University of Groningen

Bachelor Thesis

Computing Science

Time-series classification using Hankel matrix based dissimilarity measures in Learning Vector Quantization

Author:
Lars Holdijk
s2534878

Supervisor:
prof. Michael Biehl
Mohammad Mohammadi, MSc.

July 23, 2018

Abstract

Time-series classification is an interesting and challenging sub-domain of classification problems. In distance based classification algorithms, the information hidden in the ordering of the time-series and the possibility of misalignment require the use of specialized dissimilarity measures. In this thesis we look at three such measures, all of which are based on Hankel matrices and the assumption that Hankel matrices with the same subspace originate from the same LTI-series and consequently from the same class. In previous work all three dissimilarity measures have shown competitive results when combined with k-Nearest Neighbours. In our work we combine two of the three dissimilarity measure with Generalized Learning Vector Quantization (GLVQ) using a rewriting of derivatives presented in earlier work. The results presented show promise for the dissimilarity measures to be applied in GLVQ.

Contents

Introduction

The problem of classification of novel data given a labelled training set is one of the most widely researched topics within the Machine Learning community. The classification of time-series is an interesting sub-domain of this.

Traditional classification problems and time-series classification problems are differentiated by the ordering of the attributes in the feature vector. Despite what the terminology suggests, whether the ordering is by time or not is irrelevant to the problem. The important trait is that in traditional classification problems this ordering is irrelevant as long as the ordering is uniform over the items in the dataset. For time-series classification problems there may however be discriminatory features dependent on the ordering of the attributes. [1]

Uncovering these latent features and using it for the classification of time-series is challenged by the possible misalignment of the series and the presence of noise [1]. The misalignment in the series can increase the difficulty of the comparison of time-series, an effect that is further amplified by the possible difference in sample length [2]. In some instances these issues can be largely solved by pre-processing steps aligning the data and truncating where needed. However, this is a computationally expensive process that can be rather slow [3]. Additionally, at the time of pre-processing it might not be known how to align the data or what the important segments are.

In recent literature [3, 4, 5], approaches to capture the latent information of time-series in a dissimilarity measure without requiring them to be aligned are being presented. These approaches rely on the subspaces of Hankel matrices to uncover the underlying Linear Time Invariant systems of the time-series. In this project we will discuss three dissimilarity measures using this approach. The first measure, from here on referred to as the subspace angle method, computes the angles between the subspaces of two Hankel matrices and uses them to construct a dissimilarity measure. The second and third attempt to approximate the angles between the subspaces and are therefore called the angle approximation and the rotation approximation. In [2] van Loon concludes that these dissimilarity measure provide a robust method for comparing time-series in k-Nearest Neighbours classification.

In the following we elaborate on the research conducted by van Loon and look at how two of the three dissimilarity measure can potentially be used in combination with a prototype based supervised learning algorithm such as Generalized Learning Vector Quantization (GLVQ). In his thesis [3], Mohammadi presents the derivatives of the two approximation methods required for GLVQ and validates their use. In this work we present a rewriting of the angle approximation dissimilarity measure Mohammadi presents and extend the validation of both approximation methods. For this purpose we use four real-world datasets from the UCR time-series repository [6] and investigate the influence of two important hyper-parameters.

In the next chapter, the theory behind Hankel matrix based dissimilarity measures is presented. In chapter 2, the three dissimilarity measure are given and discussed. In chapter 3, background is given on Generalized Learning Vec-

tor Quantization and Generalized Relevance Learning Vector Quantization to clarify why the derivatives of the dissimilarity measures are needed. In chapter 4, we give a rewriting of the derivatives of the two approximation methods. Chapter 5, introduces the two experiments we have performed to validate the use of Hankel matrix based dissimilarity measures in GLVQ. The results for these experiments are presented in chapter 6 and discussed in chapter 7. Lastly, we discuss further work in chapter 8.

1 — LTI-systems and their relation to Hankel Matrices

Dynamical systems model temporal information of a measurement sequence $y_k \in \Re^n$ as a function of a relatively low dimensional state vector $x_k \in \Re^d$ that changes over time. In other words dynamical models capture the temporal information describing how a time-series evolves over time.

Linear Time Invariant Systems are the simplest dynamical model and can be defined by two equations:

$$y_k = \boldsymbol{C}x_k, \qquad \text{given } x_0, \tag{1.1}$$
$$x_k = \boldsymbol{A}x_{k-1} + w_k, \tag{1.2}$$

where matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ are constant over time and w_k is uncorrelated zero mean Gaussian measurement noise. Eq. (1.1) is know as the *measurement equations* and eq. (1.2) as the *state equation*.

Now, if we assume that

- All time-series are an output of an LTI-system
- Time-series from the same class originate from the same LTI-system

we can try and classify the time-series by determining which LTI-system could output the time-series. To achieve this we need to estimate $\boldsymbol{A}$, $\boldsymbol{C}$ and the initial vector x_0 given a time-series $(y_1, y_2, ..., y_k)$ of length k. However, given the finite length k the triple $(\boldsymbol{A},\boldsymbol{C}, x_0)$ is not unique. Furthermore, identifying the dynamics $(\boldsymbol{A},\boldsymbol{C})$ and x_0 jointly is a computationally challenging convex problem [5].

Luckily, Hankel Matrices provide an easy method for comparing the underlying LTI-system of a time-series. As we will show in the next two sections we can use subspace identification methods [7] on Hankel Matrices to find the subspace of the underlying LTI-system.

1.1 Hankel Matrix

A Hankel matrix is a $m \times n$ matrix with constant anti-diagonals. Given a time-series $y_0, y_1, ..., y_{m+n}$ we can construct its associated Hankel Matrix $\boldsymbol{H}_y$

$$\boldsymbol{H}_y = \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_n \\ y_1 & y_2 & y_3 & \cdots & y_{n+1} \\ y_2 & y_3 & y_4 & \cdots & y_{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_m & y_{m+1} & y_{m+2} & \cdots & y_{m+n} \end{bmatrix},$$

where the columns of the matrix H_y correspond to all overlapping sub-sequences of the data of length m, shifted by one element.

We are only interested in the classification of one-dimensional time-series and therefore the definition of a Hankel Matrix above only contains elements that are numbers. A generalization of this method for multi-dimensional time-series is given by Mohammadi et al. in [4].

1.2 Subspace identification

Given a Hankel matrix H_y constructed from the output y_k of a LTI-system defined by equations 1.1 and 1.2 we can rewrite said equations to express y_k in terms of the triple $(\boldsymbol{A}, \boldsymbol{C}, x_0)$. It must be noted that it is only possible to rewrite equations (**??**) in the absence of noise $w_k = 0$ [5].

$$\begin{aligned} y_k &= \boldsymbol{C}x_k \\ &= \boldsymbol{CA}x_{k-1} \\ &= \boldsymbol{CA}^2x_{k-2} \\ &= ... \\ &= \boldsymbol{CA}^kx_{k-k}. \end{aligned} \tag{1.3}$$

Using (1.3) we can rewrite H_y as the matrix product of a column vector and row vector

$$H_y = \Gamma X, \tag{1.4}$$

where

$$\Gamma = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^k \end{bmatrix} \qquad \text{and} \qquad X = \begin{bmatrix} x_0 & x_1 & \dots & x_n \end{bmatrix}.$$

From (1.4) we can conclude that the columns of H_y and Γ span the same subspace. In other words, regardless of the initial values X, Hankel matrices from the same LTI-system span the same subspace. The significance of this realization is that we can now compare the subspace of Hankel matrices representing a time-series to determine if the time-series originate from the same LTI-system. With the previously made assumption that time-series from the same class originate from the same LTI-system, we can use this realization to classify our time-series.

In the next chapter we will look at three different approaches for using the property of Hankel matrices described above to create a dissimilarity measure.

2 — Hankel Matrix based dissimilarity measures

As we discussed in chapter 1 Hankel matrices originating from the same Linear Time Invariant system span the same subspace regardless of their initial values. In this section we will discuss three methods that can exploit this property of Hankel matrices to compute a dissimilarity between two time-series. As mentioned before we assume for this purpose that all time-series are the output of a LTI-system and that all time-series from the same class originate from the same LTI-system.

2.1 Subspace Angle

The classic approach for comparing two subspaces it to compute the angles between them. This approach uses the canonical correlations of two subspaces.

2.1.1 Canonical correlation

Canonical correlations are a measure of the angles between the closest vectors of two subspaces. For classification purposes its important to know that a high canonical correlation value corresponds to a small subspace angle and thus indicates a low dissimilarity score.

Given two subspaces F and G with their dimensions constraint such that

$$p = dim(F) \geq dim(G) = q \geq 1,$$

the smallest principle angle $\theta_1(F, G) \in [0, \frac{1}{2}\pi]$ between F and G is defined by

$$\theta_1 = \cos^{-1}\left(\max_{u_1 \in F} \max_{v_1 \in G} u_1^T v_1\right), \tag{2.1}$$

with

$$||u||_2 = ||v||_2 = 1.$$

Using u_1 and v_1, the next principal angle $\theta_2(F, G)$ can be defined as the smallest angle between the orthogonal complement of F with respect to u_1 and that of G with respect to v_1. This process is repeated until either subspace F or G is empty. The canonical correlation can thus be defined as

$$\theta_{k+1} = \cos^{-1}\left(\max_{u \in F_{u_k}^{\perp}} \max_{v \in G_{v_k}^{\perp}} u^T v\right), \tag{2.2}$$

with

$$||u||_2 = ||v||_2 = 1,$$

where $F_{u_k}^{\perp}$ is the orthogonal complement of F with respect to u_k and $G_{v_k}^{\perp}$ is the orthogonal complement of G with respect to v_k.

A second method for computing the canonical correlations is to compute the singular value decomposition [5]. This method is however only possible when the subspaces are defined as the range of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, which is the case for our Hankel matrices.

Once the canonical correlation is computed the found subspace angles can be used to create a dissimilarity measure. Various methods for doing so are available. Example include, taking the smallest subspace angle as the dissimilarity value [2], computing the Martin distance [3] or summing the principal angles

$$d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 1 - \sum_{i=1}^{d} \theta_i \tag{2.3}$$

$$= 1 - \sum_{i=1}^{d} \cos^{-1}\left(\max_{u \in F^{\perp}_{u_{i-1}}} \max_{v \in G^{\perp}_{v_{i-1}}} u^T v \right), \tag{2.4}$$

where d is the number of orthonormal vector bases. It is important to note that in this dissimilarity measure does not use the Hankel matrices directly. F and G are the orthonormal subspaces bases of the two Hankel matrices $\hat{\boldsymbol{H}}_{\boldsymbol{p}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{q}}$.

A more complete explanation of canonical correlation is given in [3, 5]. Both papers also discuss methods for reducing the influence of noise in the Hankel matrices, using principal component analysis and discriminant canonical correlation respectively.

2.2 Angle Approximation

In [8] Binlong et al. define an alternative dissimilarity measure to the one described in section 2.1.1. Their primary motive is to be able to work with noisy data.

The angle and rotation approximation discussed in the next section rely on the use of the Frobenius norm for normalization. Given a $m \times n$ matrix $\boldsymbol{H}$ the Frobenius norm $||\boldsymbol{H}||_F$ of $\boldsymbol{H}$ is given by

$$||\boldsymbol{H}||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij})^2}. \tag{2.5}$$

Binlong et al. propose the following dissimilarity measure based on the triangle inequality

$$d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 4 - ||\hat{\boldsymbol{H}}_{\boldsymbol{p}} \cdot \hat{\boldsymbol{H}}^{\boldsymbol{T}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}} \cdot \hat{\boldsymbol{H}}^{\boldsymbol{T}}_{\boldsymbol{q}}||_F, \tag{2.6}$$

where $\boldsymbol{H}_{\boldsymbol{p}}$ is normalised using

$$\hat{\boldsymbol{H}}_{\boldsymbol{p}} = \frac{\boldsymbol{H}_p}{\sqrt{||\boldsymbol{H}_p \cdot \boldsymbol{H}_p^T||_F}}. \tag{2.7}$$

To simplify finding the derivative of the dissimilarity measure we will use a slightly modified version. By squaring the Frobenius norm we will be able to negate the effect of the root function in the Frobenius norm as show in (2.5)

$$d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 4 - ||\hat{\boldsymbol{H}}_{\boldsymbol{p}} \cdot \hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}} \cdot \hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}||_F^2. \tag{2.8}$$

2.3 Rotation Approximation

The dissimilarity measure proposed by Binlong et al. discussed in section 2.2 successfully reduces the influence of noise present in the Hankel matrices. While this is evidently important for their purpose, reducing the noise has the side effect that the matrix $\boldsymbol{H}_p \cdot \boldsymbol{H}_p^T$ used by Binlong et al. is invariant to the state change direction. Therefore, the proposed dissimilarity measure may not be suitable for classification problems where the direction of the time-series is important, such as gesture recognition [9].

To solve this issue Lo Presti et al. propose a approximation method very similar to (2.8).

$$d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 4 - ||\hat{\boldsymbol{H}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}}||_F^2, \tag{2.9}$$

where $\boldsymbol{H}_{\boldsymbol{p}}$ is normalised using

$$\hat{\boldsymbol{H}}_{\boldsymbol{p}} = \frac{\boldsymbol{H}_p}{||\boldsymbol{H}_p||_F}. \tag{2.10}$$

Similar to the angle approximation, the Frobenius norm in (2.9) is squared to simplify finding the derivative.

3 — Learning Vector Quantization

Learning Vector Quantization or LVQ in short is a prototype based classification algorithm closely related to the nearest neighbours classifiers. Both set of algorithms heavily rely on the concept of a distance, or dissimilarity, measure for making classification decisions. To better understand Learning Vector Quantization and the restrictions it imposes on dissimilarity measures we will first look at the k-Nearest Neighbours classifier.

3.1 k-Nearest Neighbours

K-Nearest Neighbours is a non-parametric classification algorithm, meaning that it does not depend on any assumptions about the data for its use [10]. The first appearance of the k-Nearest Neighbour concept was in 1951 by Fix and Hodges in [11] (republished in 1989).

In k-Nearest Neighbours classification the decision to classify x into the class θ is allowed to depend only on a collection of n already classified samples $(x_1, \theta_1), (x_2, \theta_2), ..., (x_n, \theta_n)$[10]. If the classified samples(x_i, θ_i) are independently identically distributed according to the distribution of (x, θ) it is reasonable to assume that samples close together are of the same class or atleast have the same posterior probability distribution of their respective class[10]. In the k-Nearest Neighbours, a novel data point is therefore classified by finding the k closest already classified samples and assigning the novel data point to the class that won the majority vote.

Finding the k closest already classified samples is a non-trivial tasks. The success of the classification depends greatly on the distance measure. It is therefore not a surprise that a lot of research within the machine learning community is aimed at finding and improving distance measures.

K-Nearest Neighbours is a conceptually very simple algorithm with no required training period and only one parameter k. For these reasons it has seen a lot of applications and is one of the most widely used classification algorithms. However, it has a few drawbacks: it is expensive, both computationally and on storage and it is sensitive to outliers [12].

3.2 Learning Vector Quantization

As stated earlier, LVQ is closely related to k-Nearest Neighbours. Both classify a novel data point based on a dissimilarity measure. K-Nearest Neighbours and LVQ are different by the objects they compare the novel data points to and compute the dissimilarity measure from. While k-Nearest Neighbours computes the dissimilarity measure respective to all already classified samples and decides based on the majority vote, LVQ bases its decision on a small set of prototypes representing the classes. In LVQ, classification is achieved by assigning the novel data point to the class of the closest prototype.

To define the prototypes used for classification, LVQ requires a training phase. Before the training phase, a set of prototypes is initialized for different classes. During this training phase, the prototypes are trained by iteratively presenting a single already classified sample and updating the prototype closes to this sample. The prototype is moved closer towards the sample when they are of the same class and further away otherwise.

In [13] Kohonen formally defines the basic LVQ process

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \tag{3.1}$$
$$\text{if } x \text{ and } m_c \text{ belong to the same class}$$
$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)] \tag{3.2}$$
$$\text{if } x \text{ and } m_c \text{ belong to different classes}$$
$$m_i(t+1) = m_i(t) \qquad \text{for } i \neq c \tag{3.3}$$

where x is an already classified sample and m_i is a prototype. In the training process, $x(t)$ then represent an input sample and $m_i(t)$ represents the value of the prototype m_i in the discrete time domain, $t = 0, 1, 2, ...$ The index c of the closest prototype m_i to x is then defined by $c = \arg\min_i\{d(m_i, x)\}$

$$c = \arg\min_i\{d(m_i, x)\}, \tag{3.4}$$

where function d is the chosen dissimilarity measure. α in Kohonen formal definition is known as the learning rate and lies between 0 and 1.

The process described above is known as LVQ1. Besides LVQ1 Kohonen defined a variety of similar learning algorithms, including LVQ2, LVQ3 and OLVQ [13]. We will be using a variant of Kohonens LVQ2 described in the next section.

3.2.1 Generalized Learning Vector Quantization

For our experiments with the Hankel matrix based disimilarity measures, we will use Generalized Learning Vector Quantization (GLVQ). Proposed in 1995 by Sato and Yamada GLVQ offers an alternative to heuristic LVQ which cannot be interpreted as a gradient based optimization of a cost function [14].

GLVQ is a costfunction based generalization of LVQ2.1. Like LVQ2.1, it updates both the closest prototype m_+ of the same class as the presented sample x_i and the closest prototype m_- of any other class.

The costfunction S, used by GLVQ as a criterion that should be minimized, is defined by

$$S = \sum_{i=1}^{N} f(\mu(x_i)), \tag{3.5}$$

where f is a monotonically increasing function and N is the number of input vectors for training. With function S representing the number of misclassifications, $\mu(x_i)$ should be positive when x_i is classified correctly and negative in all

other cases [14]. Sato and Yamada therefore propose to use the relative distance difference

$$\mu(x) = \left[\frac{d^+(x_i) - d^-(x_i)}{d^+(x_i) + d^-(x_i)}\right], \tag{3.6}$$

where $d^+(x_i)$ is defined as the distance from x_i to the closest prototype m_+ of the same class and $d^-(x_i)$ is defined as the distance from x_i to the closest prototype m_- of a different class.

Using equation (3.5) and (3.6), Sato and Yamada construct a modified update step for both prototypes m_+ and m_-.

$$m_+ = m_+ - \alpha\frac{\partial S}{\partial m_+} \tag{3.7}$$

$$m_- = m_- - \alpha\frac{\partial S}{\partial m_-} \tag{3.8}$$

To be able to use the update step equations (3.7) and (3.8) defined above with the gradient descent method, it is required that function f used in S is differentiable [14]. We will therefore use the sigmoid function for f and its derivative with respect to the prototypes m_- and m_+:

$$f(\mu(x_i)) = \frac{1}{1 + e^{-\mu(x_i)}}. \tag{3.9}$$

Using (3.9) we can now define the update step in terms of the derivative of the misclassification sum S.

$$m_+ \leftarrow m_+ - \alpha\frac{e^{-\mu(x_i)}}{(e^{-\mu(x_i)} + 1)^2} \cdot \frac{2d^-(x_i)}{(d^+(x_i) + d^-(x_i))^2} \cdot \frac{\partial d^+(x_i)}{\partial m_+} \tag{3.10}$$

$$m_- \leftarrow m_- - \alpha\frac{e^{-\mu(x_i)}}{(e^{-\mu(x_i)} + 1)^2} \cdot \frac{-2d^+(x_i)}{(d^+(x_i) + d^-(x_i))^2} \cdot \frac{\partial d^-(x_i)}{\partial m_-} \tag{3.11}$$

From (3.10) and (3.11) we can see that GLVQ requires a differentiable dissimilarity measure. Hence, if we wish to use GLVQ in combination with a Hankel matrix based dissimilarity measure, we will need to find the derivative of these measures with respect to the prototypes.

3.2.2 Generalized Relevance Learning Vector Quantization

In [15] B. Hammer and T. Villmann propose a further enhancement to GLVQ by introducing weighting factors λ for the input in the dissimilarity measure called Generalized Relevance Learning Vector Quantization, or GRLVQ in short.

Using stochastic gradient descent we can determine the appropriate weighting factors by introducing an update step for the n-dimensional vector λ representing the weighting factors in addition to the update step for the prototypes in (3.7) and (3.8).

$$\lambda \leftarrow \lambda - \eta \frac{e^{-\mu(x_i)}}{(e^{-\mu(x_i)}+1)^2} \cdot \left(\frac{2d^-(x_i)}{(d^+(x_i)+d^-(x_i))^2} \cdot \frac{\partial d^+(x_i)}{\partial \lambda} - \frac{-2d^+(x_i)}{(d^+(x_i)+d^-(x_i))^2} \cdot \frac{\partial d^-(x_i)}{\partial \lambda} \right), \tag{3.12}$$

where η is the learning rate for lambda with $\sum_i \lambda_i = 1$ for $\lambda_i > 0$.

Generalized Relevance Learning Vector Quantization can potentially be used in combination with the subspace angle method by introducing the weight factor lambda for each principal angle. This gives the following dissimilarity measure:

$$d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 1 - \sum_{i=1}^{d} \lambda_i \cos^{-1} \left(\max_{u \in F^{\perp}_{u_{i-1}}} \max_{v \in G^{\perp}_{v_{i-1}}} u^T v \right). \tag{3.13}$$

To be able to use relevance learning, the derivative of the dissimilarity measure is needed with respect to λ, as shown by eq. (3.12).

4 — Using GLVQ with Hankel Matrix based dissimilarity measures

In this chapter we will define the derivatives of the angle approximation and rotation approximation dissimilarity measures required for the training phase of GLVQ. In principle it is possible to derive a GLVQ variant for the subspace angle method as well. Due to the complicated structure of the subspace angle, this is however not a simple task. Therefore, we resort to the simpler approximation methods.

4.1 Angle approximation

To find the derivative of the angle approximation dissimilarity measure in (2.6) we use the fact that the derivative of a function with respect to a matrix can be defined as a combination of the derivatives of said function with respect to each element of the matrix:

$$\frac{\partial d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}} = \begin{bmatrix} \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{11}} & \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{12}} & \cdots & \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{1d}} \\ \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{21}} & \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{22}} & \cdots & \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{2d}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{r1}} & \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{r2}} & \cdots & \frac{\partial d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{rd}} \end{bmatrix}. \tag{4.1}$$

We consider the multiplication of a matrix with dimension $r \times d$ with its transpose to be defined as

$$\hat{\boldsymbol{H}}_{\boldsymbol{v}} \hat{\boldsymbol{H}}_{\boldsymbol{v}}^{\boldsymbol{T}} = \begin{bmatrix} \sum_{k=1}^{d} v_{1k} v_{1k} & \sum_{k=1}^{d} v_{1k} v_{2k} & \cdots & \sum_{k=1}^{d} v_{1k} v_{rk} \\ \sum_{k=1}^{d} v_{2k} v_{1k} & \sum_{k=1}^{d} v_{2k} v_{2k} & \cdots & \sum_{k=1}^{d} v_{2k} v_{rk} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^{d} v_{rk} v_{1k} & \sum_{k=1}^{d} v_{rk} v_{2k} & \cdots & \sum_{k=1}^{d} v_{rk} v_{rk} \end{bmatrix}, \tag{4.2}$$

with dimension $r \times r$.

This allows us to write the dissimilarity measure using the definition of the Frobenius norm given in (2.5) as

$$d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 4 - \sum_{i=1}^{r} \sum_{j=1}^{r} (\sum_{k=1}^{d} p_{ik} p_{jk} + \sum_{k=1}^{d} q_{ik} q_{jk})^2. \tag{4.3}$$

Using the chain rule we can find the derivative with respect to p_{ab} as

$$\frac{\partial d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{ab}} = -2 \sum_{i=1}^{r} \sum_{j=1}^{r} (f(i,j) \cdot \frac{\partial f(i,j)}{\partial p_{ab}}), \tag{4.4}$$

where

$$f(i,j) = \sum_{k=1}^{d} p_{ik}p_{jk} + \sum_{k=1}^{d} q_{ik}q_{jk} \tag{4.5}$$

$$= \sum_{k=1}^{d} p_{ik}p_{jk} + q_{ik}q_{jk}. \tag{4.6}$$

Considering that we are looking for the derivative with respect to p_{ab}, we are interested in determining when either $p_{ik} = p_{ab}$ or $p_{jk} = p_{ab}$. As both scenarios require $k = b$ we are left with two cases to consider, when $i = a$ and when $j = a$. Resulting in three different derivatives.

$$\frac{\partial f(i,j)}{\partial p_{ab}} = \begin{cases} p_{jb}, & \text{for } i = a \\ p_{ib}, & \text{for } j = a \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

Combining (4.4), (4.6), (4.7) and using the fact that the derivative of $f(i,j)$ with respect to p_{ab} equals 0 when both $i \neq a$ and $j \neq a$ we can write

$$\frac{\partial d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{ab}} = -2(\sum_{i=1}^{r} p_{ib} \cdot f(i,a) + \sum_{j=1}^{r} p_{jb} \cdot f(a,j)) \tag{4.8}$$

$$= -2(\sum_{i=1}^{r} p_{ib}(f(i,a) + f(a,i)). \tag{4.9}$$

To conclude our search for the derivative with respect to $p_{a,b}$, we note that $f(i,a) = f(a,i)$, allowing us to write the derivative as

$$\frac{\partial d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{ab}} = -4(\sum_{i=1}^{r} \cdot p_{ib} \cdot \sum_{k=1}^{d} p_{ik}p_{ak} + q_{ik}q_{ak}). \tag{4.10}$$

Now that the derivative with respect to one element is known, we can use this definition in combination with (4.1) to define the derivative of $d_1(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})$ with respect to the matrix $\hat{\boldsymbol{H}}_{\boldsymbol{p}}$.

4.2 Rotation Approximation

To find the derivative of $d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 4 - ||\hat{\boldsymbol{H}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}}||_F^2$ with respect to the prototype $\hat{\boldsymbol{H}}_{\boldsymbol{p}}$, we consider

$$\hat{\boldsymbol{H}}_{\boldsymbol{p}} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1d} \\ p_{21} & p_{22} & \cdots & p_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ p_{r1} & p_{r2} & \cdots & p_{rd} \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{H}}_{\boldsymbol{q}} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1d} \\ q_{21} & q_{22} & \cdots & q_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ q_{r1} & q_{r2} & \cdots & q_{rd} \end{bmatrix}.$$

Using the definition of the Frobenius norm in (2.5) this allows us to rewrite the dissimilarity measure as

$$d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) = 4 - \sum_{i=1}^{r}\sum_{j=1}^{d}(p_{ij} + q_{ij})^2. \tag{4.11}$$

The derivative with respect to one element of $d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})$ can now be defined as

$$\frac{\partial d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial p_{ab}} = -2(p_{ab} + q_{ab}). \tag{4.12}$$

Using the definition of the derivative of a function with respect to a matrix as given in (4.1), the obtained derivative for the rotation approximation distance measure is

$$\frac{\partial d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}} = -2(\hat{\boldsymbol{H}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}}). \tag{4.13}$$

5 — Experiments

In this chapter we will introduce the experiments we will conduct to validate the use of the two proposed approximations. In the first experiment we extend on the experiments van Loon conducted in his paper [2] to get an indication of the performance of GLVQ using the two dissimilarity measures.

In the second experiment we will investigate the influence of two important hyper-parameters, the dimension of the Hankel matrix and the number of prototypes for each class.

5.1 Experiment I: Real world time-series

In his paper [2], van Loon uses 42 datasets from the UCR time-series repository [6] to validate the use of the Hankel matrix based dissimilarity measures. Due to GLVQ requiring an expensive training phase, we are not be able to use the same number of datasets to validate the use of the dissimilarity measures with GLVQ. Instead we select four datasets based on the classification performance of kNN. More information on the dataset used can be found in section 5.1.1.

For each of the dataset we perform the training phase of GLVQ with both the dissimilarity measures. During the training phase, one prototype is trained for each class in the dataset. This prototype is initialized as one randomly selected time-series of the respective class. To reduce the influence of the selected prototype the training phase is repeated ten times, each time with a new randomly selected time-series as initial prototype. Due to the UCR repository providing disjoint training and testsets, it is not required to use n-fold cross validation.

For this experiment we use a fixed dimension for the Hankel matrices, the same dimensions as van Loon used in his experiments.

5.1.1 Datasets

1. **CBF.** Data in the Cylinder-Bell-Funnel dataset is generated using a standard normal noise combined with a different offset for each of the three classes. The CBF dataset was chosen to be used because of the excellent results achieved using the Angle Approximation in combination with kNN.

2. **Coffee.** Using all three dissimilarity measures it was possible to achieve perfect classification using kNN for the Coffee dataset. As the name suggests, the two classes in the dataset are spectrograms representing Robusta and Arabica coffee beans. The dataset was first introduced by Bagnall et al. in their paper "Transformation Based Ensembles for Time Series Classification" [16].

3. **ECGFiveDays.** The ECGFivedays dataset performed equally well for both the subspace angle dissimilarity measure and the angle approximation. The data is a collection of ECGs gathered from a 67 year old male over the course of two days: 12 and 17 November 1990. Both days are represented by a single class.

4. **SyntheticControl.** The six classes in the SyntheticControl dataset depict different control time-series, each with a specific characteristic: 1. Normal 2. Cyclic 3. Increasing trend 4. Decreasing trend 5. Upward shift 6. Downward shift. More information about the classes can be found in the original paper *Time-Series Similarity Queries Employing a Feature-Based Approach* by R. J. Alcock and Y. Manolopoulos [17]. The dataset is included in the experiment because of the low classification error achieved using k-Nearest Neighbours with the rotation approximation.

Metadata

Name Dataset	Number of Classes	Training Set Size	Testing Set Size	Time-series Length
CBF	3	30	900	128
Coffee	2	28	28	286
ECGFiveDays	2	23	861	136
SyntheticControl	6	300	300	60

Table 5.1: UCR time-series repository dataset metadata

5.2 Experiment II: Influence hyper-parameters

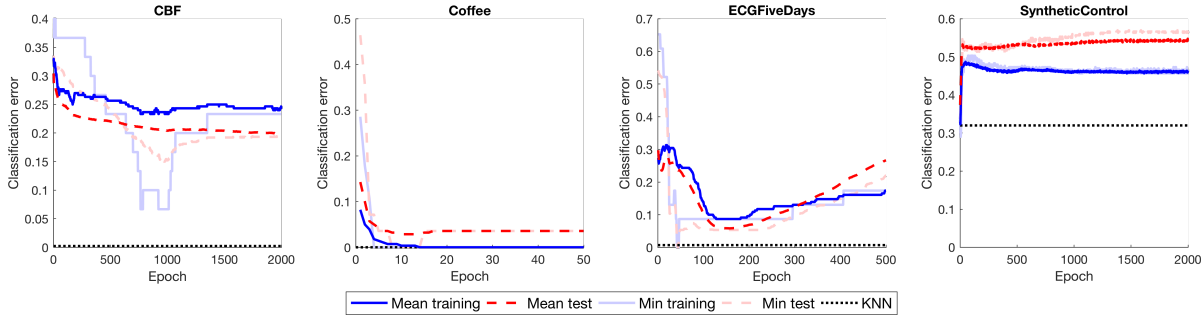
Experiment II focusses on the influence of matrix dimension and the number of prototypes used for each class on the classification error. Based on the results presented in the next chapter the CBF dataset is used for both the measures.

5.2.1 Dimension of the Hankel matrix

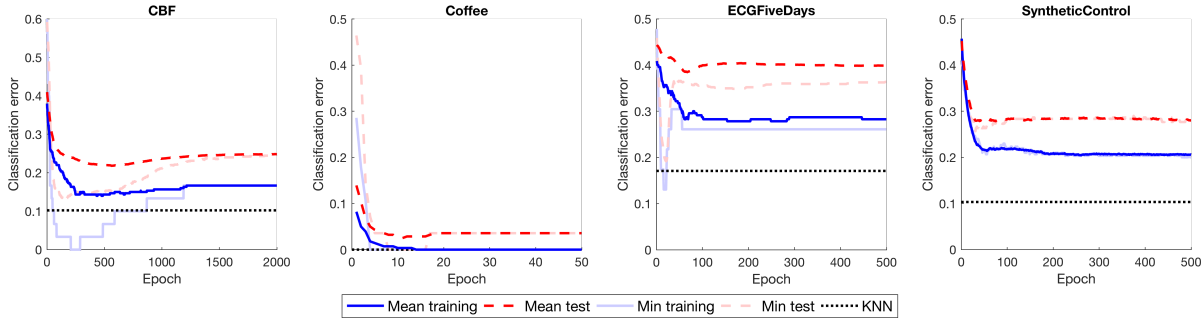
In this experiment, we investigate the influence of the dimension of a Hankel matrix on the classification error given by the dissimilarity measures. Unfortunately it is not possible to perform a full sweep of the possible dimensions for both the dissimilarity measures.

5.2.2 Number of prototypes

Increasing the number of prototypes used for each class in Learning Vector Quantization potentially improves the quality of the classification. It however also prolongates the training procedure and increases the amount of memory needed [12]. In this experiment, we will therefore investigate how much classification improves when more prototypes are used and whether this is worth the increase in overhead.



(a) *Angle approximation*



(b) *Rotation approximation*

Figure 6.1: *Mean and minimum classification error of the training and test set after each epoch for both the dissimilarity measures in combination with GLVQ*

6 — Results

6.1 Experiment I: Real world time-series

In figure 6.1 the classification error of both the training and test set of all four datasets after each epoch are presented for both the dissimilarity measures. The black dotted line in each graph represents the classification error achieved by van Loon using k-Nearest Neighbours. As kNN does not require a training phase, this value is represented as a constant.

The development of the classification error is summarized in table 6.1 where the minimum classification error achieved over the ten repetitions is given together with the classification error using kNN. In addition to the minimum classification error itself, the epoch after which it was achieved is listed between brackets.

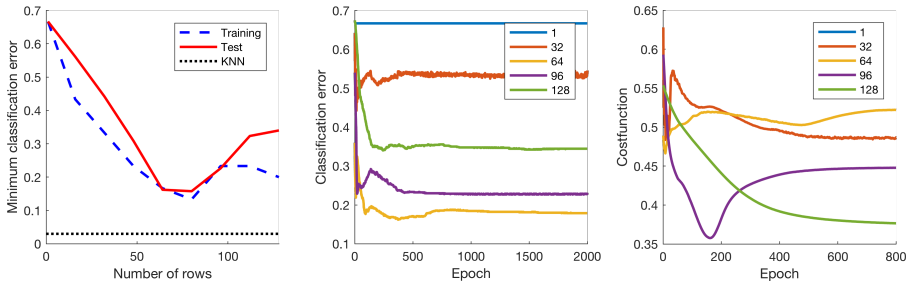
	Angle appr.		Rotation appr.	
Dataset	kNN	glvq (e)	kNN	glvq (e)
CBF	0.00	0.15 (960)	0.010	0.13 (152)
Coffee	0.00	0.00 (8)	0.00	0.00 (8)
ECGFiveDays	0.01	0.00 (44)	0.17	0.19 (21)
SyntheticControl	0.32	0.34 (1)	0.10	0.26 (29)

Table 6.1: *Minimum classification error achieved using both the dissimilarity measures in combination with kNN and GLVQ*

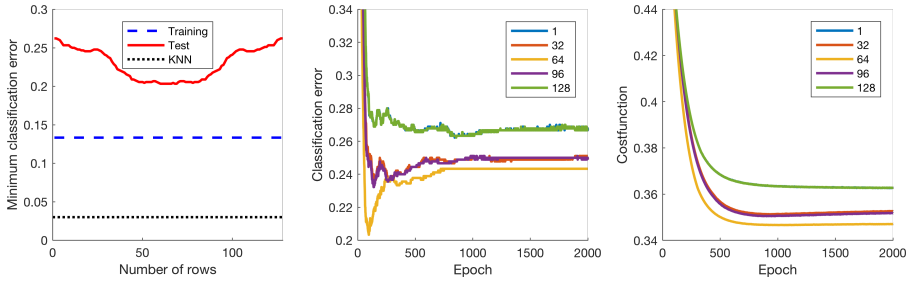
6.2 Experiment II: Influence hyper parameters

6.2.1 Dimension hankel matrix

In figure 6.2 the influence the dimension of the Hankel matrix has on the classification error and the development of the costfunction is given for both the dissimilarity measures. Each dimension is represented as the number of rows.



(a) *Angle approximation*

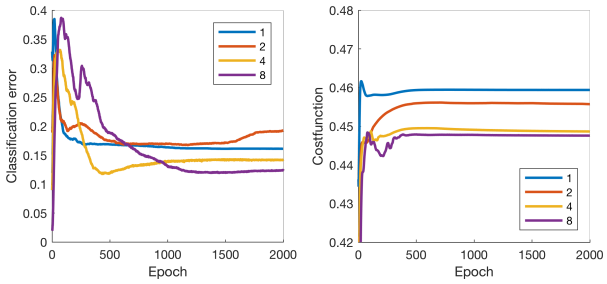


(b) *Rotation approximation*

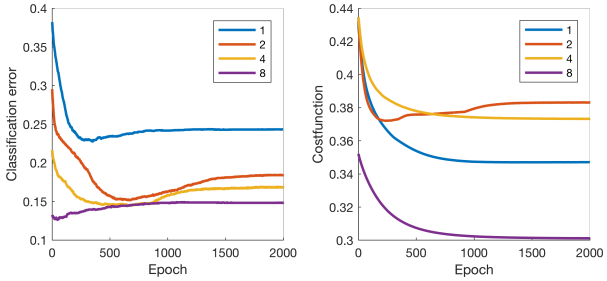
Figure 6.2: *Influence of the Hankel matrix dimension on the classification results achieved by both the dissimilarity measures. From left to right the minimum classification error achieved versus the number of rows, the classification error after each epoch and the costfunction after each epoch for different dimensions is given.*

6.2.2 Number of prototypes

The influence of the number of prototypes used for each class on the classification error is shown in figure 6.3. For both the dissimilarity measure, the classification error and costfunction development for one, two, four and eight prototypes is given. It was not possible to instantiate more prototypes due to the limited number of training samples available in the CBF dataset.



(a) *Angle approximation*



(b) *Rotation approximation*

Figure 6.3: *Influence of the number of prototypes used for each class on the classification results achieved by each of the two dissimilarity measures. From left to right for both the dissimilarity measures the development of the classification error and the costfunction for different number of prototypes is given.*

7 — Discussion

In this chapter we will discuss the results presented in the previous chapter. The results will be discussed by dissimilarity measure.

7.1 Angle approximation

The results for the angle approximation method show a large difference between the four selected prototypes. With exception of the Coffee dataset, the classification error for GLVQ is not competitive with kNN as can be seen in figure 6.1a. For two of the four used real-world datasets, ECGFiveDays and SyntheticControl, the development of the classification error indicates that the method is not working at all for these datasets. For the SyntheticControl dataset the initially chosen prototypes provide significantly better classification than the trained prototypes.

Figure 6.2a shows that the lowest classification error is achieved when using as square as possible matrices. The same figure however also shows that for most possible dimensions of the Hankel matrix the costfunction is not minimized by the used derivative, possibly explaining why the method does not work for the real world dataset. The right most plot of 6.2a shows that when a dimension is used with less columns, the derivative can better minimize the costfunction. This is in line with the two datasets with the worst results, ECGFiveDays and SyntheticControl, using the highest ratio between the time-series length and the number of columns with 0.375 and 0.567 respectively.

One more important insight is provided by the learned prototypes. Unexpectedly, the prototypes learned using the derivative of the angle approximation method are not Hankel matrices themselves.

7.2 Rotation approximation

The rotation approximation dissimilarity measure show great results for all four-real life data sets. While the results in figure 6.1b clearly show a difference between the classification error achieved using GLVQ and kNN, it is possible to achieve results comparable to kNN using a reasonable amount of prototypes per class as shown in figure 6.3a.

The classification error achieved using the rotation approximation can be further improved by using the right dimensions for the Hankel matrices. Figure 6.2b shows that the Hankel matrices should be constructed as square as possible to minimize the classification error.

8 — Conclusion

In the previous 7 chapters we have looked at the use of Hankel matrix based dissimilarity measures in combination with Generalized Learning Vector Quantization. We have tried to validate the use of the presented derivatives and evaluated two important hyper parameters of the classification approaches. In the last two chapters we have presented the results of our experiment with moderate success.

From the presented results regarding the angle approximation dissimilarity measure, we must conclude a important oversight in our used derivatives. Both the learned prototypes not representing a Hankel matrix and the development of the costfunction, indicate that we have neglected to see the importance of the constant anti-diagonals in Hankel matrices. This neglect has resulted in a derivative for the angle approximation dissimilarity that is neither symmetric, nor does it respect the properties of Hankel matrices.

Due to the simplicity of the rotation approximation, this method does not suffer from this neglect, resulting in a promising method for classifying time-series.

9 — Future work

The results presented and discussed in the last three chapter have left a lot of questions open for future work. In this chapter we will discuss some of these topics.

9.1 Subspace angle dissimilarity measure

In this work we have intentionally left out the use of the subspace angle method in Generalized Learning Vector Quantization. The subspace angle dissimilarity measure given in eq. (2.4) uses a *max* operation to find the principal angles, this makes finding the derivative of the angle dissimilarity measure a complex tasks. In [2] van Loon however shows that the subspace angle method can provide competitive results when combined with K-Nearest neighbours. If possible, finding the derivative of the subspace angle dissimilarity measure and applying it in GLVQ can potentially provide the same competitive results.

Further work related to the subspace angle method can be extended using relevance learning as introduced in section 3.2.2.

9.2 Derivatives with respect to Hankel matrices

The method used in our work to find the derivatives of the two approximation methods does not take into account the properties of Hankel matrices. Finding the derivative with respect to a Hankel matrix is not an easy task for high order functions and unfortunately not a lot of research has been conducted in this area. A possible solution to this problem can be found by comparing Hankel matrices with Toeplitz matrices.

A Toeplitz matrix is a $n \times n$ matrix with constant diagonals. Toeplitz matrices see a lot of use in a wide variety of applications and have therefore been part of more research than Hankel matrices. By exploring the idea that Hankel matrices are upside down Toeplitz matrices we can use the derivative with respect to a Toeplitz matrix presented in [18] to construct a similar method for finding the derivative with respect to a Hankel matrix $\boldsymbol{H}$:

$$
\begin{aligned}
\frac{\partial \mathrm{Tr}(\boldsymbol{AH})}{\partial \boldsymbol{H}} &= \frac{\partial \mathrm{Tr}(\boldsymbol{HA})}{\partial \boldsymbol{H}} \\
&= \begin{bmatrix}
A_{n,1} & \dots & \mathrm{Tr}([[\boldsymbol{A}^T]_{1,n}]_{2,n-1}) & \mathrm{Tr}([\boldsymbol{A}^T]_{1,n}) & \mathrm{Tr}(\boldsymbol{A}) \\
\vdots & \iddots & \iddots & \mathrm{Tr}(\boldsymbol{A}) & \mathrm{Tr}([\boldsymbol{A}^T]_{n,1}) \\
\mathrm{Tr}([[\boldsymbol{A}^T]_{1,n}]_{2,n-1}) & \iddots & \iddots & \iddots & \mathrm{Tr}([[\boldsymbol{A}^T]_{n,1}]_{n-1,2}) \\
\mathrm{Tr}([\boldsymbol{A}^T]_{1,n}) & \mathrm{Tr}(\boldsymbol{A}) & \iddots & \iddots & \vdots \\
\mathrm{Tr}(\boldsymbol{A}) & \mathrm{Tr}([\boldsymbol{A}^T]_{n,1}) & \mathrm{Tr}([[\boldsymbol{A}^T]_{n,1}]_{n-1,2}) & \dots & A_{1,n}
\end{bmatrix} \\
&\equiv \alpha(A). \qquad (9.1)
\end{aligned}
$$

We can use the presented derivative in combination with our dissimilarity measures if we are able to rewrite them as a combination of trace operations.

Luckily, the squared Frobenius norm used for both the approximation methods can be expressed using the trace operation

$$||\boldsymbol{H}||_F^2 = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2}^2 \tag{9.2}$$

$$= \mathrm{Tr}(H^T H). \tag{9.3}$$

Using (9.3) we can rewrite the rotation approximation dissimilarity measure

$$\begin{aligned} d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) &= 4 - ||\hat{\boldsymbol{H}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}}||_F^2 \\ &= 4 - \mathrm{Tr}((\hat{\boldsymbol{H}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}})^T(\hat{\boldsymbol{H}}_{\boldsymbol{p}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}})) \\ &= 4 - \mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}) - 2\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}) - \mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}) \end{aligned} \tag{9.4}$$

and the angle approximation dissimilarity measure as

$$\begin{aligned} d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}}) &= 4 - ||\hat{\boldsymbol{H}}_{\boldsymbol{p}} \cdot \hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}} \cdot \hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}||_F^2 \\ &= 4 - \mathrm{Tr}((\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}})^T(\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}} + \hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}})) \\ &= 4 - \mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}) - 2\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}) \\ &\quad - \mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}). \end{aligned} \tag{9.5}$$

.

Now that we have expressed the two approximation dissimilarity measures as a summation of trace operations in equation (9.4) and (9.5) we can use equation (9.1) to find their derivatives:

$$\begin{aligned} \frac{\partial d_3(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}} &= -\frac{\partial}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}}\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}) - \frac{\partial}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}}2\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}) - \frac{\partial}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}}\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}) \\ &= -\alpha(2\hat{\boldsymbol{H}}_{\boldsymbol{p}}) - \alpha(2\hat{\boldsymbol{H}}_{\boldsymbol{q}}) \\ &= -2(\alpha(\hat{\boldsymbol{H}}_{\boldsymbol{p}}) + \alpha(\hat{\boldsymbol{H}}_{\boldsymbol{q}})) \end{aligned} \tag{9.6}$$

$$\begin{aligned} \frac{\partial d_2(\hat{\boldsymbol{H}}_{\boldsymbol{p}}, \hat{\boldsymbol{H}}_{\boldsymbol{q}})}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}} &= -\frac{\partial}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}}\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}) - \frac{\partial}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}}2\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}{}^T) \\ &\quad - \frac{\partial}{\partial \hat{\boldsymbol{H}}_{\boldsymbol{p}}}\mathrm{Tr}(\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}) \\ &= -\alpha(4\hat{\boldsymbol{H}}_{\boldsymbol{p}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}) - \alpha(4\hat{\boldsymbol{H}}_{\boldsymbol{q}}\hat{\boldsymbol{H}}_{\boldsymbol{q}}^{\boldsymbol{T}}\hat{\boldsymbol{H}}_{\boldsymbol{p}}). \end{aligned} \tag{9.7}$$

.

Equation (9.6) shows why the rotation approximation derivative used in this thesis still shows promising results without taking the properties of Hankel matrices in account when constructing the derivative. The derivative used in the thesis is very similar to the derivative presented above. This is not the case for more complex dissimilarity measures such as the angle approximation method.

In some preliminary experiments with the derivatives presented above it has already been shown that the simple rotation approximation dissimilarity measure produces competitive results. This can be seen in figure 9.1 , showing the

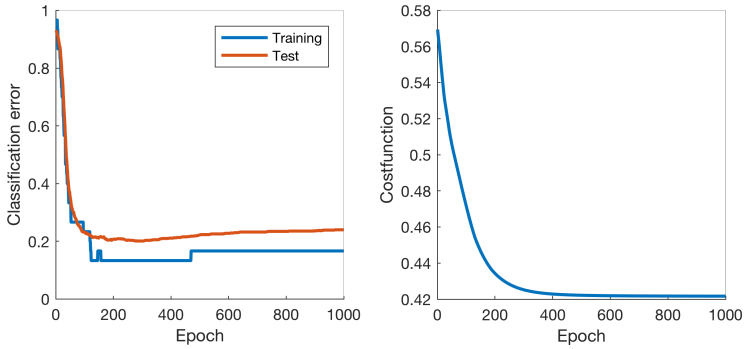


Figure 9.1: *Subspace angle*

development of the classification error and costfunction. For the more complex angle approximation measure, further research is needed in future work. It must also be noted that for the derivatives to exist as presented above the Hankel matrices are required to be square.

9.3 Multidimensional time-series

In addition to further research into the flaws of the dissimilarity measure exposed by this thesis the use of the dissimilarity measure could be further expanded by looking into n-dimensional time-series. In his thesis [3] Mohammadi already addresses the possibility of doing so with the Hanklets method. He explores this idea further in his paper [4].

Bibliography

[1] Anthony Bagnall, Aaron Bostrom, James Large, and Jason Lines. The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version. *arXiv:1602.01711 [cs]*, Feb 2016. arXiv: 1602.01711.

[2] S.J. van Loon. Comparison of hankel based similarity metrics for time-series classification. Master's thesis, University of Groningen, 2018.

[3] Mohammad Mohammadi. Hankel matrices for use in learning vector quantization. Master's thesis, Hochschule Mittweida, University of Applied Sciences, 2016.

[4] Mohammad Mohammadi, Michael Biehl, Andrea Villmann, and Thomas Villmann. *Sequence Learning in Unsupervised and Supervised Vector Quantization Using Hankel Matrices*, page 131–142. Springer International Publishing, 2017.

[5] Binlong Li, Mustafa Ayazoglu, Teresa Mao, Octavia I. Camps, and Mario Sznaier. *Activity recognition using dynamic subspace angles*, page 3193–3200. IEEE, Jun 2011.

[6] Anthony Bagnall, Eamonn Keogh, William Vickers, and Jason Lines. The uea and ucr time series classification repository.

[7] Peter Van Overschee and Bart De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, May 1993.

[8] Binlong Li, O. I. Camps, and M. Sznaier. *Cross-view activity recognition using Hankelets*, page 1362–1369. IEEE, Jun 2012.

[9] Liliana Lo Presti, Marco La Cascia, Stan Sclaroff, and Octavia Camps. *Gesture Modeling by Hanklet-Based Hidden Markov Model*, volume 9005, page 529–546. Springer International Publishing, 2015.

[10] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, Jan 1967.

[11] Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238, Dec 1989.

[12] M. Biehl. Prototype-based supervised learning and adaptive distances, October 2018.

[13] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag, Berlin, Heidelberg, 3rd edition, 2001.

[14] Atsushi Sato and Keiji Yamada. *Generalized learning vector quantization*, page 423–429. 1996.

[15] Barbara Hammer and Thomas Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8–9):1059–1068, Oct 2002.

[16] Anthony Bagnall, Luke Davis, Jon Hills, and Jason Lines. *Transformation Based Ensembles for Time Series Classification*, page 307–318. Society for Industrial and Applied Mathematics, Apr 2012.

[17] R. J. Alcock, Y. Manolopoulos, Data Engineering Laboratory, and Department Of Informatics. Time-series similarity queries employing a feature-based approach. In *In 7 th Hellenic Conference on Informatics, Ioannina*, pages 27–29, 1999.

[18] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. Technical University of Denmark, nov 2012. Version 20121115.