Universiteit Utrecht

Bachelor Mathematics

Sinkhorn's algorithm for optimal transport

Bachelor Thesis

Loek van Vonderen

First supervisor :

Prof. dr. ir. J.E. Frank :

June 18, 2024

Abstract

The optimal transport distance provides us with a method of assigning a distance between two probability vectors. One downside of this metric is that it can be computationally expensive to compute. One method of estimating the optimal transport distance is by using an entropic regularization, which allows for the use of Sinkhorn's theorem, providing a lower computational load. In this thesis we investigate the convergence of this method and utilise it to study the change of the attractor of the Hénon system. Our results show that the speed of convergence heavily depends on the level of desired accuracy, which is encapsulated by the regularization parameter λ. The results on the attractor show that it is important to use a large sample size of data points to be able to draw a solid conclusion.

Key words: Optimal transport - Sinkhorn's theorem - Numerical methods

Contents

1 Introduction

The optimal transport problem can easily be understood intuitively: how does one minimise the cost of moving resources to their destinations. The optimal transport distance has many applications in for example machine-learning [1], imaging [2] and data analysis for biology [3]. While it has many practical applications it comes with the downside of being computationally expensive for larger datasets.

In this thesis we investigate a numerical method of estimating the optimal transport distance by adding an entropic regularization to the minimisation problem as described by Ref. [4]. Adding this regularization changes the traditional problem from a linear programming problem into a matrix-scaling problem, by making use of *Sinkhorn's theorem* [5]. This has the benefit of being easier to compute, but comes at the cost of resulting in a slightly suboptimal solution, depending on the regularization parameter λ. The standard linear programming method has a complexity of $O(n^3 \log(n))$, whereas the solution described in this thesis has a complexity of $O(n^2)$ [4].

We will be using this regularization method of calculating the optimal transport distance to study the behaviour of the chaotic Hénon system. This is a dynamical system that exhibits either chaotic or periodic behaviour depending on two parameters. The shape of the attractor changes as a result of the parameters and we will be looking at the distances between these attractors.

In Chapter 2 we recap the basics of optimal transport, look at Sinkhorn's theorem and introduce the entropic regularization. Next we explain and analyse the convergence of our methodology in Chapter 3. In Chapter 4 we present and discuss the results. Finally we provide a brief explanation of the assumptions and limitations, as well as some closing words, in Chapter 5.

2 Theory

2.1 The optimal transport problem

In this section we will recap the concept of optimal transport, also known as the earth mover's distance. We will first give an intuitive description as background, after which we state the formal definition.

2.1.1 Earth mover's distance

The optimal transport problem brings forth a distance, which can be intuitively understood as the *Earth mover's distance*. Suppose that we have been given a pile of sand with a certain shape, and that we want to move each grain of sand such that our pile forms a new, predefined shape. Of course, there are many ways in which we could move each individual grain, as any one can end up at any point of the desired new shape. However, when we assign some sort of cost to moving a grain, then it is not unreasonable to want to minimise the total cost of moving the entire pile, obtained by summing over the costs of moving each individual grain. This minimum possible cost defines a distance between the two piles, the Earth mover's distance, and the problem of finding it is called the *optimal transport problem*.

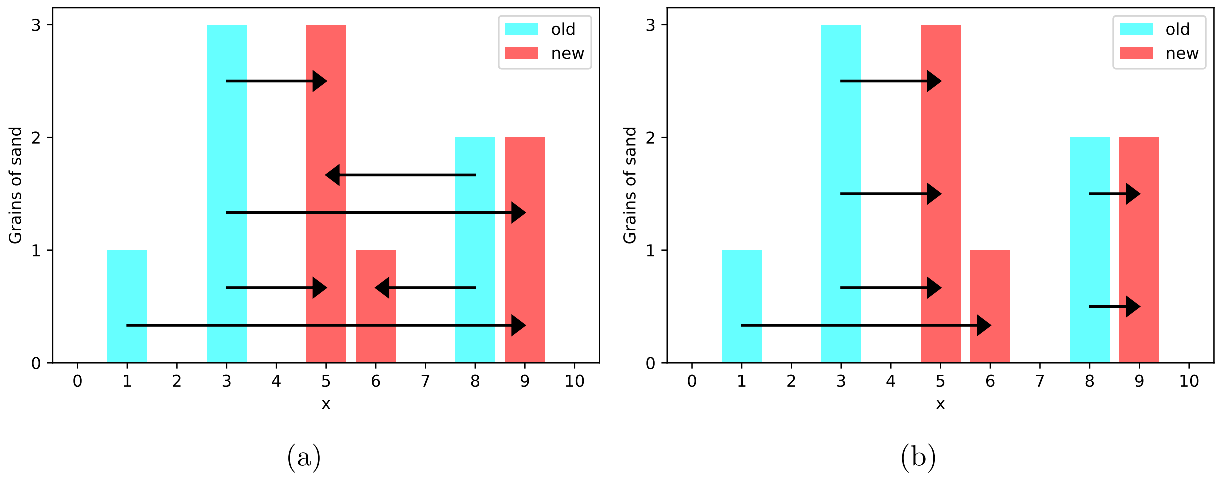


Figure 1: Two different ways of moving the grains from the old configuration into the new one. The transport plan of figure 1a has total cost 23 and the plan of figure 1b has a total cost of 13, this happens to be the minimum cost of any transport plan between these two piles, so that makes their earth mover's distance 13.

In figure 1 we compare two different plans for moving a pile of sand from one configuration into another. For the cost c of moving a single grain from x_1 to x_2, we have used $c(x_1, x_2) = d_{eucl}(x_1, x_2) = |x_1 - x_2|$, which is the Euclidean distance, but we could have used any other distance function d, called the *ground metric*. Now that we have an intuitive understanding of optimal transport we can state the formal definitions.

2.1.2 Optimal transport distance

Instead of comparing the distance between piles of sand we will instead be comparing *probability vectors.*

Definition 2.1 (Probability vector)**.** *An n-dimensional probability vector $\mathbf{x}$ is a vector of $\mathbb{R}^n$ such that all its components are positive and add up to* 1*, i.e.*

$$\mathbf{x} \in \mathbb{R}^n_+ \,, \quad \textit{such that} \quad \sum_i x_i = 1\,.^1 \tag{1}$$

The main difference between optimal transport of piles of sand and probability vectors is that with sand we have to discretely move the grains from one location to another, but with probability vectors we can split its entries and divide it over multiple destinations. The objects that describe the transport between an n-dimensional probability vector $\mathbf{r}$ and an m-dimensional probability vector $\mathbf{c}$ are called *transport matrices.* The entries P_{ij} of $\mathbf{P} \in \mathbb{R}^{n\times m}$ describe the amount of mass moving from r_i to c_j. The set of all transport matrices is the *transport polytope.*

Definition 2.2 (Transport polytope)**.** *The transport polytope describes all possible transport matrices between two probability vectors,*

$$U(\mathbf{r},\mathbf{c}) := \Big\{\mathbf{P} \in \mathbb{R}^{n\times m}_+ \mid \sum_j P_{ij} = r_i\,, \forall j\,, \quad \sum_i P_{ij} = c_j\,, \forall i\Big\} \tag{2}$$

Lemma 2.3. *The set $U(\mathbf{r},\mathbf{c})$ is compact for any $\mathbf{r}$ and $\mathbf{c}$.*

Proof. We start by showing that $U(\mathbf{r},\mathbf{c})$ is closed. Define the following function

$$f: \mathbb{R}^{n\times m} \to \mathbb{R}^n \times \mathbb{R}^m, \quad f(\mathbf{P}) := \left(\sum_j P_{1j}, \cdots, \sum_j P_{nj}, \sum_i P_{i1}, \cdots, \sum_i P_{mi}\right), \tag{3}$$

which is clearly continuous in the components of $\mathbf{P}$, hence the preimage of the closed set $\{(\mathbf{r},\mathbf{c})\}$ is also closed. By the fact that $\mathbb{R}^{n\times m}_+$ is closed we find that

$$f^{-1}\big(\{(\mathbf{r},\mathbf{c})\}\big) \cap \mathbb{R}^{n\times m}_+ = \Big\{\mathbf{P} \in \mathbb{R}^{n\times m}_+ \mid \sum_j P_{ij} = r_i\,, \forall j\,, \quad \sum_i P_{ij} = c_j\,, \forall i\Big\} = U(\mathbf{r},\mathbf{c}) \tag{4}$$

is closed.

For $\mathbf{P} \in U(\mathbf{r},\mathbf{c})$ we clearly have that

$$0 \le P_{ij} \le r_i < M, \quad \forall i,j\,, \tag{5}$$

for some $M \in \mathbb{R}$ and thus $U(\mathbf{r},\mathbf{c})$ is bounded. □

[1] Throughout this thesis we use the convention $0 \in \mathbb{R}_+$.

For now we suppose that $m = n$ and that we have been given a *distance matrix* $\mathbf{M} \in \mathbb{R}_+^{n\times n}$, this is a matrix that satisfies the following

$$\begin{aligned} (i)\ &\mathbf{M} \text{ is symmetric,} \\ (ii)\ &\mathbf{M}_{ij} = 0 \text{ if and only if } i = j, \\ (iii)\ &\mathbf{M}_{ik} \leq \mathbf{M}_{ij} + \mathbf{M}_{jk}\,, \text{ for all i,j,k,} \end{aligned} \tag{6}$$

where

$$M_{ij} = c(r_i, c_j)\,. \tag{7}$$

This lets us quantify the cost of moving $\mathbf{r}$ to $\mathbf{c}$ using the transport matrix $\mathbf{P}$ as the Frobenius inner product between $\mathbf{P}$ and $\mathbf{M}$, i.e.

$$\langle P, M\rangle_F := \sum_{i,j} P_{ij} M_{ij}\,. \tag{8}$$

This then allows us to define the *optimal transport distance.*

Definition 2.4 (Optimal transport distance)**.** *The optimal transport distance between two probability vectors* $\mathbf{r}$ *and* $\mathbf{c}$ *is given by*

$$d_M(\mathbf{r}, \mathbf{c}) := \min_{\mathbf{P}\in U(\mathbf{r},\mathbf{c})} \langle \mathbf{P}, \mathbf{M}\rangle_F\,. \tag{9}$$

Finding this minimum is called an optimal transport problem. Note that this minimum is well-defined by lemma 2.3 and the continuity of the Frobenius norm.

2.2 Sinkhorn's theorem

Definition 2.5 (Bistochastic matrix)**.** *A square matrix A is called bistochastic iff all rows and all columns sum to* 1*, i.e.*

$$\sum_i A_{ij} = 1\,, \quad \forall j \quad \textit{and} \quad \sum_j A_{ij} = 1\,, \quad \forall i\,. \tag{10}$$

Definition 2.6 (Convex combination)**.** *A convex combination X of a finite set of points $x_1, x_2, \cdots, x_N \in \mathbb{R}$ is a linear combination of said points, such that the coefficients are non-negative and sum to* 1*, i.e.*

$$X = \alpha_1 x_1 + \alpha_2 x_2 \cdots + \alpha_N x_,, \quad \textit{such that} \quad \sum_i \alpha_i = 1 \quad \textit{and} \quad \alpha_i \geq 0\,. \tag{11}$$

Lemma 2.7. [2] *Let X be a convex combination of $x_1, x_2, \cdots, x_N \in \mathbb{R}$, then*

$$\min_i x_i \leq X \leq \max_i x_i\,. \tag{12}$$

The following theorem plays a central role in this thesis, so we will review the proof given by Ref. [5].

[2] The proof has been omitted, as it is self-evident.

Theorem 2.8 (Sinkhorn's theorem)**.** *Let A be an $N \times N$ matrix, with $A_{ij} > 0$ for all i, j. There exist non-negative diagonal matrices D_1 and D_2, such that $D_1 A D_2$ is bistochastic. These matrices are uniquely defined up to a multiplicative factor.*

Proof. (This proof is adapted from Ref. [5]) We start by proving the uniqueness of D_1 and D_2. Suppose that there exist two pairs of non-negative diagonal matrices D_1, D_2 and C_1, C_2 such that both $P := D_1 A D_2$ and $Q := C_1 A C_2$ are bistochastic. We show that $D_1 = \lambda C_1$ and $D_2 = \frac{1}{\lambda} C_2$.

Let $D_1 = \operatorname{diag}(d_{11}, \cdots, d_{1N})$, with similar definitions for D_2, C_1 and C_2. Define $B_i = \operatorname{diag}\left(\frac{c_{i1}}{d_{i1}}, \cdots, \frac{c_{i1}}{d_{i1}}\right) = \operatorname{diag}(b_{i1}, \cdots, b_{iN})$ for $i = 1, 2$, then $B_1 P B_2 = Q$. Note that

$$Q_{ij} = (B_1 P B_2)_{ij} = b_{1i} p_{ij} b_{2j}, \tag{13}$$

so by the fact that Q is bistochastic we have that the sum of the j-th column is

$$\sum_i b_{1i} p_{ij} b_{2j} = 1 \quad \Rightarrow \quad \sum_i b_{1i} p_{ij} = 1/b_{2j}\,. \tag{14}$$

By the fact that P is bistochastic we find that $1/b_{2j}$ is a convex combination of the elements of B_1, so by lemma 2.7 we find

$$\min_i b_{1i} \leq 1/b_{2j} \leq \max_i b_{1i}. \tag{15}$$

A similar argument for the row-sums yields

$$\min_j b_{2j} \leq 1/b_{1i} \leq \max_j b_{2j}. \tag{16}$$

Now suppose that there exist i_0, j_0, such that $b_{1i_0} b_{2j_0} < 1$, then certainly $b_{1i_0} \min_j b_{2j} < 1$. Let $j_1 = \operatorname*{argmin}_j b_{2j}$, then by equation 14 we find the following inequality

$$1 = \sum_i b_{1i} p_{ij_1} b_{2j_1} < p_{i_0 j_1} + \sum_{\substack{i=1 \\ i \neq i_0}}^{n} b_{1i} p_{ij_1} b_{2j_1}\,, \tag{17}$$

and so

$$\sum_{\substack{i=1 \\ i \neq i_0}}^{n} b_{1i} p_{ij_1} b_{2j_1} > 1 - p_{i_0 j_1}\,. \tag{18}$$

From the fact that $\sum_i p_{ij_1} = 1$ we conclude that there must exist an i_1 such that $b_{1i_1} b_{2j_1} > 1$, i.e. $\min_j b_{2j} = b_{2j_1} > 1/b_{1i_1}$, which is in contradiction with equation (16). A similar argument holds when assuming there exist i_0, j_0 such that $b_{1i_0} b_{2j_0} > 1$, so we find that $b_{1i} b_{2j} = 1$ for all i, j, that is, $b_{2i} = 1/b_{1j} = \lambda$ for all i, j. Since $b_{1i} := \frac{c_{1i}}{d_{1i}}$ we have

$$\lambda c_{1i} = d_{1i}\,, \quad \forall\, i\,. \tag{19}$$

We conclude that $\mathbf{D}_1 = \lambda \mathbf{C}_1$ and similarly $\mathbf{D}_2 = \frac{1}{\lambda}\mathbf{C}_2$, which was to be shown.

We now show that D_1 and D_2 exist. Let

$$A_0 := A,\ A_{n+1} := \lambda_n A_n \delta_n,\ \lambda_n := \operatorname{diag}(1/\lambda_{n1}, \cdots, 1/\lambda_{nN}) \text{ and } \delta_n := \operatorname{diag}(1/\delta_{n1}, \cdots, 1/\delta_{nN}) . \tag{20}$$

Let λ_{ni} be the sum of the i-th row of A_n, such that left-hand multiplication of A_n by λ_n normalises the rows.[3] Let δ_{nj} be the sum of the j-th column of $\lambda_n A_n$, i.e.

$$\delta_{nj} := \sum_i A_{nij}/\lambda_{ni} , \tag{21}$$

such that right-hand multiplication of $\lambda_n A_n$ by δ_n normalises the columns. Let a_n denote the minimal element of A_n.[4]

We will prove that the sequence $\{A_n\}$ converges to a doubly stochastic matrix, to do this we must show two things. Firstly we need that $A_{nij} \geq c > 0$ for all n, i, j and some constant c, secondly we require that λ_{ni} approaches 1 arbitrarily close for all i. From now we assume that $n \geq 1$ such that A_n has normalised columns. Finally define $\underline{\lambda}_n = \min_i \lambda_{ni}$ and $\overline{\lambda}_n = \max_i \lambda_{ni}$.

Lemma 2.9. *For this setup we have that*

$$\underline{\lambda}_n \leq 1 \leq \overline{\lambda}_n , \quad \forall n . \tag{22}$$

Proof. Notice that the sum of all row-sums and of all column-sums of A_n should be equal, and since all column sums are 1 we find that this should amount to N. If $\lambda_{ni} = 1$ for all i than we have found a bistochastic matrix and we can prove the main theorem, so let's say that $\lambda_{ni_0} < 1$. If all $\lambda_{ni} \leq 1$, then the sum of all row-sums is

$$\sum_i \lambda_{ni} = \lambda_{ni_0} + \sum_{i \neq i_0} \lambda_{ni} < 1 + \sum_{i \neq i_0} \lambda_{ni} \leq N . \tag{23}$$

This is a contradiction, so we find that there must be at least one $\lambda_{ni} > 1$. Similar arguments can be used when assuming $\lambda_{ni_0} > 1$, this proves the lemma. $\square$

From the definition of δ_{nj} in equation 21, the fact that the columns of A_n are normalised and lemma 2.7 we find that

$$\min_i 1/\lambda_{ni} \leq \delta_{nj} \leq \max_i 1/\lambda_{ni} \quad \Rightarrow \quad 1/\overline{\lambda}_n \leq \delta_{nj} \leq 1/\underline{\lambda}_n , \quad \forall j . \tag{24}$$

Then from the fact that $\lambda_{n+1,i}$ is a convex combination of $1/\delta_{nj}$ we get

$$\min_j 1/\delta_{nj} \leq \lambda_{n+1,i} \leq \max_j 1/\delta_{nj} \quad \Rightarrow \quad \underline{\lambda}_n \leq \lambda_{n+1,i} \leq \overline{\lambda}_n , \quad \forall i . \tag{25}$$

If we then combine this with our result from lemma 2.9 then we find

$$\underline{\lambda}_n \leq \underline{\lambda}_{n+1} \leq 1 \leq \overline{\lambda}_{n+1} \leq \overline{\lambda}_n . \tag{26}$$

[3]Note that $\lambda_1 > 0$

[4]Note that $a_1 > 0$.

This gives that the minimal and maximal elements of the row-sums are monotone sequences that thus approach some limit, it remains to show that these two limits are all 1. Define the following quantities

$$X_n = \prod_{k=1}^{n} \lambda_k,\; X_{ni} = \left(\prod_{k=1}^{n} \lambda_{ki}\right)^{-1}, \quad Y_n = \prod_{k=1}^{n} \delta_k,\; Y_{nj} = \left(\prod_{k=1}^{n} \delta_{kj}\right)^{-1}. \tag{27}$$

From these definitions we obtain the following inequalities,

$$Y_{nj} = \left(\sum_i A_{1ij} X_{ni}\right)^{-1} \leq (A_{1ij}X_{ni})^{-1} \leq (a_1 X_{ni})^{-1}, \quad \forall i, j\,. \tag{28}$$

$$\sum_j X_{ni} A_{ij} Y_{nj} = \lambda_{n+1,i} \geq \underline{\lambda}_{n+1} \geq \underline{\lambda}_1 \quad \Rightarrow \quad X_{ni} \geq \underline{\lambda}_1 \left(\sum_j A_{ij} Y_{nj}\right)^{-1} \geq a_1 \underline{\lambda}_1 \max_i X_{ni}/N\,. \tag{29}$$

Lemma 2.7 and the fact that the columns of A_1 sum up to 1 gives

$$Y_{nj} = 1/\sum_i A_{ij} X_{ni} \geq 1/\max_i X_{ni}\,. \tag{30}$$

If we now combine equation (29) and equation (30), then we find for all n, i, j the desired property,

$$A_{n+1,ij} = X_{ni} A_{nij} Y_{nj} \geq a_1 \underline{\lambda_1}/N > 0\,. \tag{31}$$

From equation (26) it follows that $\overline{\lambda}_n \to 1 + c$ for some $c \geq 0$. Let $\overline{\lambda}_n = 1 + c_n$, such that $c_n \to c$, and define

$$\underline{\mu}_j = \sum_{\substack{i \\ \lambda_{ni} \leq 1}} A_{nij}\,, \quad \overline{\mu}_j = \sum_{\substack{i \\ \lambda_{ni} > 1}} A_{nij}\,, \tag{32}$$

then

$$\delta_{nj} \geq \underline{\mu}_j + \frac{1}{1+c_n}\overline{\mu}_j = \frac{\underline{\mu}_j + \overline{\mu}_j + c_n \underline{\mu}_j}{1+c_n} \geq \frac{1 + c_n a_n}{1+c_n}\,. \tag{33}$$

Finally let $i_0 = \underset{i}{\operatorname{argmin}}\, \lambda_{ni}$, then

$$1 + c \leq \lambda_{n+1,i_0} = \sum_j A_{ni_0j}/\lambda_{ni_0}\delta_{nj} \leq 1\delta_{nj} \leq \frac{1+c_n}{1+c_n a_n}\,. \tag{34}$$

However, because $a_n > 0$ and $c_n \to c$, we have that $c > 0$ leads to a contradiction, so $c = 0$. In similar fashion it follows that $\underline{\lambda}_n \to 1$. We conclude that $A_n = X_n A Y_n$ approaches a doubly stochastic matrix, such that $X_n \to D_1$ and $Y_n \to D_2$. □

Corollary 2.10. *Let* $\mathbf{A} \in \mathbb{R}_{>0}^{n\times m}$*, and* $\mathbf{r}$ *and* $\mathbf{c}$ *be n- and m-dimensional probability vectors. There exist non-negative diagonal matrices* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *such that* $\mathbf{D}_1\mathbf{A}\mathbf{D}_2$ *is unique and has row-sums* $\mathbf{r}$ *and column-sums* $\mathbf{c}$*, i.e.*

$$\sum_j (\mathbf{D}_1\mathbf{A}\mathbf{D}_2)_{ij} = r_i \quad \forall i \tag{35}$$

$$\sum_i (\mathbf{D}_1\mathbf{A}\mathbf{D}_2)_{ij} = c_j \quad \forall j\,. \tag{36}$$

Remark 2.11. *The proof of corollary 2.10 is similar to that of 2.8 and is therefore omitted. More detail can be found in Ref. [6].*

Algorithm 2.12. *In order to find* $\mathbf{D}_1\mathbf{A}\mathbf{D}_2$ *as described above, one can preform the following algorithm:*

1. *If there exist one or multiple* i_0 *such that* $r_{i_0} = 0$ *then set* $A_{i_0 j} = 0$ *for all* j*. Similarly for* $c_{j_0} = 0$ *set* $A_{ij_0} = 0$ *for all* i*.*
2. *Multiply each row* $i \neq i_0$ *of* $\mathbf{A}$ *by* $r_i / \sum_j A_{ij}$ *to obtain* $\mathbf{A}'$ *with all row sums equal to* $\mathbf{r}$*.*
3. *Multiply each column* $j \neq j_0$ *of* $\mathbf{A}'$ *by* $c_j / \sum_i A'_{ij}$ *to make all column sums equal to* $\mathbf{c}$*.*
4. *Repeat step 2. and 3. until the differences between the row-sums and* $\mathbf{r}$ *are smaller than needed.*

In order to calculate these steps we will be executing a slightly modified version of this algorithm with python, the details can be found in Section 3.2.

2.3 Entropic regularization

We have seen that solving the optimal transport problem 9 can be computationally expensive, so in order to solve this we introduce an *entropic regularization*. For this purpose we define the *entropy* of a probability vector $\mathbf{r}$ and a transport matrix $\mathbf{P}$,

$$h(\mathbf{r}) := -\sum_i r_i \log r_i\,, \quad h(\mathbf{P}) := -\sum_{i,j} P_{ij} \log P_{ij}\,, \tag{37}$$

with the substantiated[5] convention

$$x \log x = 0 \quad \text{if} \quad x = 0\,. \tag{38}$$

Note that a more equally distributed probability vector or transport matrix has a higher entropy than one that is more localised, therefore entropy can be seen as a measure of *smoothness*. Also note that $\mathbf{r}\mathbf{c}^T$ has $h(\mathbf{r}\mathbf{c}^T) = h(\mathbf{r}) + h(\mathbf{c})$ and that it is the smoothest possible transport matrix [4]. Because problem 9 is a complex problem we can make an approximation by reducing the space of the allowed transport matrices to

$$U_\alpha(\mathbf{r},\mathbf{c}) := \left\{\mathbf{P} \in U(\mathbf{r},\mathbf{c}) \,|\, h(\mathbf{P}) \geq h(\mathbf{r}\mathbf{c}^T) - \alpha\right\}\,. \tag{39}$$

This is essentially the set of transport matrices which are *sufficiently smooth.*

Lemma 2.13. *The set* $U_\alpha(\mathbf{r},\mathbf{c})$ *is compact for any* $\mathbf{r}$*,* $\mathbf{c}$ *and* α*.*

Proof. Recall from lemma 2.3 that $U(\mathbf{r},\mathbf{c})$ is compact. We show that $U_\alpha(\mathbf{r},\mathbf{c})$ is closed and bounded.

Because $U_\alpha(\mathbf{r},\mathbf{c}) \subset U(\mathbf{r},\mathbf{c})$ it is certainly bounded.

[5]This comes from $\lim_{x\to 0} x \log x = 0\,.$

The function for the entropy in equation 37 is clearly continuous so the preimage of the closed set $[h(\mathbf{rc}^T)-\alpha,\infty)$ is also closed, hence

$$h^{-1}\big([h(\mathbf{rc}^T)-\alpha,\infty)\big)\cap U(\mathbf{r},\mathbf{c})=\big\{\mathbf{P}\in U(\mathbf{r},\mathbf{c})\,|\,h(\mathbf{P})\geq h(\mathbf{rc}^T)-\alpha\big\}=U_\alpha(\mathbf{r},\mathbf{c}) \tag{40}$$

is closed. □

Definition 2.14 (Sinkhorn distance)**.** *The Sinkhorn distance is given by*

$$d_{M,\alpha}(\mathbf{r},\mathbf{c}):=\begin{cases}0\,, & \textit{if}\;\mathbf{r}=\mathbf{c}\\ \min_{\mathbf{P}\in U_\alpha(\mathbf{r},\mathbf{c})}\langle\mathbf{P},\mathbf{M}\rangle_F\,, & \textit{else}\end{cases}\;. \tag{41}$$

We will see why this distance is more easily computed than $d_M(\mathbf{r},\mathbf{c})$ in Section 2.4. Note that this minimum is well-defined by lemma 2.13 and the continuity of the Frobenius norm.

Remark 2.15. *Note that for $\mathbf{r}=\mathbf{c}$ we obviously have $\mathbf{P}=diag(\mathbf{r})$ and $\langle diag(\mathbf{r}),\mathbf{M}\rangle_F=0$. It is however not guaranteed that $diag(\mathbf{r})\in U_\alpha(\mathbf{r},\mathbf{c})$ for all α. For this reason we cannot simply say,*

$$d_{M,\alpha}(\mathbf{r},\mathbf{c})=\min_{\mathbf{P}\in U_\alpha(\mathbf{r},\mathbf{c})}\;. \tag{42}$$

2.3.1 Properties of the Sinkhorn distance

Since we have crowned Sinkhorn distances with the name *distance*, it is desirable to show that it does indeed satisfy the axioms of a metric.

Theorem 2.16. *The Sinkhorn distance $d_{M,\alpha}$ as defined in definition 2.14 satisfies the axioms for a metric for all $\alpha\in\mathbb{R}$ and all distance matrices $\mathbf{M}$, i.e.*

(i) $d_{M,\alpha}(\mathbf{r},\mathbf{c})=0 \quad\Leftrightarrow\quad \mathbf{r}=\mathbf{c}$

(ii) $d_{M,\alpha}(\mathbf{r},\mathbf{c})=d_{M,\alpha}(\mathbf{c},\mathbf{r})$

(iii) $d_{M,\alpha}(\mathbf{r},\mathbf{c})\leq d_{M,\alpha}(\mathbf{r},\mathbf{v})+d_{M,\alpha}(\mathbf{v},\mathbf{c})\,,$ *for all probability vectors* $\mathbf{c},\mathbf{r},\mathbf{v}$

Proof. This proof is inspired by Ref. [4].

(i) The first property is a trivial consequence of the definition of $d_{M,\alpha}$. Properties *(ii)* and *(iii)* are also trivial under the assumption $\mathbf{r}=\mathbf{c}$ and so their proofs are omitted. For the remainder of the proof we assume $\mathbf{r}\neq\mathbf{c}$.

(ii) Next we prove symmetry. Note that $\mathbf{M}$ is a distance matrix, where M_{ij} denotes the distance between the i-th bin of $\mathbf{r}$ and the j-th bin of $\mathbf{c}$. Since $\mathbf{M}$ is symmetric by the second property of properties (6), it also describes the distance between the i-th bin of $\mathbf{c}$ and the j-th bin of $\mathbf{r}$. Note that if $\mathbf{P}\in U(\mathbf{r},\mathbf{c})$, then $\mathbf{P}^T\in U(\mathbf{c},\mathbf{r})$, the same goes for $U_\alpha(\mathbf{r},\mathbf{c})$ and $U_\alpha(\mathbf{c},\mathbf{r})$. This gives us

$$d_{M,\alpha}(\mathbf{r},\mathbf{c})=\min_{\mathbf{P}\in U_\alpha(\mathbf{r},\mathbf{c})}\langle\mathbf{P},\mathbf{M}\rangle_F=\min_{\mathbf{P}\in U_\alpha(\mathbf{c},\mathbf{r})}\langle\mathbf{P}^T,\mathbf{M}\rangle_F=\min_{\mathbf{P}\in U_\alpha(\mathbf{c},\mathbf{r})}\langle\mathbf{P},\mathbf{M}\rangle_F=d_{M,\alpha}(\mathbf{c},\mathbf{r})\,. \tag{43}$$

(iii) Finally we want to show that the triangle inequality holds. Let $\mathbf{c},\mathbf{r},\mathbf{v}$ be three d-dimensional probability vectors. Let $P\in U_\alpha(\mathbf{r},\mathbf{v})$ and $Q\in U_\alpha(\mathbf{v},\mathbf{c})$. Define $\mathbf{S}$ as the $d\times d$ transport matrix with $S_{ik}:=\sum_j\frac{p_{ij}q_{jk}}{v_j}$. In order to prove *(iii)* we will need the following lemma.

Lemma 2.17 (Gluing lemma). [6] *Let* $\mathbf{S}$, $\mathbf{r}$ *and* $\mathbf{c}$ *be as defined above, then* $\mathbf{S} \in U_\alpha(\mathbf{r}, \mathbf{c})$.

Assume now that $\mathbf{P}$ and $\mathbf{Q}$ are optimal solutions of their respective optimal transport problems, then by using property 6 (*iii*) and the gluing lemma 2.17,

$$d_{M,\alpha}(\mathbf{r}, \mathbf{c}) = \min_{\mathbf{P} \in U_\alpha(\mathbf{r},\mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle_F \leq \langle \mathbf{S}, \mathbf{M} \rangle_F = \sum_{i,k} M_{ik} \sum_j \frac{p_{ij} q_{jk}}{v_j} \leq \sum_{i,j,k} (M_{ij} + M_{jk}) \frac{p_{ij} q_{jk}}{v_j} \tag{44}$$

$$= \sum_{i,j,k} M_{ij} \frac{p_{ij} q_{jk}}{v_j} + M_{jk} \frac{p_{ij} q_{jk}}{v_j} \leq \sum_{i,j} M_{ij} p_{ij} \sum_k \frac{q_{jk}}{y_j} + \sum_{j,k} M_{jk} q_{jk} \sum_i \frac{p_{ij}}{v_j} \tag{45}$$

$$= \sum_{i,j} M_{ij} p_{ij} + \sum_{j,k} M_{jk} q_{jk} = d_{M,\alpha}(\mathbf{r}, \mathbf{v}) + d_{M,\alpha}(\mathbf{v}, \mathbf{c})\,. \tag{46}$$

□

2.4 Computing Sinkhorn distance

We will now look at the effects of adding an entropic regularization. Take a look at the following,

$$\text{for } \lambda > 0,\ d_M^\lambda(\mathbf{r}, \mathbf{c}) := \langle \mathbf{P}^\lambda, \mathbf{M} \rangle_F\,, \quad \text{where } \mathbf{P}^\lambda = \operatorname*{argmin}_{\mathbf{P} \in U(\mathbf{r},\mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle_F - \frac{1}{\lambda} h(\mathbf{P})\,.^7 \tag{47}$$

But how does this relate to our original problem? When we take $\lambda \to \infty$, then

$$\mathbf{P}^\lambda \to \operatorname*{argmin}_{\mathbf{P} \in U(\mathbf{r},\mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle_F, \text{ and so } d_M^\lambda \to d_{M,\infty} = d_M\,. \tag{48}$$

If we take $\lambda \to 0$ on the other hand, we get

$$\mathbf{P}^\lambda \to \operatorname*{argmin}_{\mathbf{P} \in U(\mathbf{r},\mathbf{c})} \ -h(\mathbf{P}) = \operatorname*{argmax}_{\mathbf{P} \in U(\mathbf{r},\mathbf{c})} h(\mathbf{P}) = \mathbf{r}\mathbf{c}^T, \text{ and so } d_M^\lambda \to d_{M,0}\,. \tag{49}$$

What this means is that for every α in between we have that there exists a λ such that $d_{M,\alpha} = d_M^\lambda$, see Ref. [7]. By adding the constraints to our original problem (9) and introducing dual variables $\beta, \gamma \in \mathbb{R}^d$ we obtain what is known as the Lagrangian:

$$L(P, \beta, \gamma) = \sum_{i,j} p_{ij} m_{ij} + \beta^T (\mathbf{P} \mathbf{1}_d - \mathbf{r}) + \gamma^T (\mathbf{P}^T \mathbf{1}_d - \mathbf{c})\,, \quad p_{ij} \geq 0,\ \forall i, j\,. \tag{50}$$

This allows us to solve problem (9) by means of $\frac{\partial L}{\partial p_{ij}} = 0$, for more detail see Ref. [7]. In doing so we get to the following linear problem

$$m_{ij} + \beta_i + \gamma_j = 0\,, \quad p_{ij} \geq 0\,, \quad \sum_j p_{ij} = r_i\,, \quad \sum_i p_{ij} = c_j\,, \quad \forall i, j\,, \tag{51}$$

[6] Unfortunately the proof of this lemma is beyond the scope of this thesis, but for more information see Ref. [4].

[7] This minimum is well-defined by lemma 2.13 and the continuity of the Frobenius norm and the entropy.

consisting of $2nm + n + m$ constraints, and the following unknown variables,

$$p_{ij}\,, m_{ij}\,, \beta_i\,, \gamma_i\,, \quad \text{for } 1 \leq i \leq n\,,\ 1 \leq j \leq m\,, \tag{52}$$

such that the number of variables are equal to the number of constraints. This problem can be solved using linear programming methods, and in the worst case scenario of $n = m$ has a complexity of $O(n^3 \log(n))$, see Ref. [4]. Conversely, the Lagrangian of problem (47) is given by

$$L(P, \beta, \gamma) = \sum_{i,j} \frac{1}{\lambda} p_{ij} \log p_{ij} + p_{ij} m_{ij} + \beta^T(\mathbf{P}\mathbf{1}_d - \mathbf{r}) + \gamma^T(\mathbf{P}^T\mathbf{1}_d - \mathbf{c})\,. \tag{53}$$

This implicitly requires $p_{ij} \geq 0$ for all i, j. Furthermore if we again set all partial derivatives to 0, then this time we get

$$\frac{1}{\lambda} \log p_{ij} + \frac{1}{\lambda} + m_{ij} + \beta_i + \gamma_j = 0 \quad \Rightarrow \quad p_{ij} = e^{-\frac{1}{2} - \lambda \beta_i} e^{-\lambda m_{ij}} e^{-\frac{1}{2} - \lambda \gamma_j} > 0\,. \tag{54}$$

Some simple matrix multiplication allows us to rewrite this as

$$\mathbf{P}^\lambda = \operatorname{diag}\bigl(e^{-1/2 - \lambda\beta}\bigr)\, e^{-\lambda \mathbf{M}}\, \operatorname{diag}\bigl(e^{-1/2 - \lambda\gamma}\bigr)\,, \tag{55}$$

where the exponential denotes component-wise exponentiation. Note that $e^{-\lambda M_{ij}} > 0$, for all i, j. By corollary 2.10 it follows that there exist matrices D_1 and D_2 such that $D_1 e^{-\lambda \mathbf{M}} D_2$ is unique and has row-sums $\mathbf{r}$ and column-sums $\mathbf{c}$. Since $\mathbf{P}^\lambda$ has these exact row- and column sums it follows that $D_1 = \operatorname{diag}\bigl(e^{-1/2-\lambda\beta}\bigr)$, and $D_2 = \operatorname{diag}\bigl(e^{-1/2-\lambda\gamma}\bigr)$, and so in order to find $\mathbf{P}^\lambda$ we can simply perform algorithm 2.12 on the matrix $e^{-\lambda \mathbf{M}}$, from here it is a simple multiplication to find $d_M^\lambda(\mathbf{p}, \mathbf{q})$ with equation (47). For $n = m$ this algorithm has a complexity of $O(n^2)$, which is better than the $O(n^3 \log(n))$ of the linear programming solution, see Ref. [4].

3 Methodology

We denote with $[\![a, b]\!]$ the set $\{n \in \mathbb{N} \mid a \leq n \leq b\}$, or simply $[\![b]\!]$ if $a = 1$.

3.1 Sinkhorn distances between points

Let X be a *set of n points* and Y a set of m points of a metric space Z, with normalised weights, i.e.

$$X := \{(\mathbf{x}_i, w_i) \in Z \times (0,1) \mid i \in [\![n]\!], \sum_i w_i = 1\}, \tag{56}$$

such that $\mathbf{x_i} \neq \mathbf{x_j}$ for $i \neq j$, and with a similar definition for Y. Define the $(n+m) \times (n+m)$ *distance matrix*[8] $\mathbf{M}$ for X and Y with the following entries

$$M_{ij} = d(\mathbf{z}_i, \mathbf{z}_j)\,, \tag{57}$$

where d is the *ground metric*, and

$$\mathbf{z}_i = \begin{cases} \mathbf{x}_i & \text{if } i \in [\![n]\!] \\ \mathbf{y}_{i-n} & \text{if } i \in [\![n+1, n+m]\!] \end{cases}. \tag{58}$$

Now define the following two *probability vectors*,

$$\mathbf{p} \in R^{n+m}, \quad \text{where } p_i = w_i \text{ for } i \in [\![n]\!] \text{ and } p_i = 0 \text{ for } i \in [\![n+1, n+m]\!]\,, \tag{59}$$

$$\mathbf{q} \in R^{n+m}, \quad \text{where } q_j = 0 \text{ for } j \in [\![n]\!] \text{ and } q_j = v_{j-n} \text{ for } j \in [\![n+1, n+m]\!]\,, \tag{60}$$

where v_j are the weights of Y. These definitions allow us to use the Sinkhorn distance to define a distance between two sets of points, like X and Y, i.e.

Definition 3.1 (Sinkhorn distance (of sets))**.** *Let X and Y be two sets of points of $\mathbb{R}^2$ with normalised weights, then their Sinkhorn distance is given by*

$$d_\lambda(X, Y) := d_M^\lambda(\mathbf{p}, \mathbf{q}) \xrightarrow{\lambda \to \infty} d_M(\mathbf{p}, \mathbf{q})\,, \tag{61}$$

where $\mathbf{p}, \mathbf{q}$, and $\mathbf{M}$ are as defined above.

We have seen a method of computing $d_M^\lambda(\mathbf{r}, \mathbf{c})$ for arbitrary $\mathbf{r}, \mathbf{c}$ and $\mathbf{M}$ in Section 2.4, of which we will now show a small example.

Example 3.2. *Let $X = \{(0, 0, \frac{1}{2}), (1, 0, \frac{1}{2})\}$ and $Y = \{(0, 3, \frac{1}{4}), (2, 1, \frac{3}{4})\}$, then we have the following quantities*

$$\mathbf{M} = \begin{pmatrix} d(\mathbf{x_1}, \mathbf{x_1}) & d(\mathbf{x_1}, \mathbf{x_2}) & d(\mathbf{x_1}, \mathbf{y_1}) & d(\mathbf{x_1}, \mathbf{y_2}) \\ d(\mathbf{x_2}, \mathbf{x_1}) & d(\mathbf{x_2}, \mathbf{x_2}) & d(\mathbf{x_2}, \mathbf{y_1}) & d(\mathbf{x_2}, \mathbf{y_2}) \\ d(\mathbf{y_1}, \mathbf{y_1}) & d(\mathbf{y_1}, \mathbf{x_2}) & d(\mathbf{y_1}, \mathbf{y_1}) & d(\mathbf{y_1}, \mathbf{y_2}) \\ d(\mathbf{y_2}, \mathbf{y_1}) & d(\mathbf{y_2}, \mathbf{x_2}) & d(\mathbf{y_1}, \mathbf{y_1}) & d(\mathbf{y_2}, \mathbf{y_2}) \end{pmatrix} \approx \begin{pmatrix} 0.0 & 1.0 & 3.0 & 2.2 \\ 1.0 & 0.0 & 3.2 & 1.4 \\ 3.0 & 3.2 & 0.0 & 2.8 \\ 2.2 & 1.4 & 2.8 & 0.0 \end{pmatrix} \tag{62}$$

$$\mathbf{p} = \left(\frac{1}{2}, \frac{1}{2}, 0, 0\right)^T \tag{63}$$

$$\mathbf{q} = \left(0, 0, \frac{1}{4}, \frac{3}{4}\right)^T. \tag{64}$$

[8]It can be easily shown that this matrix meets the conditions (6).

With the method described in Section 2.4 and taking $\lambda = 200$ we find,

$$\mathbf{P}^\lambda = \begin{pmatrix} 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0.50 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{65}$$

which can be easily checked by hand for such a small example. The cost associated with this transport matrix is, $d_M^\lambda(\mathbf{p}, \mathbf{c}) = \langle \mathbf{P}^\lambda, \mathbf{M} \rangle_F = 2.06\dots$.

This example illustrates one small problem with this methodology - the resulting transport matrix contains a lot of zeroes. This is not much of a problem for small systems as in this example, but because the matrix scales with $(n+m)^2$ it can quickly becomes computationally expensive for larger systems. Ideally we would like to shrink the problem down a bit. Luckily, with the setup described in this section, we can slightly simplify the problem. This leads us to the following theorem.

Theorem 3.3. *Let X and Y be two sets of points as in Section 3.1. One can find the Sinkhorn distance between X and Y by carrying out the following steps.*

1. *First define*

$$\mathbf{M}' \in \mathbb{R}^{n\times m}, \quad M'_{ij} := d(\mathbf{x}_i, \mathbf{y}_j)\,, \tag{66}$$

   this will serve as a sort of distance matrix, but it does not satisfy the requirements for a real distance matrix. We will also need the following probability vectors,

$$\mathbf{w} \in \mathbb{R}^n, \quad \mathbf{w} := (w_1, w_2, \cdots, w_n)^T\,, \tag{67}$$

$$\mathbf{v} \in \mathbb{R}^m, \quad \mathbf{v} := (v_1, v_2, \cdots, v_m)^T\,. \tag{68}$$

2. *Now perform algorithm 2.12, starting with the matrix $e^{-\lambda \mathbf{M}'}$, such that the row-sums are $\mathbf{w}$ and the column-sums are $\mathbf{v}$, and name the resulting matrix $\mathbf{P}'^\lambda$.*

3. *We now claim the following*

$$d_\lambda(X, Y) = \langle \mathbf{P}'^\lambda, \mathbf{M}' \rangle_F\,. \tag{69}$$

Proof. What we know from Section 2.4, is that if we take $\mathbf{M}$ as above, and $\mathbf{P}^\lambda$ found with algorithm 2.12, then

$$d_\lambda(X, Y) := d_M^\lambda(\mathbf{p}, \mathbf{q}) = \langle \mathbf{P}^\lambda, \mathbf{M} \rangle_F\,. \tag{70}$$

What we would like to show is that this is equal to $\langle \mathbf{P}'^\lambda, \mathbf{M}' \rangle_F$. From step 1. in algorithm 2.12 and equations (59) and (60) it follows that

$$P_{ij} = 0 \text{ for } i \in [\![n+1, n+m]\!] \text{ for all } j, \tag{71}$$

$$P_{ij} = 0 \text{ for } j \in [\![n]\!] \text{ for all } i. \tag{72}$$

This means that

$$\langle \mathbf{P}^\lambda, \mathbf{M} \rangle_F = \sum_{\substack{i\in[\![n+m]\!] \\ j\in[\![n+m]\!]}} P_{ij} M_{ij} = \sum_{\substack{i\in[\![n]\!] \\ j\in[\![n+1,n+m]\!]}} P_{ij} M_{ij} = \sum_{\substack{i\in[\![n]\!] \\ j\in[\![m]\!]}} P_{i,j+n} M_{i,j+n}\,. \tag{73}$$

From the definitions of $\mathbf{M}$ and $\mathbf{M}'$ we get that

$$M'_{ij} = M_{i,j+n}, \quad \forall\, i \in [\![n]\!], \forall\, j \in [\![m]\!]\,, \tag{74}$$

and by extension

$$e^{-\lambda M'_{ij}} = e^{-\lambda M_{i,j+n}}, \quad \forall\, i \in [\![n]\!], \forall\, j \in [\![m]\!]\,. \tag{75}$$

Similarly

$$w_i = p_i\,, \quad \forall i \in [\![n]\!]\,, \tag{76}$$
$$v_j = q_{j+n}\,, \quad \forall j \in [\![m]\!]\,. \tag{77}$$

Let $\mathbf{P}_0^\lambda$ and $\mathbf{P}_0'^\lambda$ be the matrices after performing step 1. of algorithm 2.12 on $e^{-\lambda \mathbf{M}}$ and $e^{-\lambda \mathbf{M}'}$ respectively, then by (75),

$$P'_{0,ij} = P_{0,i,j+n}\,, \quad \forall\, i \in [\![n]\!], \forall\, j \in [\![m]\!]\,. \tag{78}$$

Suppose now that

$$P'_{k,ij} = P_{k,i,j+n}\,, \quad \forall\, i \in [\![n]\!], \forall\, j \in [\![m]\!], \text{ for some } k \in \mathbb{N}\,, \tag{79}$$

Then for all $i_0 \in [\![n]\!]$ and all $j_0 \in [\![m]\!]$,

$$P_{k+1,i_0,j_0+n} = \tag{80}$$
$$P_{k,i_0,j_0+n} \frac{p_{i_0}}{\sum_{j\in[\![n+m]\!]} P_{k,i_0,j}} \frac{q_{j_0+n}}{\sum_{i\in[\![n+m]\!]} \left(P_{k,i,j_0+n}\, p_i / \left(\sum_{j\in[\![n+m]\!]} P_{k,ij}\right)\right)} = \tag{81}$$
$$P'_{k,i_0,j_0} \frac{w_{i_0}}{\sum_{j\in[\![m]\!]} P_{k,i_0,j+n}} \frac{v_{j_0}}{\sum_{i\in[\![n]\!]} \left(P_{k,i,j_0+n}\, p_i / \left(\sum_{j\in[\![m]\!]} P_{k,i,j+n}\right)\right)} = \tag{82}$$
$$P'_{k,i_0,j_0} \frac{w_{i_0}}{\sum_{j\in[\![m]\!]} P'_{k,i_0,j}} \frac{v_{j_0}}{\sum_{i\in[\![n]\!]} \left(P'_{k,i,j_0}\, p_i / \left(\sum_{j\in[\![m]\!]} P'_{k,ij}\right)\right)} = \tag{83}$$
$$P'_{k+1,i_0,j_0}\,. \tag{84}$$

By induction we find that

$$P'_{k,ij} = P_{k,i,j+n}\,, \quad \forall\, i \in [\![n]\!], \forall\, j \in [\![m]\!], \forall k \in \mathbb{N}\,. \tag{85}$$

Since $\mathbf{P}_k^\lambda \to \mathbf{P}^\lambda$ and $\mathbf{P}_k'^\lambda \to \mathbf{P}'^\lambda$ we find

$$P'_{ij} = P_{i,j+n}\,, \quad \forall\, i \in [\![n]\!], \forall\, j \in [\![m]\!]\,. \tag{86}$$

By combining equations (73), (74) and (86) we finally conclude.

$$\langle \mathbf{P}^\lambda, \mathbf{M}\rangle_F = \sum_{\substack{i\in[\![n]\!]\\ j\in[\![m]\!]}} P_{i,j+n} M_{i,j+n} = \sum_{\substack{i\in[\![n]\!]\\ j\in[\![m]\!]}} P'_{i,j} M'_{i,j} = \langle \mathbf{P}'^\lambda, \mathbf{M}'\rangle_F\,. \tag{87}$$

□

3.2 Python algorithm

In order to perform algorithm 2.12 we will be using python. The script used to obtain the results of this thesis will be made available together with the thesis itself, but below is a short overview in pseudo-code. Let $\mathbf{w}$, $\mathbf{v}$ and $\mathbf{M}$ be as defined in theorem 3.3, then we numerically calculate the Sinkhorn distance as follows,

Algorithm 3.4 (Python Sinkhorn)**.**

Division of two vectors is defined as $\mathbf{a}/\mathbf{b} = (a_1/b_1, \cdots, a_n/b_n)$.

```
1: $\mathbf{K} = e^{-\lambda \mathbf{M}}$
2: $\bar{\mathbf{K}} = \mathrm{diag}(\mathbf{1}_n/\mathbf{w})\mathbf{K}$
3: $\mathbf{u} = (1/n, \cdots, 1/n)^T \in \mathbb{R}^n$
4: while (row sums of $\mathbf{P}^\lambda$) - $\mathbf{w}$ are bigger than desired do
5:     $u = \mathbf{1}_n/(\bar{\mathbf{K}}(\mathbf{v}^T/(\mathbf{K}^T\mathbf{u})))$
6: end while
7: $\mathbf{P}^\lambda = \mathrm{diag}(\mathbf{u})\,\mathbf{K}\,\mathrm{diag}(\mathbf{v}/\mathbf{K}^T\mathbf{u})$
8: $d_M^\lambda(\mathbf{w}, \mathbf{v}) = \mathrm{sum}\ (P_{ij}M_{ij})$
```

3.2.1 On the choice of λ

The smallest positive number that can be represented in python is approximately $10^{-323.6}$, or roughly $e^{-744.03}$; anything smaller is rounded down to 0. Our matrix $e^{-\lambda\mathbf{M}}$ contains entries $e^{-\lambda M_{ij}}$, and we do not want any of these to be rounded down to 0. This has some implications for the values of λ that we can choose. A safe choice for λ is the following,

$$\lambda = \lfloor 744/\max_{i,j} M_{ij} \rfloor \leq 744/\max_{i,j} M_{ij} \quad \Rightarrow \quad e^{-\lambda M_{ij}} > e^{-744.03} \approx 0\,, \quad \forall i,j\,. \tag{88}$$

We will demonstrate why this is required with a small example.

Example 3.5. *Suppose that we are interested in the distance between X and Y such that they both contain two elements. In order to find the Sinkhorn distance between these two sets we are going to use the method described in theorem 3.3. Let $\mathbf{M}$ be as defined in said theorem and suppose that*

$$\mathbf{M} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}\,. \tag{89}$$

Now note that $\lfloor 744/\max_{i,j} M_{ij} \rfloor = 372$, but for this example we will take $\lambda = 400$. We are assuming that none of the weights are zero, so we start with the matrix

$$e^{-\lambda\mathbf{M}} = \begin{pmatrix} e^{-400} & e^{-800} \\ e^{-800} & e^{-400} \end{pmatrix}\,. \tag{90}$$

However, in memory this is stored as

$$e^{-\lambda\mathbf{M}} = \begin{pmatrix} e^{-400} & 0 \\ 0 & e^{-400} \end{pmatrix}\,, \tag{91}$$

and so algorithm 2.12 simply converges to

$$e^{-\lambda \mathbf{M}} = \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} . \tag{92}$$

In general this obviously does not converge to a matrix with row-sums $\mathbf{w}$*.*

One might argue that the condition (88) is insufficient, since we perform a number of operations on the entries of $e^{\lambda \mathbf{M}}$. While this is true, we do not have to worry about this, since python will raise a *ZeroDivisionError* in these cases as we will now show.[9] The first operation that could cause issues is the first time $\mathbf{K}^T \mathbf{u}$ is calculated, i.e.

$$(\mathbf{K}^T \mathbf{u})_j = \sum_{i=1}^{n} \frac{1}{n} e^{-\lambda M_{ij}} . \tag{93}$$

While it is unlikely if could be that y_{j_0} happens to be relatively far away from all $\mathbf{x_i}$, such that for all i,

$$\frac{1}{n} e^{-\lambda M_{ij_0}} \approx \frac{1}{n} e^{-744} < e^{-744} , \tag{94}$$

which would mean that $(\mathbf{K}^T \mathbf{u})_{j_0} = 0$, however, this would raise a ZeroDivisionError in the next step when calculating $\mathbf{v}^T / (\mathbf{K}^T \mathbf{u})$.

Python has another limitation on the other and of the scale, because any number greater than approximately e^{710} will be assigned inf. While this does not crash the script, it does notify you of an *overflow*. This 'number' inf has the property that 1/inf = 0. Because $|710| < |-744|$ we can have that $1/e^{-740} = \text{inf}$. This allows $\mathbf{v}^T / (\mathbf{K}^T \mathbf{u})$ to have some infinity entries, however this is not a problem since we are notified, allowing us to lower λ accordingly.

The final component that could potentially suffer from round-off errors is $\bar{\mathbf{K}}$, however, note that $0 < w_i < 1$ for all i and so $1/w_i > 1$. We now that $\mathbf{K}$ does not suffer from round-off errors and so neither does

$$\bar{K}_{ij} = K_{ij} \frac{1}{w_i} > K_{ij} > e^{-744} , \quad \forall i, j . \tag{95}$$

Since none of the operations cause round-off errors that go unnoticed we conclude that restraint (88) is sufficient, provided that no errors are raised, in which case we can simply lower λ a little.

3.3 Method analysis

Theorem 3.3 provides us with a method of estimating the Sinkhorn distance of points in an underlying metric space. The accuracy of our estimation of the optimal transport distance mainly depends on two factors.

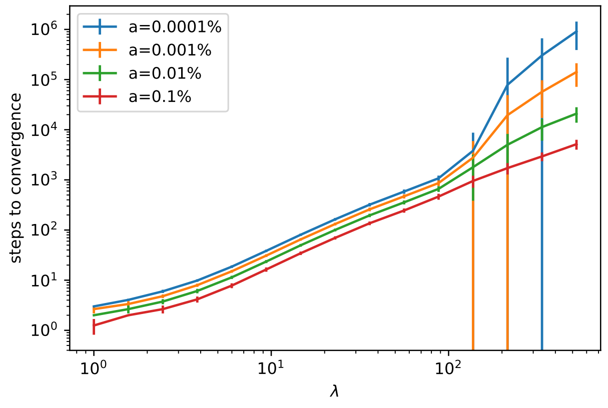
[9] Of course, a ZeroDivisionError is not something we want, but if this does happen we can easily lower λ a bit, which resolves the issue. This is different from the problem with $e^{-\lambda \mathbf{M}}$ since in that case the error goes undetected, which results in an incorrect estimate of $\mathbf{P}^\lambda$. If one performs algorithm 3.3 then it will run indefinitely since the difference between the row sums and $\mathbf{w}$ does not converge to 0.

The first of these is our value of λ. Since for $\lambda \to \infty$ we have $d_M^\lambda \to d_M$, so higher values for λ provide a higher accuracy, we cannot however, increase λ as much as we like as we have just seen in Section 3.2.1. For values that exceed a certain limit, computers will round $e^{-\lambda M_{ij}}$ down to zero, which breaks our algorithm, this is commonly referred to as the "machine-precision limit".

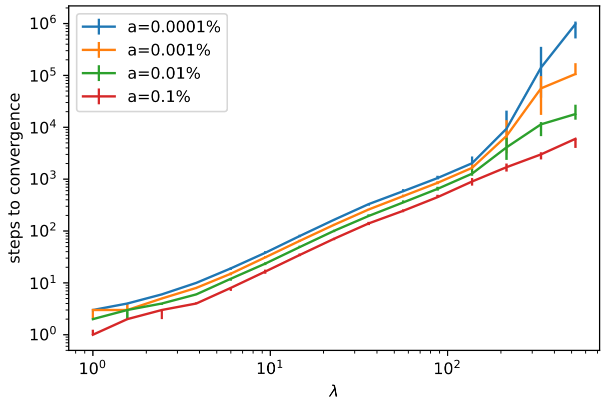
Another factor that impacts the accuracy of our estimation is the number of times (or *steps*) we repeat step 2. and 3. of algorithm 2.12. While the repeatedly scaled matrix eventually converges to $\mathbf{P}^\lambda$, it takes some steps to get there, and if we were to prematurely stop before we have sufficiently converged then this would negatively impact the validity of our results.

Other sources of inaccuracies like the miscalculations caused by the use of floating-point arithmetic, are not considered and deemed insignificant.

In order to analyse the convergence of algorithm 3.4 we generated 20 pairs of sets of 200 points of $(0,1)^2 \subset \mathbb{R}^2$. We performed steps 1.-3. of the algorithm. Then we performed one iteration of step 5. and calculated $\mathbf{P}^\lambda$, if all row-sums were within $a\%$ of $\mathbf{w}$ then we considered the algorithm to have converged[10], if not we performed another iteration and repeated. We repeated this process for different values of λ and a on the same pairs of sets and recorded the number of steps needed for the algorithm to converge. We then took the mean and median over the 20 pairs of sets for each combination of λ and a. The results can be seen in figure 2.



(a)



(b)

Figure 2: Figures that show the number of steps needed for algorithm 3.4 to converge for multiple values of λ. Convergence is achieved when the difference between the row-sums of $\mathbf{P}^\lambda$, and $\mathbf{w}$ differ no more than $a\%$. On the left-hand side we show the mean of the steps needed, error bars denote the standard deviation. On the right-hand side the median is shown, error bars denote the 25-th and 75-th percentile.

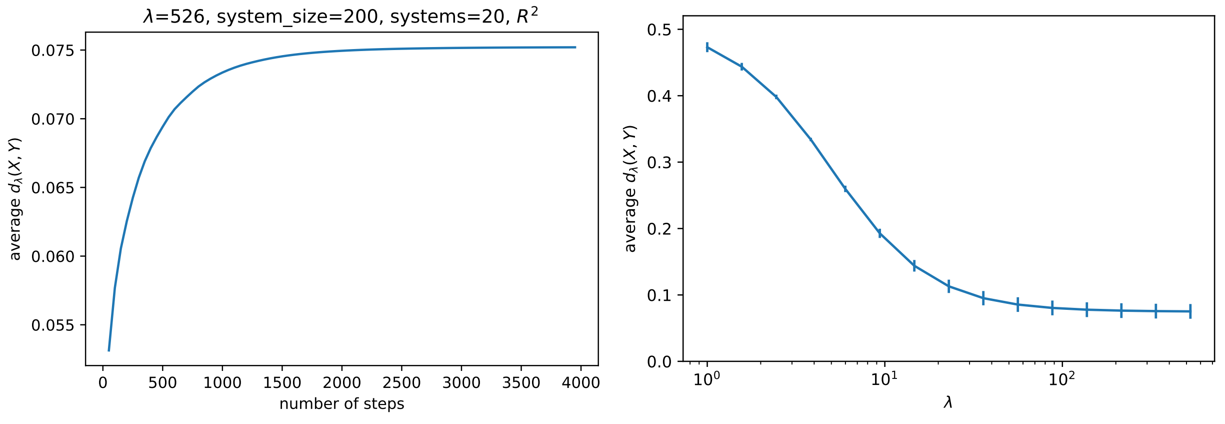
In order to see the differences in d_M^λ for various numbers of steps we generated 20 systems of two sets of 200 random points in $(0,1)^2$ with uniform weights. For each of these systems we calculated the Sinkhorn distances between the two sets after multiples of 50 steps. Then for each number of steps we averaged over the distances of all 20 systems. Since all points

[10]Note that it is guaranteed that the column-sums yield $\mathbf{v}$, so there is no need to check this.

lie within the unit cube, we know that the distance between two points cannot exceed $\sqrt{2}$, and so we have chosen $\lambda = 526 = \lfloor 744/\sqrt{2} \rfloor$. The results can be seen in figure 3a. We have done a similar analysis for different values of λ, where we fixed the number of steps to be the 100-th percentile, i.e. the maximum, of the 20 data points for each λ of $a = 0.001\%$ of figure 2a. This was done to ensure that we at least estimate $\mathbf{P}^\lambda$ with an error less than 0.001%. The results are shown in figure 3b. These two variables show different behaviour - a higher λ results in a lower distance, whereas a higher number of steps results in a higher distance. However, this does not imply that we can simply find the 'best' solution with a high λ and a low number of steps. Notice that a high λ is a *luxury*, whereas a high number of steps is a *necessity*. From theory we know that a higher λ corresponds to a higher α for $d_{M,\alpha}$, and that if $a_1 < a_2$ then $U_{a_1}(\mathbf{r}, \mathbf{c}) \subset U_{a_2}(\mathbf{r}, \mathbf{c})$ and so

$$d_{M,\alpha_1} := \min_{\mathbf{P} \in U_{\alpha_1}(\mathbf{r},\mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle_F \leq \min_{\mathbf{P} \in U_{\alpha_2}(\mathbf{r},\mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle_F =: d_{M,\alpha_2} . \tag{96}$$

What we mean by a *luxury* is that while a higher λ provides us with an answer closer to the optimal transport distance, it also comes at a higher computational cost as is described by figure 2. Conversely, the number of steps is required to be high enough, because if it is not then $\mathbf{P}^\lambda$ has not fully converged to a matrix with row sums $\mathbf{w}$, hence it is a *necessity* to have a high number of steps.



(a) (b)

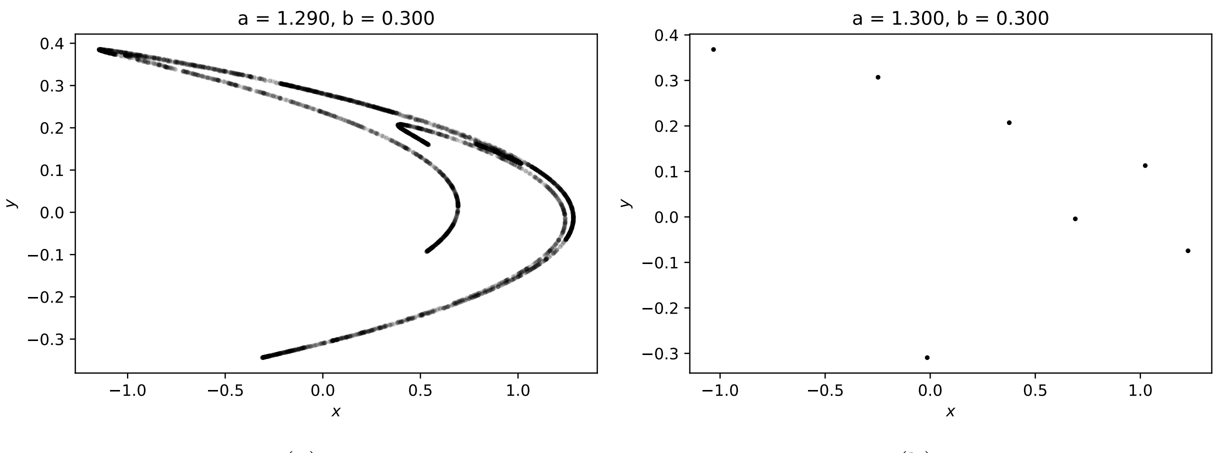
Figure 3: Figures that show the average Sinkhorn distance between two random sets of points of $\mathbb{R}^2$. On the left we varied the number of steps for the calculation of $\mathbf{P}^\lambda$, and on the right the value of λ. For every value of the two parameters we averaged the Sinkhorn distance of 20 systems, each containing two set of 200 random points in $(0,1)^2$. On the left we fixed $\lambda = 526$ and on the right we fixed the number of steps, *steps* = 3000.

3.4 Hénon systems

We will be testing our methodology on a dynamical system called the *Hénon system*, we have taken inspiration from Ref. [8]. Given some initial (x_0, y_0), we calculate subsequent points as follows,

$$(x_{n+1}, y_{n+1}) = (1 + y_n - a x_n^2,\ b x_n) . \tag{97}$$

The behavior of this system is governed by the parameters a and b, the sets generated for some different parameters can be seen in figure 4.



(a) (b)

Figure 4: Two sets of points generated with the Hénon map. Both sets result from the starting conditions $x_0 = 0.1$ and $y_0 = 0$, but the first 1000 points of the iteration are considered transient and do not appear in the plot. Both sets contain 5000 points. Figure 4a has parameters $a = 1.29, b = 0.30$ and figure 4b has parameters $a = 1.30, b = 0.30$. The dots are transparent, so dark spots indicate a more data points and lighter spots indicate fewer data points.

While the parameters of figure 4a and figure 4b are very close, the resulting set of points are completely different. The latter appears to be periodic, while the former is attracted to some attractor. This results from the fact that the Hénon system is a chaotic one. Figure 5 shows the behavior of the Hénon iteration for different values of a at $b = 0.3$. For some values of a the system exhibits periodic behaviour, like $a = 1$ and $a = 1.25$, and for others, like $a = 1.4$ (the most commonly studied value), the system exhibits chaotic behaviour.

In order to see the change in the attractor as a result of varying the parameters a and b we have created a python script. In this script we generate subsequent points with the Hénon map starting from $(x_0, y_0) = (0.1, 0)$. Since we consider the first 1000 points as transient, the impact of the exact starting position is minute. We define the set $H_{a,b}$ as the set that contains N non-transient points of the iteration with parameters a and b, where N can vary from plot to plot. We generate two of these sets which differ in either a or b and calculate $\mathbf{M}'$ as in Theorem 3.3. For the histograms we simply use uniform ones, i.e.

$$\mathbf{r} = \mathbf{c} = (1/N, \cdots, 1/N) \in \mathbb{R}^N . \tag{98}$$

Finally we calculate the Sinkhorn distance between the two sets of Hénon points using algorithm 3.4.

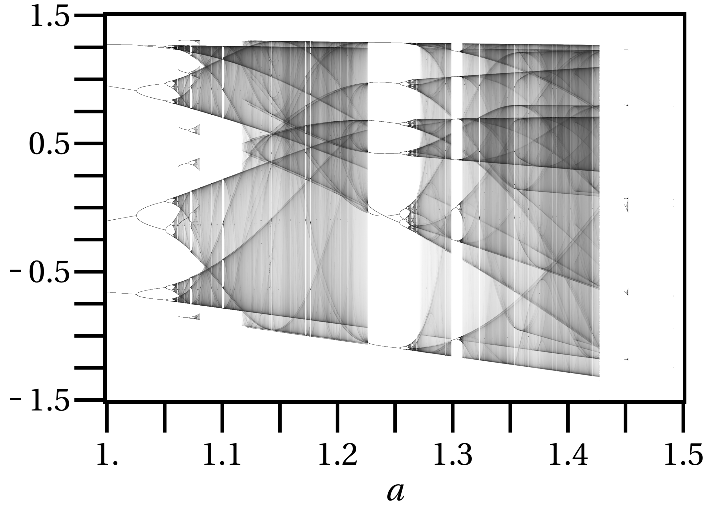


Figure 5: The y-axis shows the probability of x for a given a, a darker colour indicates a higher probability. Credit: [9].

4 Results

In this chapter we present the results of the method described in Section 3.4.

In Figure 6 we show the Sinkhorn distances between points on the attractor of $H_{1.22,0.3}$ and $H_{a,0.3}$, where a is shown on the x-axis. As expected the attractors differ more for greater differences in a, however the curve is not nice and smooth. This is likely a result from the fact that we have a limited number of points of the Hénon system. This hypothesis is further substantiated by the fact that the graph of the "fast calculation" dataset which used less points is more jagged than the graph of the "slow calculation" which used more points of the Hénon system. This shows that the system is sensitive to the number of data points used. We expect that the graph will become more smooth if the number of points is increased and the steps between a made smaller. Unfortunately we do not currently posses the resources to perform a more *sophisticated* calculation.[11] We do not necessarily expect the graph to decrease monotonically as the attractors can have different topologies for different ranges of a. In order to demonstrate this we would also have to perform a more sophisticated calculation.

[11]Another theory we have is that the graph approaches a fractal as the number of points increase to infinity and as the step size becomes infinitesimally small. We think this could be a possibility due to the fractal-like nature of the attractor. We have however, no supporting evidence for this theory and it should be seen as a mere suggestion.

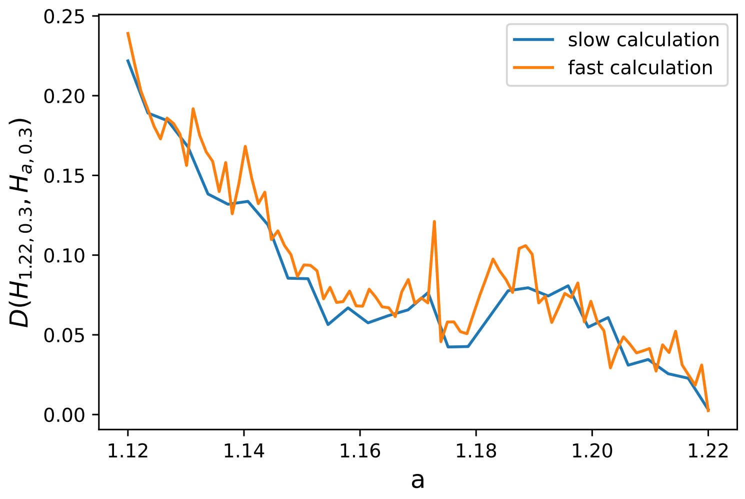


Figure 6: Figure that shows the distance between the points of a Hénon system with $b = 0.3$ and a as shown on the x-axis, and the points of a Hénon system with $b = 0.3$ and $a = 1.22$. The 'slow calculation' shows 30 values of a on a regular interval, with the distances calculated with 10000 steps of algorithm 3.4 and 6000 data points of the Hénon system. The 'fast calculation' was made with 2000 steps and 2000 points of the Hénon system and was calculated for 90 values of a with regular intervals. For every distance calculation we picked λ as described in Section 3.2.1.

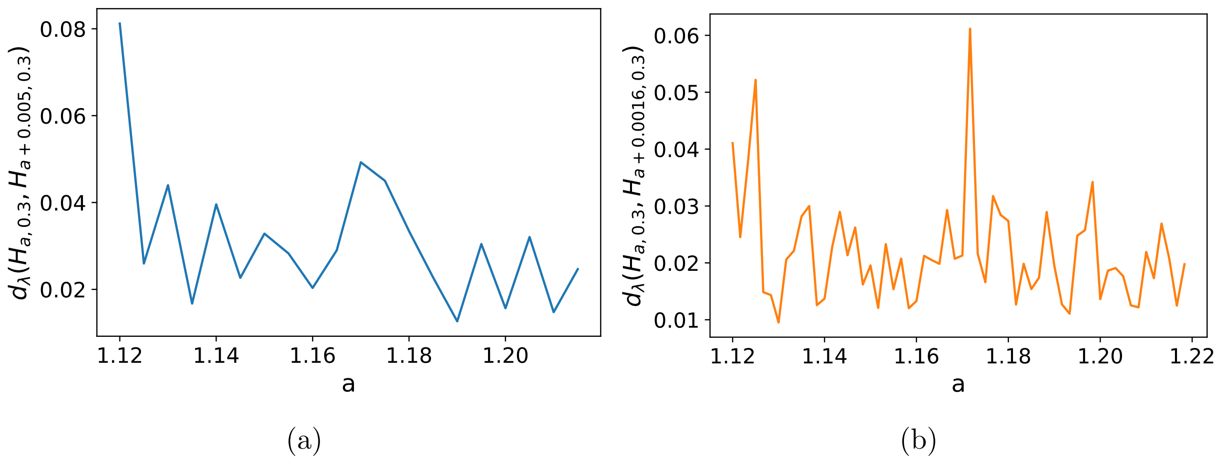


Figure 7: Figure that shows the distance between the points of a Hénon system with $b = 0.3$ and a as shown on the x-axis, and the points of a Hénon system with $b = 0.3$ and $a+0.005$ in Figure 7a, and $a+0.001667$ in Figure 7b. Figure 7a shows 20 values of a on a regular interval, with the distances calculated with 5000 steps of algorithm 3.4 and 5000 data points of the Hénon system. The Figure 7b was made with 1500 steps and 2000 points of the Hénon system and was calculated for 60 values of a with regular intervals. For every distance calculation we picked λ as described in Section 3.2.1.

In Figure 7 we performed a similar analysis as in Figure 6 but instead of calculating the distance to a fixed set of points we calculated the distance to the "next" set of points. While the data is very noisy, both the ”slow” and the ”fast” calculation show two peaks at similar values of a. Combined with the small peak in Figure 6 at the same value of a one might expect that this is due to a change in topology, but the sets $H_{0.3,1.17}$ and $H_{0.3,1.17166}$ show no drastic differences.

Unlike Figure 6 we have split the results into two plots. We made this decision because in Figure 6 both graphs show to same data for a given a, i.e. the Sinkhorn distance between $H_{0.3,a}$ and $H_{0.3,1.22}$, for different numbers of Hénon points. Conversely, in Figure 7, the data shown for a given a is the Sinkhorn distance between $H_{0.3,a}$ and $H_{0.3,a+\delta a}$, where a differs for Figure 7a and 7b making it unfair to compare the two.

In Figure 8 we performed an analysis similar to the one in Figure 6, but this time we varied b. This curve shows a more monotonic behaviour. As a sanity check we see that the distance approaches zero as the two sets approach each other. In Figure 9 the analysis was the same as in Figure 7 but instead we varied b. The huge peak is caused by the fact that the Hénon system is periodic for $a = 1.4$ and $b = 0.225$ as can be seen in Figure 10. The reason we do not see a similar peak in Figure 8 is because it uses slightly different values for b and hence the value of b that has a periodic behaviour was narrowly avoided.[12]

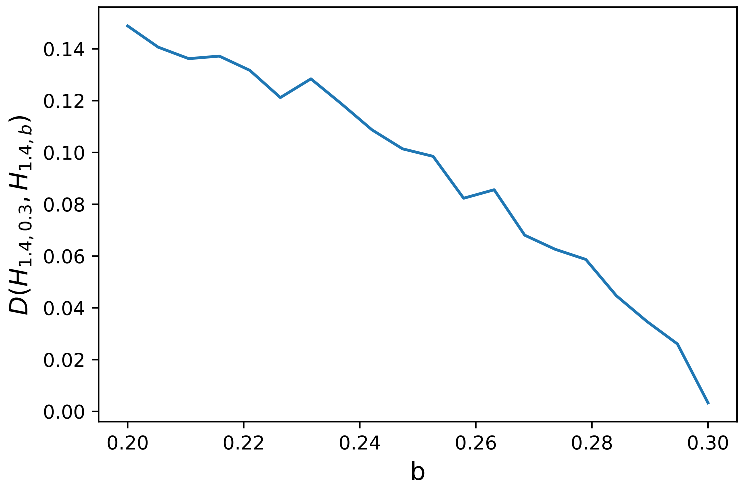
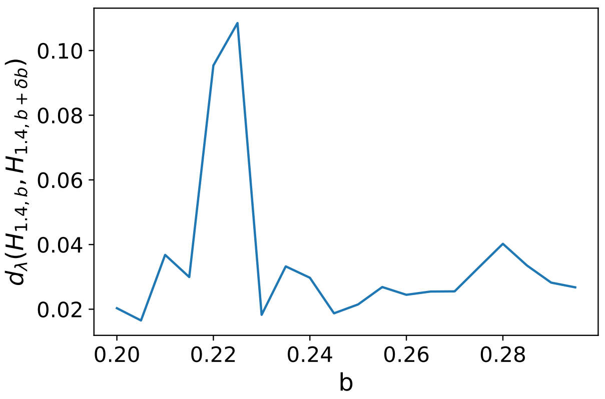
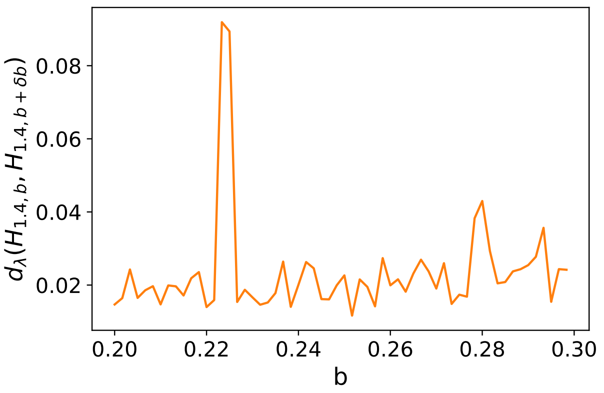


Figure 8: Figure that shows the distance between the points of a Hénon system with $a = 1.4$ and b as shown on the x-axis, and the points of a Hénon system with $a = 1.4$ and $b = 0.3$. The calculation was done by performing 5000 steps of algorithm 3.4 for each of the 20 values of b, and by using 5000 data points for the Hénon system, without the first 1000 transient data points.

[12]This is one of the effects caused by the coarseness of the calculations; using a finer x-axis will result in less detail being missed, but comes at the cost of being more computationally expensive.

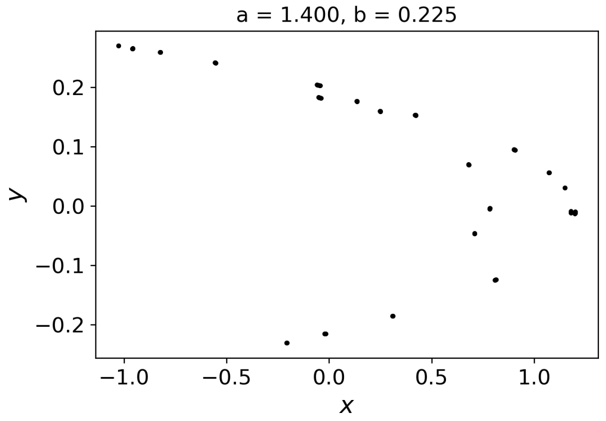


(a)

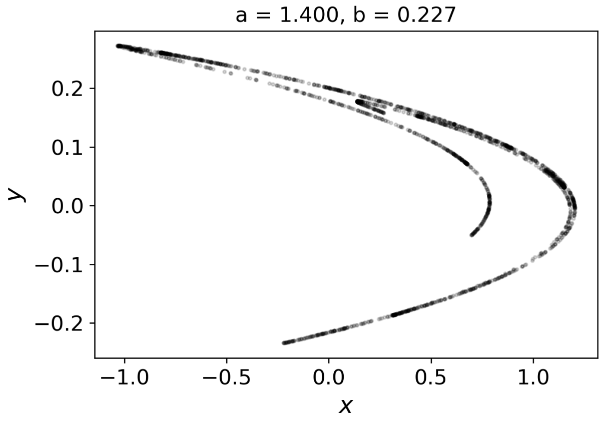


(b)

Figure 9: Figure that shows the distance between the points of a Hénon system with $a = 1.4$ and b as shown on the x-axis, and the points of a Hénon system with $a = 1.4$ and $b+0.005$ in Figure 9a, and $b+0.001667$ in Figure 9b. Figure 9a shows 20 values of a on a regular interval, with the distances calculated with 5000 steps of algorithm 3.4 and 5000 data points of the Hénon system. The Figure 9b was made with 1500 steps and 2000 points of the Hénon system and was calculated for 60 values of b with regular intervals. For every distance calculation we picked λ as described in Section 3.2.1.



(a)



(b)

Figure 10: On the left $H_{1.4,0.225}$ and on the right $H_{1.4,0.227}$. The number of points visible are 5000. The Sinkhorn distance between these sets can be seen in Figure 9.

5 Conclusion

In this thesis we looked at the concept of optimal transport and how an entropic regularisation allowed the application of Sinkhorn's theorem. We investigated the rate of convergence of this method and applied it to investigate the attractor of the Hénon system.

Our results show that the distribution of the points along the attractor is impacted by the number of data points. Unfortunately we did not posses the computational power required to generate enough data points such that this dependency can be mitigated. We tried to generate a system with 20000 points but python could not handle a matrix this big and raised a *MemoryError*.

In order to find the exact optimal transport distance between two sets you would need to let λ and the number of steps go to infinity. Of course this is not possible with any computer. Because of this we had to settle for the concerned values, but our results in Chapter 3.3 show that the concerned values provided ample convergence of the Sinkhorn distance to the optimal transport distance.

If I were to revisit this project I would start by rewriting my python script into a faster language like $C++$, this way I can calculate the Sinkhorn distances between the Hénon systems much faster which allows me to use more data points for the Hénon systems. This would hopefully resolve the noisy data like the one in Figure 6. With the extra speed in mind I could vary both a and b simultaneously, allowing for a "grid" of parameters. This would give more insight in the effects these parameters have on the attractor.

6 References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. arXiv: `1701.07875 [stat.ML]`.

[2] John Lee, Nicholas P. Bertrand, and Christopher J. Rozell. "Unbalanced Optimal Transport Regularization for Imaging Problems". In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1219–1232. DOI: `10.1109/TCI.2020.3012954`.

[3] Geoffry Schiebinger. "Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming". In: *Cell* (2019).

[4] Marco Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances*. 2013. arXiv: `1306.0895 [stat.ML]`.

[5] Richard Sinkhorn. "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices". In: *The Annals of Mathematical Statistics* 35.2 (1964), pp. 876–879. DOI: `10.1214/aoms/1177703591`. URL: `https://doi.org/10.1214/aoms/1177703591`.

[6] Martin Idel. *A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps*. 2016. arXiv: `1609.06349 [math.RA]`.

[7] David Knowles. "Lagrangian Duality for Dummies". In: (2010).

[8] Sjoerd Verduyn-Lunel. “Wasserstein distances in the analysis of time series and dynamical systems”. In: *Utrecht University* (2016).

[9] Wikipedia contributors. *Hénon map — Wikipedia, The Free Encyclopedia*. [Online; accessed 8-June-2024]. 2023. URL: `https://en.wikipedia.org/w/index.php?title=H%C3%A9non_map&oldid=1158650280`.