, Y)$ for $X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_m\}$ (assume $n \geq m$), can be computed by solving a network flow problem, more specifically by finding a maximal flow with minimal cost. We construct a network $N = (V, E, cap, c, s, t)$, with $V = X \cup Y \cup \{s, t\}$. s and t are two new vertices that are added to V to act as the source and sink respectively. We construct E, cap and c as follows:

- For every $x_i \in X$ there is an edge $e = (s, x_i)$ with $cap(e) = 1$ and $c(e) = 0$.
- For every $x_i \in X$ and every $y_j \in Y$ there is an edge $e = (x_i, y_j)$ with $cap(e) = 1$ and $c(e) = d(x_i, y_j)$.
- For every $y_j \in Y$ there is an edge $e = (s, y_j)$ with $cap(e) = 1$ and $c(e) = \Omega$, where Ω is an arbitrary real number greater than the sum of all distances between points.
- For every $y_j \in Y$ there is an edge $e = (y_j, t)$ with $cap(e) = \lceil n/m \rceil$ and $c(e) = 0$.

An example of this construction is shown in Figure 3.8. This construction guarantees that at most $\lceil n/m \rceil$ units of flow can arrive in each vertex y_i, which is also the maximum possible size of the pre-image $\xi^{-1}(y_i)$ if ξ is a fair surjection.

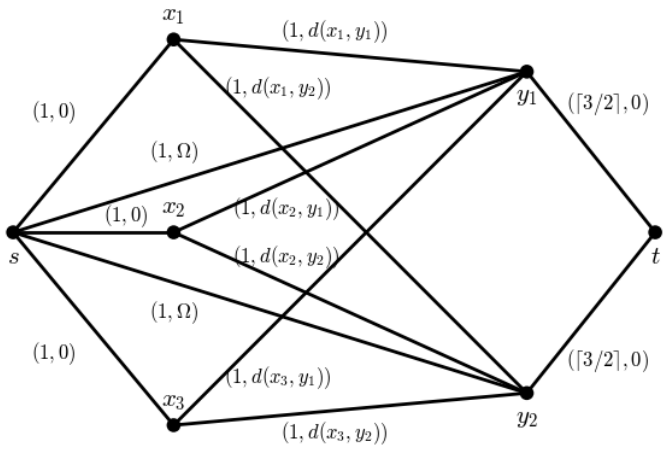

Figure 3.8: The network N constructed from the bags $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2\}$. Note that this is a directed graph, but for clarity the arrows were omitted. Instead the graph is illustrated in such a way that every edge goes from its left vertex to its right vertex. The edge labels in this figure denote the capacity $cap(e)$ and cost $c(e)$ for every edge e.

Theorem 3.3.2. *Let f be a maximum $s-t$ flow of minimum cost in a network N defined as above from two multisets X, Y. Then $\delta_{fs}(X, Y) = c(f) - k\Omega$, with $k = \lceil n/m \rceil m - n$.*

Proof. We first show that $c(f) \geq \delta_{fs}(X, Y) + k\Omega$. Assume f is a maximal flow of minimum cost. It can be easily verified that the maximum flow possible is $m\lceil n/m \rceil$ by looking at the vertices of the form y_i. Each of these vertices has an outgoing edge with capacity $\lceil n/m \rceil$, which acts as a bottleneck. Since there are m such vertices, an upper bound of the maximum flow is thus $m\lceil n/m \rceil$. To show that this flow can actually be achieved, we can construct a flow. We partition X in classes $X_1, \ldots, X_{m+1}$, where $|X_1| = \cdots = |X_m| = \lfloor n/m \rfloor$ and X_{m+1} contains the remaining $n - m\lfloor n/m \rfloor$ vertices of X. We now assign a flow of 1 to all edges of the form (s, x) with $x \in X_1 \cup \cdots \cup X_m$ and edges of the form (x, y_i) with $x \in X_i$ and $i \in \{1, \ldots, m\}$. Every vertex $y_i \in Y$ now receives a flow of $\lfloor n/m \rfloor$. If $\lfloor n/m \rfloor = \lceil n/m \rceil$, no extra flow is needed to vertices $y_i \in Y$. If this is not the case we can assign a flow of 1 to all edges of the form (s, y_i). Either way we can now assign an outgoing flow of $\lceil n/m \rceil$ to edges of the form (y_i, t), leading to a total flow of $m\lceil n/m \rceil$. However, f is supposed to be the flow with minimum cost that achieves this maximal flow. To achieve the maximal flow every vertex $y_i \in Y$ has to receive a flow of $\lceil n/m \rceil$. There are two types of paths the flow can take from s to a vertex y_i: on the one hand there are paths that go through a vertex $x_j \in X$, and on the other hand there is a path straight from s to y_i. The direct path has cost Ω, while the path through a vertex x_j has cost $d(x_j, y_i)$. By definition the direct path with cost Ω is the path of highest cost. Since f is a flow of minimum cost, the use of these direct paths will be minimized. Thus, we can infer that f has to assign a flow of 1 to all edges of the form (s, x_i). However, the flow that can go through a vertex x_i is capped by the capacity of the edges of the form (s, x_i), such that only a flow of n can flow through such paths. Therefore a flow of 1 has to be assigned to k edges of the form (s, y_j) such that the maximal flow can be obtained. We can now construct a fair surjection

ξ' for which

$$\sum_{(x,y)\in\xi} d(x,y) = c(f) - k\Omega$$

We construct ξ' as follows:

$$(x_i, y_j) \in \xi' \iff f((x_i, y_j)) = 1.$$

We have to show that this is indeed a fair surjection, i.e.

$$\forall y_1, y_2 \in im(\xi') : ||\xi'^{-1}(y_1)| - |\xi'^{-1}(y_2)|| \leq 1.$$

We can rewrite this condition such that a surjection ξ is fair if and only if

$$\forall y \in im(\xi') : \lfloor n/m \rfloor \leq |\xi'^{-1}(y)| \leq \lceil n/m \rceil.$$

Because all vertices $y_j \in Y$ only have a single outgoing edge, by construction all going to t with capacity $\lceil n/m \rceil$, and all of its incoming edges have a flow of either 0 or 1, we can conclude that each vertex y_j will have exactly $\lceil n/m \rceil$ incoming edges with a flow of 1. This means that for every $y_j \in Y$ there exist at most $\lceil n/m \rceil$ vertices $x \in X$ for which $f(x, y_j) = 1$. Therefore we can conclude that

$$\forall y \in im(\xi') : |\xi'^{-1}(y)| \leq \lceil n/m \rceil.$$

We previously discussed that every vertex $x_i \in X$ has an incoming flow of 1 that is passed through to a vertex $y_j \in Y$. Because the total flow is $m\lceil n/m \rceil$, the outgoing flow and thus the incoming flow of every $y_j \in Y$ must be $\lceil n/m \rceil$. Suppose a vertex $y_1 \in Y$ receives less than $\lfloor n/m \rfloor$ units of flow coming from vertices in X. There are two cases now. In the case where $\lfloor n/m \rfloor = n/m = \lceil n/m \rceil$, having a vertex y_1 that does not receive $\lfloor n/m \rfloor$ units of flow from vertices in X means that the edge (s, y_1) must have flow going through it. However, this is in contradiction with the minimal cost of f, because it can be easily verified that a flow using no edges of the form (s, y_i) can be constructed. This flow would have a lower cost because it doesn't have any edges with a cost of Ω. The case where $\lfloor n/m \rfloor < n/m < \lceil n/m \rceil$ and y_1 receives less than $\lfloor n/m \rfloor$ units of flow come from vertices $x_i \in X$, implicates that the remaining flow must come from the edge (s, y_1). However, this means that a flow greater than one goes through the edge (s, y_1), while the capacity of this edge is 1. These contradictions imply

$$\forall y \in im(\xi') : \lfloor n/m \rfloor \leq |\xi'^{-1}(y)|.$$

We can thus conclude that ξ' is fair.
Now to show that the sum of distances of pairs $(x, y) \in \xi'$ added to $k\Omega$ is indeed equal to $c(f)$, it suffices to notice that only edges of the form (x_i, y_j) and (s, y_j) have a cost associated with it. We already mentioned that there are k edges of the form (s, y_j) that have a capacity of 1 and each of these edges attributes a cost of Ω to $c(f)$. We can thus write $c(f)$ as

$$\begin{aligned} c(f) &= \sum_{e\in E} c(e)f(e) \\ &= \sum_{x_i\in X, y_j\in Y} c((x_i, y_j))f((x_i, y_j)) + k\Omega \\ &= \sum_{(x_i,y_j)\in\xi'} d(x_i, y_j) + k\Omega \end{aligned}$$

And because δ_{fs} is minimized over all fair surjections, we have that

$$\delta_{fs}(X,Y) + k\Omega \leq \sum_{(x_i,y_j)\in\xi'} d(x_i,y_j) + k\Omega \leq c(f).$$

Now we will show that $\delta_{fs}(X,Y) + k\Omega \geq c(f)$. Assume we have a fair surjection ξ that minimizes $\sum_{(x,y)\in\xi} d(x,y)$. We will now construct a maximum flow f_ξ on the network N as follows:

$$f_\xi(e) = \begin{cases} \lceil n/m \rceil & \text{if } e = (y_j, t) \\ 1 & \text{if } e = (s, x_i) \\ & \text{or } e = (s, y_j) \text{ and } |\xi^{-1}(y_j)| < \lceil n/m \rceil \\ & \text{or } e = (x_i, y_j) \text{ and } (x_i, y_j) \in \xi \\ 0 & \text{otherwise} \end{cases}$$

It can be easily verified that f_ξ is indeed a $s-t$ flow. Furthermore, the total flow is $m\lceil n/m \rceil$ and f_ξ is thus a maximal flow. In a similar way as before we can find that

$$c(f_\xi) = \sum_{(x,y)\in\xi} d(x,y) + k\Omega.$$

Since f is a minimal cost maximum flow we find that

$$c(f) \leq c(f_\xi) = \delta_{fs}(X,Y) + k\Omega.$$

Both inequalities together give: $c(f) = \delta_{fs}(X,Y) + k\Omega$. □

An algorithm to find the minimum cost maximal flow in a network is implemented in the `NetworkX` Python package.

3.3.3 Computing the linking distance δ_l

Finally, to compute the link distance $\delta_l(X,Y)$ for $X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_m\}$ (assume $n \geq m$) we will show that this is once again equivalent to finding a minimum weight perfect matching in a weighted graph G. From X, Y the bipartite graph $G = ((X \cup Y') \sqcup (Y \cup X'), E, w)$ is constructed as follows: for every instance $x_i \in X$ we add a new vertex x'_i, similarly we add a new vertex y'_j for every $y_j \in Y$.

$$\begin{aligned} &X' = \{x'_1, \ldots x'_n\} \\ &Y' = \{y'_1, \ldots y'_m\} \\ &E = \{(x_i, y_j) | x_i \in X, y_j \in Y\} \\ &\cup \{(y'_j, x'_i) | y'_j \in Y', x'_i \in X'\} \\ &\cup \{(x_i, x'_i) | x_i \in X, x'_i \in X'\} \\ &\cup \{(y'_i, y_i) | y'_i \in Y', y_i \in Y\} \end{aligned}$$

$$\forall e = (a, b) \in E :$$
$$w(e) = \begin{cases} d(a,b) & \text{if } (a,b) \in \{(x_i, y_j) | x_i \in X, y_j \in Y\} \\ 0 & \text{if } (a,b) \in \{(y'_j, x'_i) | y'_j \in Y', x'_i \in X'\} \\ d(a,Y) & \text{if } (a,b) \in \{(x_i, x'_i) | x_i \in X, x'_i \in X'\} \\ d(b,X) & \text{if } (a,b) \in \{(y'_i, y_i) | y'_i \in Y', y_i \in Y\} \end{cases}$$

An example of this construction can be seen in Figure 3.9.

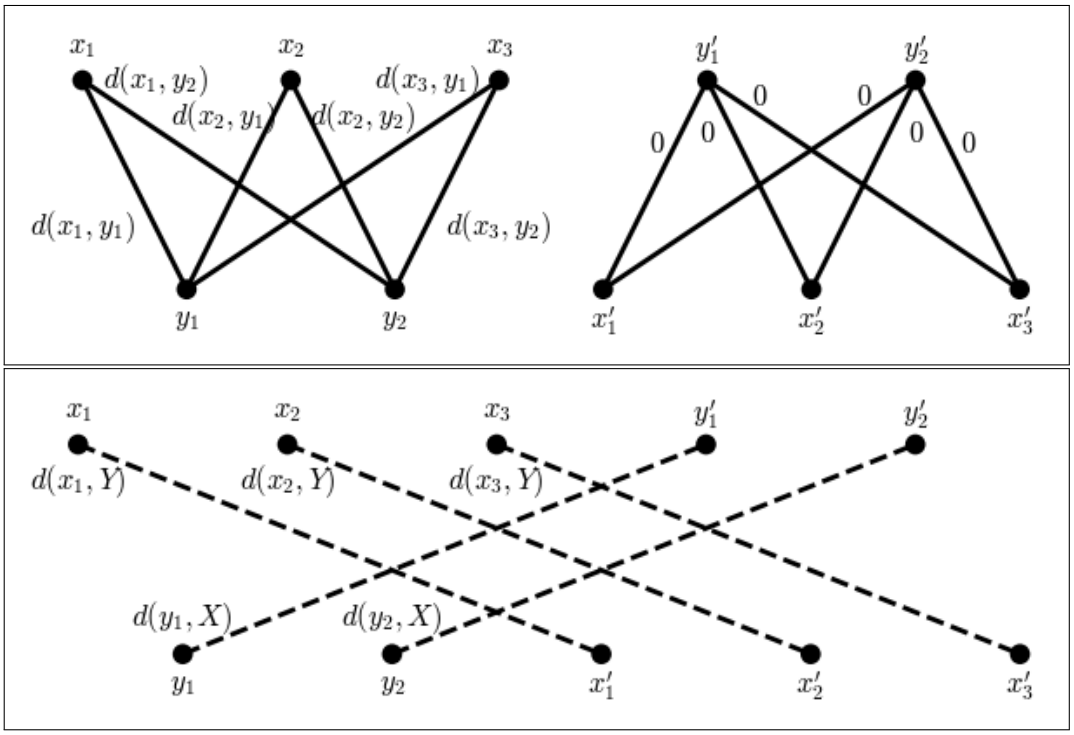

Figure 3.9: The graph G is constructed from the bags X and Y. For clarity this graph has been split up in two parts. The first part contains all the edges of the form (x_i, y_j) and (x'_i, y'_j), while the second part shows all the edges of the form (x_i, x'_i) and (y_i, y'_i) with dotted lines.

Lemma 3.3.3. *Suppose $R \subseteq X \times Y$ is a compatibility relation that is a solution to the optimization problem*

$$\min_{R'} \sum_{(x,y) \in R'} d(x, y).$$

Consider the graph $G(R) = (V, E)$ induced by this compatibility relation in the following manner:

$$V = X \cup Y$$
$$E = \{(x, y \mid (x, y) \in R)\}$$

Then $G(R)$ is a disjoint union of pairs and stars.

Proof. Suppose that $G(R)$ contains a connected subgraph K that is not a pair or a star. Since R is a compatibility relation every vertex has at least degree 1. Therefore K must have at least 2 vertices a, b with a degree of at least 2. Since K is connected there exists a path from a to b. Every vertex on this path has a degree of at least 2. It can be verified that removing any of the edges along this path leads to another compatibility relation R'. Moreover, since d is non-negative this new compatibility relation will have a smaller sum of distances. This is however a contradiction since R already minimized this sum. Therefore, every connected subgraph of $G(R)$ is either a star or a pair. □

Theorem 3.3.4. *Let M be an arbitrary minimum weight perfect matching of a graph G defined as above from two multisets X, Y. Then $\delta_l(X, Y) = w(M)$.*

Proof. $\delta_l(X, Y)$ makes use of an optimal compatibility relation R. We will show that this compatibility relation gives rise to a perfect matching M' of G with $w(M') = \delta_l(X, Y)$. We construct M' by taking all edges of G of the form $(x_{\sigma_1(j)}, y_j)$ and $(x'_{\sigma_1(j)}, y'_j)$ with $y_j \in Y$ the center of a star in $G(R)$ and $\sigma_1(j) = min(\{i|(x_i, y_j) \in R\})$, along with the edges of the form $(x_i, y_{\sigma_2(i)})$ $(x'_i, y'_{\sigma_2(i)})$ with $x_i \in X$ the center of a star in $G(R)$ and $\sigma_2(i) = min(\{j|(x_i, y_j) \in R\})$. The remaining vertices of G are paired up with their copy. The unsaturated vertices are thus paired up as follows: (x_i, x'_i), (y'_j, y_j). These copies are guaranteed to be still unsaturated because the edges were added in pairs $(x_i, y_j), (x'_i, y'_j)$ to the matching M'.

M' is clearly a perfect matching of G, because every vertex is either connected to its copy or to a vertex in the opposing set. Moreover, we can once again verify that $\delta_l(X, Y) = w(M')$. M is assumed to be a perfect matching of minimal weight, so we find $w(M) \leq w(M') = \delta_l(X, Y)$.
On the other hand we can define a linking R' from the matching M for which

$$c(R') = \sum_{(x,y)\in R'} d(x, y) = w(M).$$

We say that $(x_i, y_j) \in R'$ if one of the following conditions is satisfied:

1. $(x_i, y_j) \in M$
2. $(x_i, x'_i) \in M$ and $d(x_i, y_j) = d(x_i, Y)$
3. $(y_j, y'_j) \in M$ and $d(x_i, y_j) = d(y_j, X)$

Since M is a perfect matching in G, we know that for every $x_i \in X$, there will exist at least one $y_j \in Y$ such that (x_i, y_j) satisfies one of the three conditions, and likewise for all $y_j \in Y$. Therefore R' is indeed a linking. Furthermore, it is easy to check that $c(R') = w(M)$. Therefore $w(M) \geq \delta_l(X, Y)$, because δ_l uses a linking R for which $c(R)$ is minimal. The result now follows from the two inequalities. □

4 Multi-instance learning

4.1 Introduction to multi-instance learning

Multi-instance learning (MIL) is a particular machine learning task in supervised learning, where the data does not consist of a set of instances that are individually labeled, but instead each observation consists of a labeled multiset of instances. As previously mentioned, these observations are called bags. This problem was first explored in an artificial problem [11]. Nowadays several real world applications exist, notably in the field of biochemistry, bio-informatics [12], and image classification [13, 14]. In [15] a comprehensive overview of the MIL domain is given.

4.1.1 Structure of multi-instance classification

Training data for multi-instance consists thus of a set of bags $\mathcal{B}$. The set $\mathcal{B}$ consists of m bags $B_1, \ldots, B_m$. Each bag B_i in turn contains n_{B_i} instances and is assigned a class label y_i. Each one of these instances can be represented as a d-dimensional vector. These d dimensions correspond to d features describing each instance. To differentiate the different instances we will use the following naming convention: x_{ij} represents the jth instance from bag number i. The kth value of the d-dimensional vector of a particular instance in a problem with descriptive features $a_1, \ldots, a_d$ can then be described using $a_k(x_{ij})$. A general representation of a multi-instance dataset can thus be seen in Table 4.1.

Bag	$\langle a_1, \ldots, a_d \rangle$	Class Label
B_1	$\langle a_1(x_{11}), \ldots, a_d(x_{11}) \rangle$ $\vdots$ $\langle a_1(x_{1n_{B_1}}), \ldots, a_d(x_{1n_{B_1}}) \rangle$	y_1
B_2	$\langle a_1(x_{21}), \ldots, a_d(x_{21}) \rangle$ $\vdots$ $\langle a_1(x_{2n_{B_2}}), \ldots, a_d(x_{2n_{B_2}}) \rangle$	y_2
$\vdots$	$\vdots$	$\vdots$
B_m	$\langle a_1(x_{m1}), \ldots, a_d(x_{m1}) \rangle$ $\vdots$ $\langle a_1(x_{mn_{Bm}}), \ldots, a_d(x_{mn_{Bm}}) \rangle$	y_m

Table 4.1: General structure of a multi-instance dataset.

4.2 Multi-instance classification

As in other classification problems, the aim of multi-instance classification (MIC) is to predict the label of unseen test subjects using our training dataset of labeled bags. It is important to notice that the unseen test objects are bags themselves. There are a few common assumptions made when dealing with multi-instance classification. These common assumptions will first be explored in Section 4.2.1.

4.2.1 Multi-instance assumptions

The most common setting in MIC is that of binary classification where a bag is either given a *positive* or a *negative* label. The following assumptions offer some possible relations between the instances of a bag and the class label of the bag. An overview of multi-instance assumptions is given in [16].

- **Standard MI assumption**: The standard MI assumption implies that every instance $x_1, \ldots, x_{n_B}$ of a bag B has a class label associated with it. The bag is labeled positive if and only if it contains a positive instance. This in turn implies that a bag is labeled as negative when it only contains negative instances. We model the labels using 0 for the negative label and 1 for the positive label. Assume that the instances have the (hidden) labels $y_1, \ldots, y_{n_B}$, then the label of the bag can be expressed as:

$$c(B) = 1 - \prod_{i=1}^{n_B} (1 - y_i)$$

- **Presence-, threshold-, and count-based assumptions**: These assumptions all generalize the standard assumption by not limiting the instances to either be positively or negatively labeled. Instead the instances each have a hidden label that represents a concept. This means that a bag can be considered positive if and only if it contains at least one instance of every required label. This is exactly the case in the presence-based assumption. The threshold-based assumption generalizes this even further by requiring a certain threshold of instances to be present for each corresponding required label. The count-based assumption is a final generalization by enforcing both a lower and upper bound for the amount of instances corresponding to a required label.

4.2.2 Taxonomy of multi-instance classifiers

Multi-instance classifiers are often divided in two or three categories based on their approach to solving the multi-instance problem. A taxonomy has been proposed in [17].

- **Instance space (IS) paradigm**: The MI classifier relies on the classification of individual instances and then uses an MI assumption to aggregate the label of the bag. In essence every single-instance classifier can be used to classify the instances. This can however be challenging because training instances are not individually labeled, but are contained in a labeled bag.
- **Bag space (BS) paradigm**: A classifier using the BS paradigm considers the bag as a whole. These methods rely on a concept of similarity or distance between bags.
- **Embedded space (ES) paradigm**: The idea of the embedded space paradigm is to represent each bag as a single instance. Therefore a mapping has to be defined to transform each bag in a single feature vector. Subsequently, any single-instance classifier can be used to predict the bag label of unseen bags.

Sometimes the BS and ES paradigm are grouped together as meta-based approaches because neither of them relies on any of the MI assumptions, but instead try to extract instance-independent information about the bags.

4.3 Fuzzy multi-instance classification

The concept of fuzzy multi-instance classification was researched in [18]. In general fuzzy classifiers define a mapping

$$f : \mathcal{F} \to \mathcal{C} : X \to \arg\max_{C \in \mathcal{C}} [C(X)]$$

Here $\mathcal{F}$ is the feature space and $\mathcal{C}$ is the set of possible classes. In our case the feature space is of course the bag space $\mathcal{B}$. The bag X is assigned to the class C for which its membership degree $C(X)$ is largest. The way this class membership is determined varies with the chosen paradigm, which leads to two families of algorithms:

- Instance-based fuzzy multi-instance classifiers (IFMIC family): Based on the instance space paradigm. The value of $C(X)$ is calculated using the class membership degrees $C(x_i)$ of instances $x_i \in X$.
- Bag-based fuzzy multi-instance classifiers (BFMIC family): Based on the bag space paradigm. The value of $C(X)$ is calculated using only bag information. Therefore, a concept of how similar two bags are to each other is needed.

4.3.1 The BFMIC family

The idea of algorithms in the BFMIC family is to compute the similarity between the test bag X and all of the bags $B_1, \ldots, B_m$ of the training dataset. To calculate these bag-wise similarities, a similarity measure has to be chosen. A common similarity measure is to use the similarity measure corresponding to the Hausdorff metric or the average Hausdorff metric. But for the remainder of the explanation we will stick with the general relation R.

The algorithm will thus compute the values of $R(X, B_1), \ldots, R(X, B_m)$. Now for each class label $C \in \mathcal{C}$ we call T_C the set of bags with class label C. To compute the membership of X to a specific class $C \in \mathcal{C}$ we will aggregate the values $R(X, B)$ with $B \in T_C$ using an OWA-operator.

This means that algorithms in the BFMIC family have two hyperparameters:

1. **The relation** R: this relation can be derived from a distance metric using a strictly decreasing function f with $f(0) = 1$.
2. **The weight vector** W: this vector determines the behavior during the OWA-aggregation of the similarity to bags of a certain class.

In essence, the algorithm finds the value of

$$\arg\max_{C \in \mathcal{C}} \left(OWA_W\big(\{R(X, B) | B \in T_C\}\big) \right)$$

and assigns it as the class label of the test bag X. When we choose a relation derived from a distance, and the weight vector such that it determines the maximum, this method assigns to X the class label of the bag that is closest to X with regards to the corresponding distance. The algorithm thus essentially becomes a one-nearest neighbour algorithm.

The BFMIC family can be implemented with the following pseudocode:

Algorithm 1 BFMIC

Input: $\mathcal{B} = B_1, \ldots B_m$ (training bags); $\mathcal{C}$ (set of decision classes); X (test Bags)
Hyperparameters: R (relation), W (weight vector)
Output: Classification for X
best_class ← None
best_score ← 0
for $C \in \mathcal{C}$ **do**
    score ← $OWA_W\big(\{R(X, B) | B \in T_C\}\big)$
    if score ≥ best_score **then**
        best_score ← score
        best_class ← C
    end if
end for
return best_class

5 Experimental study

In this chapter we conduct an experimental study of the performance of different similarity relations in the BFMIC algorithm. We also perform an experiment to investigate the effect of pre-processing the dataset with these different similarity relations. These experiments were implemented in the programming language Python.

5.1 Datasets

To evaluate the performance of the introduced distances between bags for multi-instance classification, an experimental study was performed. We considered 15 different datasets. These datasets originate from the KEEL dataset repository [19] and UCI machine learning repository [20]. The 15 datasets are split into two groups: 10 balanced datasets and 5 imbalanced ones. The characteristics of these datasets are listed in Table 5.1 and Table 5.2.

			Number of bags			Bag Sizes		
Dataset	# features	# instances	# - bags	# + bags	IR	Min	Max	Mean
eastWest	24	213	10	10	1.0	4	16	10.65
elephant	230	1391	100	100	1.0	2	13	6.96
fox	230	1320	100	100	1.0	2	13	6.6
musk1	166	476	45	47	1.05	2	40	5.17
musk2	166	6598	62	39	1.59	2	1044	65.33
mutag-atoms	10	1618	63	125	1.98	5	15	8.61
mutag-bonds	16	3995	125	63	1.98	8	40	21.25
mutag-chains	24	5349	63	125	1.98	8	52	28.45
tiger	230	1220	100	100	1.0	1	13	6.10
westEast	24	213	10	10	1.0	4	16	10.65

Table 5.1: Properties of the balanced datasets.

			Number of bags			Bag Sizes		
Dataset	# features	# instances	# - bags	# + bags	IR	Min	Max	Mean
corel1	9	7947	1900	100	19.0	2	13	3.97
corel2	9	7947	1900	100	19.0	2	13	3.97
corel3	9	7947	1900	100	19.0	2	13	3.97
corel4	9	7947	1900	100	19.0	2	13	3.97
corel5	9	7947	1900	100	19.0	2	13	3.97

Table 5.2: Properties of the imbalanced datasets.

The imbalanced datasets originate from a multiclass multi-instance dataset with 20 mutually exclusive classes, from which 5 classes where chosen at random to perform a one-vs-rest classification. This explains why the properties of all imbalanced datasets are equal.

5.2 Experiment design

We considered 10 different similarity relations in our experiment. All 10 of these similarity relations originate from a distance function between bags. To convert a distance function δ between two bags X, Y we used the function

$$f(x) = \frac{1}{1+x}.$$

Such that the similarity relation between two bags X, Y is defined as

$$S_\delta(X, Y) = \frac{1}{1+\delta(X, Y)}.$$

The 10 similarity relations can be split up in two groups. The first group consists of the OWA-based versions of the Hausdorff distance, average Hausdorff distance, minimal Hausdorff distance and the sum of minimal distances. These distances rely on two hyperparameters. Firstly the internal distance d between instances of the bags, and secondly a pair of OWA-weights. The second group consists of the surjection distance, fair surjection distance, link distance and the version of these three distances that normalize for the number of instances in the bags X, Y. These distances only rely on a single hyperparameter: the internal distance d between instances of the bags.

Furthermore, the BFMIC classifiers rely on a OWA-weight vector to aggregate the results. While we can use the same type of OWA-weight vectors as for the bag distances, it makes sense to somewhat alter the weights such that only the closest k bags are considered. This means that in reality this k is also a hyperparameter. For the internal distance hyperparameter we considered the euclidean distance and the manhattan distance. The OWA-pairs that we considered were: strict weights, additive weights, inverse additive weights, exponential weights, and average weights. Finally, we considered the following values of k: $1, 2, 3, 5, 10$.

The experiment design is similar to that of [21]. The results are generated using a 10-fold cross validation. To determine the hyperparameters the training data is split up further, such that a 5-fold cross validation can be performed on the training dataset with all the different combinations of hyperparameters. The hyperparameters that achieve the best results in this 5-fold cross validation are then used to classify the test data. To determine the best results we considered the accuracy for the balanced datasets and the $F1$-score for the imbalanced datasets. To compare the various similarity relations we use a Friedman test [22] combined with a Conover post-hoc analysis whenever significant results are encountered in the Friedman test.

In Chapter 3 we observed that the fair surjection distance and surjection distance become metrics when the number of instances in all bags are equal. Therefore we considered the following pre-processing steps:

1. Select a number of instances per bag based on the training dataset. To pick the number of instances we considered the following two methods.
    - Pick the median size of the training bags.
    - Plot the bag sizes of the training data using a boxplot. Pick the size of the largest bag that isn't considered an outlier.
2. Reduce the number of instances in bags that contain too many instances and increase the number of instances in bags that contain not enough instances. Once again we considered two different methods.
    - Random undersampling and Random oversampling.
    - SMOTE oversampling [23] and k-means undersampling [24].

Initially the undersampling of bags was not considered, and all the bags were just oversampled using the desired oversampling technique. However, some of the datasets contained bags with an exceptionally large amount of instances. This lead to extreme oversampling, which introduced bias and increased the computational complexity drastically. Therefore, I decided to undersample these outliers. This gives rise to four pre-processing methods

1. Median size and random over- and undersampling.
2. No-outliers size and random over- and undersampling.
3. Median size and SMOTE oversampling and k-means undersampling.
4. No-outliers size and SMOTE oversampling and k-means undersampling.

Each experiment has thus been repeated five times. Once without any pre-processing and with each of the four pre-processing methods. This enables us to compare each of the pre-processing methods to the experiments where no pre-processing method were used. This comparison is done using the Wilcoxon signed-rank test [25].

5.3 Results of the balanced datasets

First we will take a look at the performance of the similarity relations on the balanced datasets. The accuracy of the classifiers along with their average rank in the Friedman test are shown in Tables 5.3 and 5.4.

Dataset	H	avgH	MinH	SumMin
eastWest	0.999	0.945	0.981	0.621
elephant	0.964	0.956	0.957	0.914
fox	0.850	0.875	0.872	0.810
musk1	0.906	0.905	0.932	0.932
musk2	0.714	0.754	0.724	0.724
mutag-atoms	0.989	0.999	1.000	1.000
mutag-bonds	0.960	0.952	0.569	0.569
mutag-chains	0.926	0.930	0.415	0.415
tiger	0.869	0.848	0.888	0.888
westEast	0.971	0.935	0.88	0.888
Average accuracy	0.915	0.910	0.917	0.776
Average rank	3.5 (3)	3.1 (2)	2.9 (1)	6.4 (7)

Table 5.3: Accuracy and average rank of the classifiers using the similarity relations based on the Hausdorff distance, average Hausdorff distance, minimal Hausdorff distance and sum of minimal distances.

Dataset	link	surj	fair_surj	norm_link	norm_surj	norm_fair_surj
eastWest	0.816	0.825	0.798	0.747	0.747	0.742
elephant	0.948	0.948	0.921	0.915	0.915	0.910
fox	0.855	0.855	0.840	0.804	0.804	0.792
musk1	0.923	0.882	0.810	0.866	0.866	0.843
musk2	0.738	0.738	0.731	0.819	0.732	0.775
mutag-atoms	0.970	0.974	0.974	0.982	0.995	0.970
mutag-bonds	0.932	0.930	0.965	0.952	0.948	0.939
mutag-chains	0.923	0.858	0.866	0.945	0.892	0.918
tiger	0.831	0.831	0.833	0.841	0.841	0.807
westEast	0.941	0.934	0.809	0.790	0.790	0.785
Average accuracy	0.888	0.878	0.855	0.866	0.853	0.848
Average rank	5.4 (4)	6.2 (6)	6.8 (9)	5.85 (5)	6.75 (8)	8.1 (10)

Table 5.4: Accuracy and average rank of the classifiers using the similarity relations based on the link distance, surjection distance, fair surjection distance and their normalized versions.

The Friedman test reports a p-value of 0.000338, which is significant on the 5% significance level. Therefore, we can reject the null hypothesis that all similarity relations perform equally well. Hence, we perform a post-hoc analysis. The combinations of similarity relations that came out as significant on the 5% significance level are shown in Table 5.5.

S_1	S_2	p
H	norm_fair_surj	0.023
avgH	norm_fair_surj	0.014
MinH	norm_fair_surj	0.014

Table 5.5: Pairs of similarity relations where a significant difference in performance was found on the 5% significance level.

We can thus conclude that the normalized fair surjection distance performs significantly worse than the Hausdorff distance, average Hausdorff distance and minimal Hausdorff distance.

Now we will take a look at the effect of pre-processing on the performance of the classifiers. In Tables 5.6 and 5.7 the results of the Wilcoxon signed-rank test are shown.

	H		avgH		MinH		SumMin	
	diff	p	diff	p	diff	p	diff	p
(1)	-0.03	0.770	0.06	0.846	0.17	0.557	1.21	0.049
(2)	-0.19	0.020	-0.07	0.770	0.03	0.846	1.39	0.049
(3)	-0.00	1.000	0.02	0.922	-0.22	0.160	1.39	0.037
(4)	-0.10	0.275	0.09	0.492	-0.03	0.232	1.45	0.014

Table 5.6: Difference in average accuracy and the p-value obtained by performing a Wilcoxon signed-rank test to compare the use of no pre-processing to the use of the median + random pre-processing method (1), no-outlier + random method (2), median + SMOTE & k-means pre-processing method (3), and no-outlier + SMOTE & k-means method (4) for the classifiers using similarity relations based on the Hausdorff distance, average Hausdorff distance, minimal Hausdorff distance and sum of minimal distances.

	link		surj		fair_surj		norm_link		norm_surj		norm_fair_surj	
	diff	p	diff	p	diff	p	diff	p	diff	p	diff	p
(1)	0.01	0.557	0.12	0.322	0.21	0.275	0.23	0.232	0.31	0.275	0.36	0.160
(2)	0.01	0.846	-0.16	1.000	0.12	0.557	0.20	0.432	0.06	1.000	0.45	0.193
(3)	-0.02	0.492	0.00	1.000	0.21	0.131	0.09	0.492	0.19	0.492	0.40	0.131
(4)	-0.31	0.432	-0.33	0.105	-0.09	1.000	0.11	0.375	-0.11	0.922	0.06	0.432

Table 5.7: Difference in average accuracy and the p-value obtained by performing a Wilcoxon signed-rank test to compare the use of no pre-processing to the use of the median + random pre-processing method (1), no-outlier + random method (2), median + SMOTE & k-means pre-processing method (3), and no-outlier + SMOTE & k-means method (4) for the classifiers using the similarity relations based on the link distance, surjection distance, fair surjection distance and their normalized versions.

We can conclude that all of the considered pre-processing methods significantly increase the performance of the sum of minimal distances on the 5% significance level. Similarly, we can conclude that using random over- and undersampling to make all bags have the same amount of instances as the largest bag that isn't considered an outlier significantly decreases the performance of the Hausdorff distance. While no other differences are significant on the 5% significance level, it is worth to mention that the average accuracy has increased for several similarity relations.

5.4 Results of the imbalanced datasets

Finally, we will take a look at the performance of the similarity relations on the imbalanced datasets. The $F1$-score of the classifiers along with their average rank in the Friedman test are shown in Tables 5.8 and 5.9.

Dataset	H	avgH	MinH	SumMin
corel1	0.441	0.364	0.280	0.323
corel2	0.503	0.584	0.492	0.534
corel3	0.684	0.746	0.762	0.603
corel4	0.830	0.895	0.942	0.940
corel5	0.522	0.602	0.541	0.556
Average $F1$-score	0.596	0.638	0.604	0.591
Average rank	7.0 (9)	4.4 (2)	6.2 (7)	6.4 (8)

Table 5.8: $F1$-score and average rank of the classifiers using the similarity relations based on the Hausdorff distance, average Hausdorff distance, minimal Hausdorff distance and sum of minimal distances.

Dataset	link	surj	fair_surj	norm_link	norm_surj	norm_fair_surj
corel1	0.379	0.384	0.368	0.413	0.413	0.389
corel2	0.525	0.515	0.529	0.591	0.584	0.581
corel3	0.768	0.660	0.734	0.753	0.729	0.702
corel4	0.687	0.687	0.690	0.900	0.898	0.898
corel5	0.579	0.573	0.606	0.558	0.552	0.541
Average $F1$-score	0.588	0.564	0.585	0.643	0.635	0.622
Average rank	5.3 (4)	7.1 (10)	5.4 (5)	2.9 (1)	4.6 (3)	5.7 (6)

Table 5.9: $F1$-score and average rank of the classifiers using the similarity relations based on the link distance, surjection distance, fair surjection distance and their normalized versions.

The Friedman reports a p-value of 0.514051, which is not significant on the 5% significance level. Therefore we can not reject the null hypothesis that all similarity relations perform equally well. The $F1$-score can be somewhat difficult to interpret, and therefore we also show the balanced accuracy of all classifiers in Tables 5.10 and 5.11.

Dataset	H	avgH	MinH	SumMin
corel1	0.714	0.648	0.597	0.657
corel2	0.733	0.784	0.728	0.744
corel3	0.849	0.850	0.892	0.720
corel4	0.960	0.989	0.987	0.991
corel5	0.752	0.825	0.807	0.772
Average balanced accuracy	0.802	0.819	0.802	0.777
Average rank	7.6 (10)	4.0 (2)	6.0 (6)	6.6 (8)

Table 5.10: Balanced accuracy and average rank of the classifiers using the similarity relations based on the Hausdorff distance, average Hausdorff distance, minimal Hausdorff distance and sum of minimal distances.

Dataset	link	surj	fair_surj	norm_link	norm_surj	norm_fair_surj
corel1	0.654	0.655	0.649	0.674	0.674	0.683
corel2	0.752	0.751	0.757	0.767	0.762	0.762
corel3	0.864	0.861	0.831	0.900	0.899	0.883
corel4	0.976	0.976	0.976	0.986	0.985	0.985
corel5	0.814	0.809	0.806	0.798	0.785	0.784
Average balanced accuracy	0.812	0.810	0.804	0.825	0.821	0.819
Average rank	5.7 (5)	6.1 (7)	6.8 (9)	3.3 (1)	4.2 (3)	4.7 (4)

Table 5.11: Balanced accuracy and average rank of the classifiers using the similarity relations based on the link distance, surjection distance, fair surjection distance and their normalized versions.

The classifier using the similarity relation based on the normalized link distance achieves rank 1 in this experiment. While none of the similarity relations significantly outscores another similarity relation, we can conclude that the new distances are just as viable for classifying problems than the similarity relations based on the Hausdorff distance, and other more standard similarity relations.

It looks like the performance of the BFMIC-classifiers is heavily dependent on the dataset. In Tables 5.10 and 5.11, we can clearly see that the corel1 dataset was the hardest to classify. On the other hand the results for the corel5 dataset are exceptionally good. This is even more surprising because all five of these datasets come from the same multiclass multi-instance problem.

Finally, we will take a look at the effect of pre-processing on the performance of the classifiers. In Tables 5.12 and 5.13 the results of the Wilcoxon signed-rank test are shown.

	H		avgH		MinH		SumMin	
	diff	p	diff	p	diff	p	diff	p
(1)	-0.42	0.312	-0.38	0.062	-0.36	0.625	-0.18	0.312
(2)	-0.30	0.438	-0.26	0.062	-0.53	0.312	0.07	0.438
(3)	-0.37	0.625	-0.26	0.438	-0.54	0.438	-0.08	0.625
(4)	-0.21	0.625	-0.21	0.062	-0.55	0.062	0.05	1.000

Table 5.12: Difference in average $F1$-score and the p-value obtained by performing a Wilcoxon signed-rank test to compare the use of no pre-processing to the use of the median + random pre-processing method (1), no-outlier + random method (2), median + SMOTE & k-means pre-processing method (3), and no-outlier + SMOTE & k-means method (4) for the classifiers using the similarity relations based on the Hausdorff distance, average Hausdorff distance, minimal Hausdorff distance and sum of minimal distances.

	link		surj		fair_surj		norm_link		norm_surj		norm_fair_surj	
	diff	p	diff	p	diff	p	diff	p	diff	p	diff	p
(1)	0.06	1.000	0.15	0.812	0.04	0.812	-0.21	0.312	-0.23	0.125	-0.16	0.125
(2)	0.09	1.000	-0.12	0.812	-0.23	0.625	-0.18	0.062	-0.51	0.062	-0.44	0.062
(3)	-0.10	0.438	-0.13	0.625	-0.23	0.438	-0.38	0.188	-0.49	0.062	-0.43	0.062
(4)	0.04	1.000	-0.02	1.000	-0.13	0.438	-0.25	0.188	-0.39	0.125	-0.32	0.125

Table 5.13: Difference in average $F1$-score and the p-value obtained by performing a Wilcoxon signed-rank test to compare the use of no pre-processing to the use of the median + random pre-processing method (1), no-outlier + random method (2), median + SMOTE & k-means pre-processing method (3), and no-outlier + SMOTE & k-means method (4) for the similarity relations based on the link distance, surjection distance, fair surjection distance and their normalized versions.

Pre-processing seems to have no significant impact on the performance, as not a single pairwise test can be used to reject the null hypothesis. However, it should be noted that the average F1-score decreased in almost all instances, making it questionable if it is beneficial to perform pre-processing on imbalanced datasets. A potential cause for this decrease in average $F1$-score can possibly be explained by the effect of the oversampling. Since there are significantly more negative bags, we potentially have to add a lot of instances to the negative bags. This would only further increase the imbalance in the dataset, making it harder to classify.

6 Conclusion

This master's thesis can be split up in two roughly equal parts. The first part is a theoretical study of useful concepts for multi-instance learning. This started by examining distances between bags that are common in the literature, such as the Hausdorff distance and average Hausdorff distance, and proving some of their properties. Next, we introduced the link distance, surjection distance and fair surjection distance. These distances originally come from the field of philosophy of science to measure the distance between theories in a logical language, but are also suitable as bag distances.

We continued our theoretical study by looking at the relation of similarity and distance between bags. We have shown that we can go from a distance function to a similarity measure using a function f for which $f(0) = 1$. However, we proposed that it only makes sense to consider functions f that are (strictly) decreasing. Using these assumptions we have proven that there exist no function f and metric d such that $S(x, y) = f(d(x, y))$ is a $T_{\min}$-transitive similarity measure. Furthermore, we showed that given a strictly decreasing function f, convexity of f is a sufficient condition for $S(x, y) = f(d(x, y))$ to be a T_L-transitive similarity measure for any metric d. Finally, we have proven that similarity measures coming from a semi-metric using a strictly decreasing function f, satisfy the separation property.

We then focused on the novel link distance, surjection distance, and fair surjection distance. The computation of these distances rely on solving an optimization problem, that at first sight seems completely infeasible to solve in acceptable computational time. However, we have shown that these optimization problems are equivalent to well-known problems in graph theory. These clever conversions in equivalent problems, made it possible to implement these bag distances with an acceptable computational complexity.

Finally, we introduced some concepts from the field of multi-instance learning (MIL), and more specifically those regarding multi-instance classification (MIC). What distinguishes multi-instance classification from other classification problems is that the objects to classify are bags. We introduced the most common multi-instance assumptions and paradigms. Our theoretical results mostly fit in the bag space paradigm, and therefore we introduced the BFMIC algorithm. BFMIC stands for bag-based fuzzy multi-instance classifiers, and is in essence a family of multi-instance classifiers.

The introduction of BFMIC takes us straight to the second part of this master's thesis: the experimental study. The BFMIC algorithm, all the introduced bag distances, and OWA-aggregation were all implemented using the Python programming language. This implementation was then used to conduct an experimental study to compare the performance of different similarity relations on both balanced datasets and imbalanced datasets. Some

of the theoretical results made us consider the use of some pre-processing steps to make the number of instances per bag equal. Therefore we also studied the effect of these pre-processing methods on the performance of the different similarity relations. The experiments were conducted on 15 datasets, 5 of which were imbalanced.

We used a 10-fold cross validation to generate the results. The hyperparameters of each experiment were determined using a 5-fold cross validation on the training dataset. The actual test bags were then evaluated using the best performing hyperparameters in this 5-fold cross validation. We opted for the accuracy as a performance metric for the balanced datasets and the $F1$-score for the imbalanced datasets. The various similarity relations were compared using a Friedman test combined with a Conover post-hoc analysis whenever significant results were encountered in the Friedman test.

Furthermore, each experiment has been repeated five times. Once without any pre-processing and then four more times using different pre-processing methods. This enabled us to compare each of the pre-processing methods to the performance without any pre-processing steps. This comparison was done using the Wilcoxon signed-rank test.

From our experimental results we could conclude that the similarity relations based on the link distance, surjection distance and fair-surjection distance have similar performance to the similarity relations based on the more standard Hausdorff distance. Especially with the imbalanced datasets, we could see a potential performance gain from these novel distances. However, we must keep in mind that the computational complexity of the link distance, surjection distance and fair surjection distance is significantly higher than that of the more simple Hausdorff distance and average Hausdorff distance. Depending on the situation, this trade-off might not be worth it.

Finally, the effect of the pre-processing steps heavily varied between the balanced datasets and the imbalanced datasets. The sum of minimal distances performed significantly better on the balanced datasets when we had applied any pre-processing, and overall the average accuracy rose. The opposite can be said for the imbalanced datasets. While no similarity relation performed significantly worse on the imbalanced dataset, the average $F1$-score almost decreased universally when any form of pre-processing was applied. Presumably, this is a result of our pre-processing methods of choice. Equalizing the amount of instances per bag most certainly adds a lot of instances to negative bags. By doing so we inadvertently increase the imbalance in the dataset, making the classification problem harder.

A Samenvatting

Deze masterthesis gaat hoofdzakelijk over similariteitsmaten voor multi-instance classification. De masterthesis bestaat uit twee grote delen. Enerzijds is er een grotendeels theoretisch deel, en anderzijds is er een deel experimentele studie.

Het theoretische gedeelte start met een opfrissing van enkele concepten uit de vaagverzamelingenleer en de grafentheorie. Deze concepten zullen immers verder in de masterproef benodigd zijn. Vervolgens introduceren we het concept van een bag. Dit concept is heel belangrijk in multi-instance learning. Bij multi-instance classification hebben we namelijk geen label voor individuele instanties. De instanties zitten namelijk gegroepeerd in een multiset, dat een bag wordt genoemd. Elke bag heeft wel een klasselabel. Een potentiële manier om dit probleem op te lossen is om toch de klasselabels van individuele instanties te voorspellen. Vervolgens kunnen we deze voorspellingen combineren tot een voorspelling van het klasselabel van de bag. Deze aanpak heeft echter enkele nadelen. Ten eerste moeten we een manier vinden om de voorspellingen van individuele instanties te combineren tot een voorspelling voor het label van de bag. Een gebruikelijke assumptie is om een bag een positief label toe te kennen als de bag minstens één instantie met een positief label bevat. Een tweede probleem is dat we het label van de bag niet zomaar kunnen veralgemenen naar de labels van de instanties. Indien we toch het klasselabel van de bag veralgemenen naar de klasselabels van de instanties kan veel fout geclassificeerde data geïntroduceerd worden in de trainingsdata.

In deze masterthesis zullen we deze problemen proberen te omzeilen door niet de individuele instanties te classificeren, maar door meteen het label van de bag te voorspellen. Hiervoor bestuderen we eerst afstandsfuncties voor deze bags. In de literatuur bestaan reeds afstanden, zoals de Hausdorff afstand, die het mogelijk maakt om een afstand tussen twee bags te bepalen. In deze masterthesis bewijzen we onder andere dat de Hausdorff afstand ook een metriek is. Vervolgens introduceren we enkele afstanden die nieuw zijn in de context van multi-instance learning. Deze afstanden zijn de surjectie afstand, de faire surjectie afstand en de linking afstand.

Vervolgens bestuderen we wat de relatie is tussen het concept afstand en similariteit. Intuïtief kunnen we hier een betekenisvol verband maken. Hoe groter de similariteit tussen twee objecten, hoe kleiner de afstand tussen deze twee objecten. We bewijzen dat bepaalde overgangsfuncties om van afstand naar similariteit te gaan, voldoen aan enkele bijkomende eigenschappen voor similariteitsmaten. Zo tonen we aan dat de convexiteit van een strikt dalende overgangsfunctie een voldoende voorwaarde is om de T_L-transitiviteit van de bekomen similariteitsmaat te garanderen wanneer we vertrekken vanaf een metriek.

Vervolgens bestudeerden we de surjectie afstand, faire surjectie afstand en linking afstand. Het berekenen van deze afstanden steunt op het oplossen van een optimalisatieprobleem, dat op het eerste gezicht een hoge computationele kost heeft. We hebben echter aangetoond dat deze optimalisatieproblemen equivalent zijn aan bekende problemen in de grafentheorie. Deze omzettingen maken het mogelijk om deze afstanden te implementeren met een aanvaardbare computationele complexiteit. Vervolgens introduceren we een familie van classifiers die gebruikmaken van similariteitsrelaties om labels van bags te voorspellen: BFMIC. BFMIC staat voor bag-based fuzzy multi-instance classifiers. We zullen dit soort classifier ook gebruiken in de experimentele studie.

De geïntroduceerde concepten werden geïmplementeerd in Python. Deze implementatie werd vervolgens gebruikt om een experimentele studie uit te voeren op 15 datasets. Het doel van deze experimentele studie is om de performantie van de verschillende similariteitsrelaties te testen. Bovendien bekeken we ook het effect van enkele pre-processing stappen. Een van de theoretische resultaten was dat de surjectie afstand en faire surjectie afstand metrieken zijn wanneer het aantal instanties in alle bags gelijk is. Daarom bestudeerden we ook het effect van deze pre-processingmethodes op de prestaties van de verschillende classifiers die deze similariteitsrelaties gebruiken.

We gebruikten een 10-fold cross-validation om de resultaten te genereren. De hyperparameters van elk experiment werden bepaald met behulp van een 5-fold cross-validation op de training dataset. We kozen voor de nauwkeurigheid als performantiemaat voor de gebalanceerde datasets en de $F1$-score voor de niet-gebalanceerde datasets. De verschillende similariteitsrelaties werden vergeleken met behulp van een Friedman test in combinatie met een Conover post-hoc analyse wanneer de Friedman test aangaf dat er significante verschillen waren tussen de similariteitsrelaties.

Elk experiment werd vijf keer herhaald. Eén keer zonder enige pre-processing en vervolgens nog vier keer met verschillende pre-processingmethodes. Dit stelde ons in staat om elk van de pre-processingmethodes te vergelijken met de prestaties zonder enige preprocessing. Deze vergelijking werd gedaan met behulp van de Wilcoxon signed-rank test.

Uit onze experimentele resultaten kunnen we concluderen dat de similariteitsrelaties die gebaseerd zijn op de linking afstand, surjectie afstand en faire surjectie afstand gelijkaardige resultaten oplevert als de similariteitsrelaties die gebaseerd zijn op Hausdorff afstand.

Vooral bij de niet-gebalanceerde datasets zouden we een potentiële prestatiewinst kunnen zien van deze nieuwe afstanden. We moeten echter in gedachten houden dat de computationele complexiteit van de linking afstand, surjectie afstand en faire surjectie afstand aanzienlijk hoger is dan die van de eenvoudige Hausdorff afstand en zijn varianten.

Ten slotte bespreken we de resultaten van de pre-processingmethodes. De effectiviteit van deze methodes was sterk afhankelijk van het soort dataset. Op de gebalanceerde datasets werkten enkele afstanden significant beter wanneer we pre-processing hadden toegepast, en over het algemeen steeg de gemiddelde nauwkeurigheid. Het tegenovergestelde werd waargenomen bij de niet-gebalanceerde datasets. Terwijl geen enkele similariteitsrelatie significant slechter presteerde, daalde de gemiddelde $F1$-score bijna universeel.

B Bibliography

[1] Cornelis, Chris and Verbiest, Nele and Jensen, Richard, "Ordered weighted average based fuzzy rough sets," in *LECTURE NOTES IN COMPUTER SCIENCE* (Yu, Jian and Greco, Salvatore and Lingras, Pawan and Wang, Guoyin and Skowron, Andzrej, ed.), vol. 6401, pp. 78–85, Springer, 2010.

[2] P. Hall, "On representatives of subsets," *Journal of The London Mathematical Society-second Series*, pp. 26–30, 1935.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows - theory, algorithms and applications," 1993.

[4] A. V. Goldberg, É. Tardos, and R. E. Tarjan, "Network flow algorithms," 1989.

[5] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, p. 921–940, oct 1988.

[6] J. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows," *Math. Prog.*, vol. 78, pp. 109–129, 01 1996.

[7] T. R. Ervolina and S. McCormick, "Two strongly polynomial cut cancelling algorithms for minimum cost network flow," *Discrete Applied Mathematics*, vol. 46, no. 2, pp. 133–165, 1993.

[8] T. Eiter and H. Mannila, "Distance measures for point sets and their computation," *Acta Informatica*, vol. 34, 02 1999.

[9] H. N. Gabow, "Data structures for weighted matching and nearest common ancestors with linking," in *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '90, (USA), p. 434–443, Society for Industrial and Applied Mathematics, 1990.

[10] R. Ahuja, J. Orlin, C. Stein, and R. Tarjan, "Improved algorithms for bipartite network flow," *SIAM J. Comput.*, vol. 23, pp. 906–933, 10 1994.

[11] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[12] G. Fu, X. Nan, H. Liu, R. Patel, P. Daga, Y. Chen, D. Wilkins, and R. Doerksen, "Implementation of multiple-instance learning in drug activity prediction," *BMC bioinformatics*, vol. 13 Suppl 15, p. S3, 09 2012.

[13] L. Kejriwal, V. Darbari, and O. P. Verma, "Multi instance multi label classification of restaurant images," *2017 IEEE 7th International Advance Computing Conference (IACC)*, pp. 722–727, 2017.

[14] S. Feng, W. Xiong, B. Li, C. Lang, and X. Huang, "Hierarchical sparse representation based multi-instance semi-supervised learning with application to image categorization," *Signal Process.*, vol. 94, pp. 595–607, 2014.

[15] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, and S. Vluymans, *Multiple Instance Learning: Foundations and Algorithms*. Springer, 2016.

[16] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, no. 1, p. 1–25, 2010.

[17] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81–105, 2013.

[18] S. Vluymans, *Dealing with Imbalanced and Weakly Labelled Data in Machine Learning using Fuzzy and Rough Set Methods*. PhD thesis, Ghent University, 2018.

[19] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. Garc'ia, L. Sanchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, pp. 255–287, 01 2010.

[20] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[21] V. Cheplygina, D. M. Tax, and M. Loog, "Multiple instance learning with bag dissimilarities," *Pattern Recognition*, vol. 48, pp. 264–275, jan 2015.

[22] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002.

[24] Q. Zhou, B. Sun, Y. Song, and S. Li, "K-means clustering based undersampling for lower back pain data," in *Proceedings of the 2020 3rd International Conference on Big Data Technologies*, ICBDT 2020, (New York, NY, USA), p. 53–57, Association for Computing Machinery, 2020.

[25] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.